Generate or Rerank? Where to Invest an LLM's Budget in Cold-Start E-Commerce Recommender Systems

Christian Freitas[1], **Lilian Berton**[1]

[1]Instituto de Ciência e Tecnologia – Universidade Federal de São Paulo (UNIFESP)
São José dos Campos – SP – Brazil

{christian.freitas, lberton}@unifesp.br

Abstract. *E-commerce recommenders retrieve candidates cheaply, then rerank them at higher cost, and large language models (LLMs) increasingly occupy that second stage, expected to help customers with little purchase history (user cold-start). We show the second stage has little left to work with: in an Amazon category where 65.6% of customers have at most three purchases, the item bought next reaches the top ten candidates in only ≈2% of cases, capping any reranker at ≈3%. The cause is behavioural, as the next purchase usually falls in a different subcategory. Off-the-shelf LLM generation does not help either, losing to classical retrieval at every cutoff for two providers (paired McNemar, $p < 0.02$). What helps is free: reciprocal rank fusion of four cheap signals recovers ≈40% more coverage ($p < 0.001$). The twofold residual gap to an oracle reflects intent the purchase history never records.*

1. Introduction

The use of LLMs in recommender systems (RecSys) has grown along two main lines: direct generation of recommendations and **reranking** of candidates produced by classical models. Reranking is the more common and seemingly safer form, since it confines the LLM to the top of an already generated list, limiting cost and latency. The implicit promise is that the LLM's semantic understanding of items helps precisely where collaborative models are weakest: the **cold-start**, where the user has scarce history.

Recent work, however, shows this promise does not hold: in cold-start, LLM-based rerankers or cross-encoders can be beaten by trivial baselines such as popularity, and the failure has been attributed to **retrieval coverage**: the relevant item is rarely among the candidates that reach the reranker [Lemdiasova and Zmanovskii 2026]. That diagnosis, however, was established for a discriminative *cross-encoder*, in a movie domain, measuring quality only. Three questions remain open, which this work attacks: (i) does the diagnosis hold for **e-commerce**? (ii) does a **generative LLM**, whose value proposition is world knowledge rather than similarity, change the picture? (iii) what if the LLM's budget is reallocated from *ranking* to *generating* candidates?

We begin by confirming the diagnosis in our domain, measuring how often the target item reaches the top-K of each generator. It rarely does: in severe cold-start the target is in the top-10 in only ≈2% of cases for any model, which leaves nothing to reorder in ≈97% of them and imposes a **ceiling of ≈3% on any reranker** over the top-20. The cause is that the next purchase tends to lie in a **different subcategory** from the history (mean content similarity 0.43): similarity retrieval looks for the alike, but the customer

buys the complementary. This shifts the research question from how to better order the candidates to where to invest the LLM's budget:

> **RQ1:** In e-commerce cold-start, does allocating a generative LLM's budget to *generating* complementary candidates cover more target items than (a) classical content-based retrieval and (b) using the same LLM as a reranker, and at what latency and monetary cost?
>
> **RQ2:** If the LLM does not recover the lost coverage, how much of it is recoverable *without* interaction, by combining cheap, complementary classical signals?

Contributions. (1) A *coverage diagnosis* quantifying the structural ceiling of any reranker in e-commerce cold-start ($\approx$3% in the top-20), and the cross-subcategory purchasing that causes it. (2) A *negative result* on the generative LLM: reallocating its budget from reranking to generating complementary candidates does not beat classical content-based retrieval in any configuration (paired McNemar, $p < 0.02$, for two LLMs from distinct providers), evaluated under an ROI protocol that reports coverage beside latency and monetary cost, and controlled for the language of the generated text. This extends to *e-commerce* and to *generative* LLMs a diagnosis previously restricted to cross-encoders on movies. (3) A *positive, prescriptive result*: RRF fusion of four classical signals, at negligible cost, recovers $\approx$40% more coverage than the best single signal (Severe@20: 4.6% vs. 3.4%, $p < 0.001$), while a *learned* fusion only ties it, marking the ceiling of interaction-free methods. The residual gap to a fusion oracle ($\approx 2\times$) reflects the customer's *unexpressed intent*, which no static signal reveals.

2. Related Work

Collaborative filtering. Matrix factorization, with implicit (ALS) [Hu et al. 2008] or explicit feedback, is the reference method when history is abundant but degrades in cold-start by depending on observed co-occurrence between users and items: with one or two interactions per user, the latent factors become uninformative and the model falls back on popular items. This is precisely the predominant regime in our dataset.

Content-based recommendation. Similarity search between item embeddings, here via FAISS [Johnson et al. 2019] over Sentence-BERT representations [Reimers and Gurevych 2019], sidesteps *user* cold-start by not requiring collaborative history, and is widely adopted as a first-stage retriever. It is, however, indifferent to co-occurrence patterns and, as we show, penalized when the next purchase changes subcategory: it retrieves the semantically alike, not the complementary item the customer actually buys.

LLMs in RecSys. Recent surveys map the field by adaptation paradigm, from feature encoding through fine-tuning to prompting [Zhao et al. 2024], and by the role the LLM plays in the pipeline, such as item encoder, generator, and reranker [Shehmir and Kashef 2025]. Concretely, LLMs have been used for sequential recommendation [Harte et al. 2023], diversity-oriented reranking [Carraro and Bridge 2024], and reasoning over items in cold-start [Li et al. 2026]. Most of these works assume the LLM's gain comes from semantic understanding applied *over* already retrieved candidates. The work closest to ours questions that premise,

diagnosing the failure of LLM-based rerankers in cold-start as one of retrieval coverage (a recent preprint [Lemdiasova and Zmanovskii 2026]), but does so for a *discriminative* cross-encoder, in a movie domain and measuring quality only. We differ on four axes: (i) a *generative* LLM, whose value proposition is world knowledge rather than similarity; (ii) *e-commerce*; (iii) reallocating the LLM from ranking to *generating* candidates; and (iv) cost and latency (ROI) as a first-class metric.

Rank list fusion. Reciprocal Rank Fusion (RRF) [Cormack et al. 2009] aggregates ordered lists by the reciprocal of ranks, without training or score calibration, and is notoriously hard to beat with learned methods. We use it not as a text-retrieval technique but as a cheap way to combine heterogeneous recommendation signals in cold-start, a use little explored in this regime.

Conversational recommendation. When history is insufficient, an alternative to passive inference is to *elicit* the preference through dialogue; conversational recommender systems (CRS) formalize this interaction and are covered by comprehensive surveys [Jannach et al. 2021, Gao et al. 2021], with recent approaches modelling the CRS as a multi-agent orchestration of LLMs in which architecture, not just model size, determines robustness [Santos Silva et al. 2026]. Our work is logically prior to this line: by quantifying the ceiling of what is recovered *without* interaction and the residual gap to an oracle, we supply an empirical motivation for when conversational interaction becomes necessary, rather than assuming it.

3. Coverage Diagnosis

3.1. Dataset

We use the Amazon Reviews 2023 dataset [Hou et al. 2026], restricted to the *Cell Phones and Accessories* category, with explicit 1-to-5-star ratings. We chose this category deliberately: it is an accessory-heavy domain, where complementary purchasing (a case, then a charger, then a mount) is the norm rather than the exception, which is precisely the behaviour a semantic retriever should struggle with and an LLM's world knowledge should, in principle, capture. It is therefore a favourable setting for the LLM, not an adversarial one. After filtering, the dataset comprises 36,689 training interactions across 10,139 users and 13,174 items, corresponding to an extremely sparse user-item matrix with density of about 0.027%. The rating distribution is strongly skewed toward positive ratings: about 68% of interactions have 5 stars, which reduces the signal variance that rating-based methods can exploit. We therefore treat every interaction as *implicit* feedback, the signal being the presence of the purchase and not its star rating; held-out targets are not filtered by rating (8% carry 1–2 stars), so the positive skew does not bias the signals we use. We adopt the *leave-one-out* protocol: for each user, one interaction is held out for validation, constituting a single relevant item, and the rest form the training history. Table 1 summarizes the statistics. The content catalog contains 16,123 items, all with a textual title and embedding; the target item is present in the catalog in 100% of cases and was seen in training in 75.7%, so there is no structural impossibility of retrieving it.

3.2. The challenge of recommending without interactions

The median history is just 3 interactions and 65.6% of users have at most three, so cold-start is the predominant regime, not an edge case. Two complementary reasons make it

hard here. Collaborative signal is scarce: with one or two purchases per user the latent factors become little more than noise and the model falls back on popular items. Less obviously, content information is also of limited use in this domain, for a reason we quantify in Section 3.5. It is this lack of signal that motivates employing an LLM, and also what limits what the LLM can do.

Table 1. Dataset statistics and segmentation by history density.

Interactions (train / valid / test)	36,689 / 10,139 / 10,139
Users	10,139
Items in train / catalog with embedding	13,174 / 16,123
Median history (train)	3
Severe (cold-start, ≤ 3)	6,656 (65.6%)
Moderate (4–10)	3,102 (30.6%)
Warm (11–50)	375 (3.7%)

3.3. Coverage metric

We define coverage@K as the fraction of users whose target item appears among the top K candidates of a generator. Under leave-one-out with exactly one relevant item per user, this quantity *coincides with* HR@K (hit rate): it is the standard metric, named here for the role it plays in our argument, namely the quality ceiling achievable by any reranker operating over those K candidates. We prefer the name *coverage* because it emphasises that the number is a property of the *retrieval* stage, which the ranking stage cannot exceed.

Because a single relevant item makes the ideal DCG equal to one, the only further information a ranking metric can add is *where* in the list the target fell. We therefore also report NDCG@$K = 1/\log_2(r+2)$ when the target is retrieved at zero-based rank $r < K$, and zero otherwise. Precision@K and recall@K add nothing here: recall@K equals coverage@K, and precision@K equals coverage@K divided by K. We evaluate four generators: Popularity, FAISS (content), ALS (collaborative), and a dynamic ALS+FAISS Ensemble.

3.4. Result: the bottleneck is generation

Table 2 shows the central finding. In severe cold-start, no generator places the target in the top-10 in more than 2.5% of cases; even in the top-100 coverage stays below 10%. Since the reranking stage typically operates over the top-20, its hit ceiling is ≈3.3% (Ensemble@20), *regardless* of how good the LLM is. We further note that (i) popularity, a trivial baseline, beats the dynamic Ensemble, and (ii) collaborative ALS is the worst in all segments, confirming the collapse of the collaborative signal under scarce history.

3.5. Why coverage is so low

The content similarity (cosine) between the target item and the user's history averages only 0.43 (median 0.43; p25 0.34; p75 0.51). This indicates that the next purchase usually belongs to a *different subcategory* from the previous ones: similarity retrieval looks for the alike, but the real behavior is complementarity. This is the mechanism that lowers coverage and that motivates the next section.

Table 2. Coverage@K (%) per segment. The target is rarely among the candidates, and coverage stays below 10% even at $K = 100$; reranking, which operates at $K \leq 20$, inherits this ceiling.

Segment	Model	@10	@20	@50	@100
Severe	Popularity	2.5	3.4	6.0	9.1
	Ensemble	2.1	3.3	5.6	8.5
	FAISS	1.8	2.9	5.4	8.0
	ALS	1.3	2.0	3.2	4.6
Moderate	Popularity	1.9	2.8	5.1	7.5
	Ensemble	1.4	2.2	4.4	7.4
	FAISS	1.2	2.2	4.6	7.3
	ALS	1.2	1.7	2.9	4.2
Warm	FAISS	1.1	1.6	2.9	5.1
	Ensemble	1.1	1.6	2.9	5.1
	Popularity	0.8	1.6	4.0	7.2
	ALS	0.5	0.5	1.1	2.7

4. Reallocating the LLM Budget: Candidate Generation

Figure 1 summarizes the two strategies we evaluate from the same cold-start history: reallocating the LLM budget to candidate generation, detailed in this section, and the cheap fusion of classical signals, presented later.

If the bottleneck is coverage and the cause is the jump between subcategories, then reordering cannot help, but *generating* candidates might. We propose using the generative LLM as a generator: from the history titles, the LLM produces N natural-language descriptions of likely next purchases, with an explicit instruction to favor *complementary* items (accessories, matching items, natural next purchases). Each description is encoded by the same catalog encoder (Sentence-BERT, 384 dimensions) and mapped to the nearest real items via FAISS; the scores are aggregated to form the final list. The LLM never sees the full catalog, it only generates text, which keeps cost and latency low by construction.

5. Experimental Setup

Models. ALS (`implicit`, factors=100, iterations=30, regularization=0.05, seed=42); FAISS over Sentence-BERT embeddings (`all-MiniLM-L6-v2`), $k_{similar} = 200$; dynamic Ensemble with per-segment weights. **LLM** generators: two small models from distinct providers, `gemini-2.5-flash-lite` (Google) and `gpt-4o-mini` (OpenAI), both with `temperature`=0 (deterministic) and $N = 10$ descriptions per user. **Sampling:** stratified sample of 375 users per segment (1,125 in total), seed 42. **Metrics:** coverage@$\{10, 20, 50\}$ and NDCG@$\{10, 20, 50\}$; end-to-end latency; tokens and monetary cost per user.

Prompt. The LLM receives only the titles of the training history and is asked, in a single turn, for $N = 10$ specific products the customer would plausibly buy next, favouring complementary needs (accessories, matching items, natural next purchases) rather than identical ones, answering with a JSON list of short strings. The prompt is written in Portuguese and carries no catalog, no candidate list and no example items, so the LLM is used strictly zero-shot.

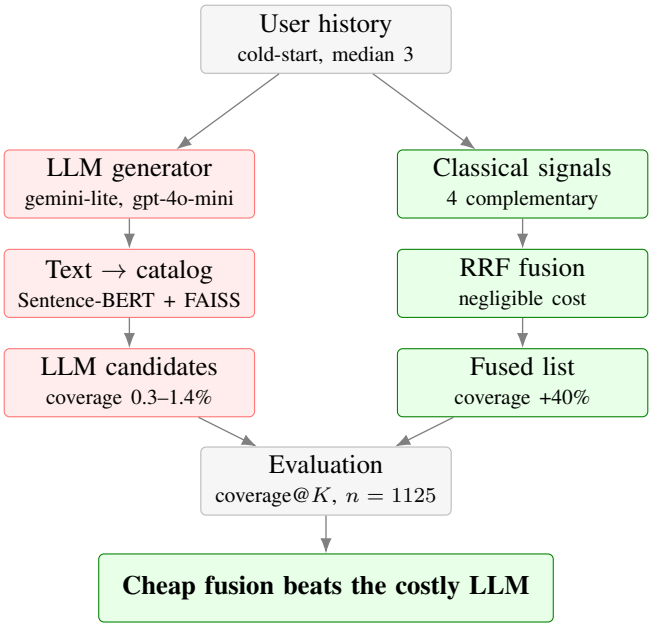


Figure 1. Two strategies evaluated from the same cold-start history: reallocating the LLM budget to candidate generation (left, red) versus a cheap reciprocal rank fusion of classical signals (right, green). Both are scored by coverage@K under leave-one-out.

Splitting. The split is temporal per user: interactions are ordered by `timestamp` and the most recent one becomes the held-out target, the previous ones the training history. Users with a single interaction therefore contribute an empty history and are excluded from the LLM sample, which requires at least one title to prompt with. Preprocessing keeps verified purchases with a non-empty title, maps every product to its `parent_asin` (so colour and size variants collapse to one item), and de-duplicates repeated user-item pairs.

Text-to-item mapping. Each generated description is encoded by the same `all-MiniLM-L6-v2` encoder used for the catalog and mapped to real items by FAISS, aggregating scores across the N descriptions. We validate this channel by querying it with an item's own title: the item is returned in the top-5 in 19 of 20 cases and in the top-10 in 20 of 20. This validates the mapping for *English* queries; Section 6.1 reports what happens when the generated text is in Portuguese, and controls for it.

No leakage. Train/validation isolation is preserved at every stage: the held-out target is excluded from training and from the ALS factors, the co-purchase, lexical, and popularity tables, and the learned-fusion features. The FAISS index stores content (title) embeddings and is queried only with the training history, with already-seen items filtered out. The target belongs to the searchable catalog (it must be retrievable) but is never used as a query or label.

6. Results

Table 3 presents the coverage of LLM generation against classical content-based retrieval (FAISS), for *two* LLM generators from distinct providers (Google and OpenAI), per segment ($n = 375$ each) and aggregated ($n = 1{,}125$). The result is **negative, consistent,**

and robust to the choice of model: *neither* LLM beats FAISS in any segment-K configuration; its coverage is typically one third to one half of the classical baseline, and the two models are practically tied with each other. A paired per-user McNemar test, aggregating the segments, confirms the difference at all cutoffs for both models (gemini-lite: @10 $p = 0.002$, @20 $p = 0.001$, @50 $p < 0.001$; gpt-4o-mini: @10 $p = 0.012$, @20 $p < 0.001$, @50 $p < 0.001$), always with FAISS winning the majority of discordant pairs. On cost, generation consumed on average ≈450/160 input/output tokens and 1.5 to 2.6 s per user, at US$ 0.12 to 0.17 per thousand users: cheap in absolute terms, yet with no return in quality. That two models from distinct providers and training scales fail equally indicates the bottleneck is *structural* (complementary purchasing mediated by embeddings) and not an artifact of a specific LLM.

Table 3. Coverage@K (%) of two small LLM generators, from distinct providers and with `temperature=0`, against classical retrieval (FAISS), per segment ($n = 375$ each) and aggregated ($n = 1{,}125$). No LLM beats FAISS in any configuration; in the aggregate, the LLM vs. FAISS difference is significant at all cutoffs (paired McNemar, $p < 0.02$). In bold, the best of each column.

Segment	Generator	@10	@20	@50
Severe	gemini-lite	0.8	1.3	1.9
	gpt-4o-mini	0.8	1.3	2.7
	FAISS	**1.6**	**3.2**	**4.8**
Moderate	gemini-lite	0.0	0.5	1.1
	gpt-4o-mini	0.3	0.8	1.1
	FAISS	**1.6**	**2.1**	**4.0**
Warm	gemini-lite	0.0	0.3	0.5
	gpt-4o-mini	0.0	0.0	0.5
	FAISS	**1.1**	**1.6**	**2.9**
Aggregate	gemini-lite	0.3	0.7	1.2
	gpt-4o-mini	0.4	0.7	1.4
	FAISS	**1.4**	**2.3**	**3.9**

6.1. Is the negative result an artifact of language?

Our prompt is in Portuguese while the catalog and the encoder (`all-MiniLM-L6-v2`) are English, so a description generated in Portuguese is matched against English titles. This penalty is real and we measure it: on the same 20 items, querying with an item's own English title returns it in the top-5 in 19 of 20 cases, but querying with a faithful Portuguese translation of that same title returns it in only 8 of 20. Roughly half of the mapping's precision is lost in the crossing, which could in principle depress the LLM's coverage for reasons that have nothing to do with the LLM.

The two models let us control for this, because they answered the same prompt in different languages: `gemini-2.5-flash-lite` replied in Portuguese for 95.5% of users, whereas `gpt-4o-mini` replied in English for 94.0% of them, favouring brand-and-model strings. If the language crossing drove the negative result, the English-answering model should have enjoyed a visible advantage. It did not: aggregate coverage is 0.3/0.7/1.2% for gemini-lite against 0.4/0.7/1.4% for gpt-4o-mini at $K = 10/20/50$, a tie.

The sharpest control restricts attention to the 1,058 users for whom `gpt-4o-mini` answered in English, where no crossing occurs at all. There the LLM still reaches only 0.4/0.7/1.3% against FAISS's 1.4/2.4/3.9% on the very same users, and the paired difference remains significant at every cutoff (McNemar $p = 0.019$, 0.0009, 0.0001). Within each model, comparing its own English- and Portuguese-answering users yields no significant difference and the two models disagree on the direction (Fisher exact, $p = 0.11$ and $p = 0.25$), consistent with the effect being small next to the gap that separates the LLM from classical retrieval. We conclude that the language crossing is a genuine limitation of our pipeline, and worth avoiding in replications by prompting in the catalog's language, but that it does not account for the negative result.

7. Cheap Fusion of Classical Signals

The negative LLM result leaves RQ2 open: if neither reordering nor generating recovers coverage, what does? Each classical signal sees the customer through a distinct, *complementary* lens: semantic similarity (FAISS), purchase co-occurrence (co-purchase), lexical overlap of titles (TF-IDF), and global popularity, and all are practically free. We test whether *combining* them covers more target items than any one alone.

Method. We use Reciprocal Rank Fusion (RRF) [Cormack et al. 2009]: for item i, the score is $\sum_\ell 1/(k_0 + r_{\ell,i})$, summing over lists ℓ the reciprocal of its rank $r_{\ell,i}$ in each ($k_0 = 60$). It is a single pass, without training and without a per-user hyperparameter, at negligible cost. Unlike the LLM experiment (sample of 1,125 users), we evaluate the fusion over the 10,133 users with a validation item, across all segments.

Result. Table 4 shows that the RRF fusion of the four signals beats the *best* single signal in every cell, not a weak baseline but the strongest available, generally popularity. At the point where the reranker would operate (top-20) and in the severe segment, coverage rises from 3.4% to 4.6% (+36%, paired McNemar $p < 0.001$); aggregating all users, from 3.2% to 4.5% ($p < 10^{-7}$). NDCG moves in the same direction and by a similar factor (Severe@20: 0.0206 vs. 0.0159), so the fusion is not merely padding the list with the target near the bottom. The gain is significant in severe and moderate cold-start; in the warm segment the direction holds on coverage, but $n = 375$ lacks statistical power, and NDCG@20 there is marginally lower than the best single signal, indicating that when fusion helps a warm user it tends to place the target deeper in the list.

Is there more to extract? We also trained a *learned* fusion (logistic regression over the same signals, with the co-occurrence tables built without the held-out item to avoid leakage). It only *ties* the RRF (Severe@20: 5.0% vs. 4.6%), with sane weights, indicating that the retrievers' ranks do not carry enough information to further separate the target. A fusion *oracle* (union of the same four lists, the ceiling of what these signals could yield if perfectly combined) covers about *twice* the RRF (Severe@20: 8.9% vs. 4.6%; $1.9\times$ aggregated over all users). This gap does not close with more signals or smarter weighting: it corresponds to targets that *no* static signal exposes: the residual of unexpressed intent, which we return to in the discussion.

What about sequential signals? A natural objection is that a *transition* signal would recover part of the jump between subcategories. We tested a fifth signal counting item-to-item transitions observed in training. In isolation it is the *weakest* of all (Severe@20: 1.7%, against 2.9% for FAISS), and including it *degrades* the fusion (RRF-5:

4.3% vs. 4.6% at Severe@20; worse if weighted), because with a median history of 3 there are too many transitions to estimate and too few observations each. The limit is not one of *more signals*, but of information.

Table 4. Best single signal vs. RRF fusion of four signals, by coverage@K (%) and NDCG@K, with paired McNemar p (RRF vs. best single on coverage). Oracle = union of the same four lists (fusion ceiling). Pop = popularity; CoC = co-purchase; Lex = lexical. Bold marks where RRF beats the best single signal; note Warm@20, where fusion retrieves the target more often but places it deeper, so it wins on coverage and loses on NDCG.

		Coverage@K (%)				NDCG@K		
Segment	K	Best single	RRF-4	p	Oracle	Best	RRF-4	Oracle
Severe	20	3.4 (Pop)	**4.6**	<0.001	8.9	0.0159	**0.0206**	0.0429
	50	6.0 (Pop)	**8.1**	<0.001	14.7	0.0210	**0.0274**	0.0542
Moderate	20	2.8 (Pop)	**4.1**	<0.001	7.9	0.0120	**0.0178**	0.0344
	50	5.1 (Pop)	**7.6**	<0.001	13.5	0.0166	**0.0247**	0.0455
Warm	20	2.4 (CoC)	4.0	0.109	6.4	0.0120	0.0112	0.0258
	50	4.3 (Lex)	5.3	0.503	11.2	0.0106	0.0137	0.0354

8. Discussion

Unexpressed intent. The $\approx 2\times$ gap between the RRF fusion and the oracle, together with the learned fusion only tying the RRF, suggests the limit is not one of signal *combination* but of *information*: the static history does not reveal the intent that would lead to the complementary target. This is the reading that unifies the negative and positive findings: what the signals expose is recovered (the fusion), but the rest would require the customer to *say* what they seek, which only a conversational interaction provides, as we take up in Section 10.

Why the LLM failed. The prompt explicitly asked for *specific*, complementary items, yet the generations are generic, modal accessories whose embedding retrieves vague items rather than the specific target. For a history of iPhone cases, the LLM proposes a power bank, a screen protector and AirPods, while the held-out target was a specific USB-C adapter; for an Apple Watch band and a car phone mount, it proposes AirPods and chargers, while the target was a flexible phone tripod. The model captures the *theme* (electronics accessories) but not the specific complementary item the customer actually buys, consistent with the information limit above.

Beyond recommendation. Although we instantiate it in a recommender, the argument is about any *retrieve-then-rerank* architecture, and its shape is not specific to recommendation. Whenever a cheap first stage selects a small candidate set and an expensive second stage reorders it, the second stage inherits a ceiling equal to the first stage's coverage, and no amount of sophistication in the reranker can cross it. What makes the ceiling bite here is a mismatch between the retrieval criterion (semantic similarity) and the relevance criterion (complementarity), and that mismatch is a general failure mode: retrieval-augmented generation systems that retrieve passages similar to the question, rather than passages that *answer* it, face the same structure, as do search pipelines whose recall stage is tuned for topical proximity. The practical lesson transfers: before investing in a better reranker, measure how often the answer is in the candidate set at all, and report that

number alongside end-to-end quality. Our contribution to this broader point is the measurement protocol and the observation that, in one concrete domain, the ceiling binds much earlier than the literature's focus on ranking would suggest.

9. Threats to Validity

Zero-shot LLM versus fitted baselines. Our comparison is informationally asymmetric by design: the classical signals are estimated *from* the Amazon interaction data, whereas the LLMs are used off the shelf, without fine-tuning on interactions and without access to the catalog beyond the text-to-item mapping. This asymmetry is deliberate, because it is the configuration practitioners actually deploy when they add an LLM to an existing pipeline through an API, and the question we ask is one of return on that investment. It does, however, bound the claim: our result establishes that *this zero-shot, API-based generation pipeline* does not outperform cheap data-driven retrieval in our setting. It does *not* establish a general limitation of LLM-based candidate generation, and in particular says nothing about LLMs fine-tuned on interaction data, prompted with catalog-grounded retrieval, or given the collaborative signal in context. Testing whether a fitted or catalog-aware LLM crosses the coverage ceiling is the natural follow-up, and our protocol is the instrument for it.

Generalization. A single Amazon category and two small LLMs (not frontier models); the coverage numbers should not be extrapolated without replication in other domains and at larger model scales.

Coverage vs. conversion. We measure coverage of the held-out item (leave-one-out), a quality ceiling, not conversion; in production, a coverage gain is a necessary but not sufficient condition.

Item cold-start. We evaluate user cold-start; items outside the embedding catalog are not retrievable by any method.

10. Conclusion

We showed that, in e-commerce recommendation dominated by cold-start, the bottleneck is candidate *generation* and not ranking: the target is rarely among the candidates, imposing a ceiling of $\approx$3% on any reranker. We then reallocated the LLM budget from reranking to generating complementary candidates and found a **negative result**: off-the-shelf LLM generation does *not* beat classical content-based retrieval, at nonzero monetary and latency cost, for two LLMs from distinct providers over 1,125 users (paired McNemar, $p < 0.02$ at all cutoffs), and the gap survives controlling for the language of the generated text. Both ways of dropping a general-purpose LLM into the pipeline hit the same limit, cross-subcategory purchasing, which neither content similarity nor the LLM's world knowledge, mediated by embeddings, anticipates. We stress the scope: the LLMs are used zero-shot, so what we bound is the return on the plug-and-play configuration practitioners most often deploy, not the potential of LLM-based generation in general (Section 9).

The prescriptive side is positive: the lost coverage *is* partially recoverable, but by cheap fusion of classical signals rather than by an LLM. RRF over four complementary signals recovers $\approx$40% more coverage than the best single signal ($p < 0.001$, negligible

cost), and a learned fusion only ties it, marking the ceiling of interaction-free methods. What remains up to the oracle reflects the customer's **unexpressed intent**, inaccessible on a static benchmark by construction. This points to *conversational recommendation*, separating what to recommend (the cheap fusion that proved effective here) from how to convey it (the LLM in dialogue, over an already filtered list rather than one generated blindly).

As future work we intend to measure how much of the gap to the oracle a dialogue turn recovers. Evaluating elicitation requires overcoming the very limitation of static benchmarks we exposed here, so we plan a *user simulator* with controlled information asymmetry, in which the user answers at the attribute level without the recommender seeing the target item, avoiding the ground-truth leakage that would invalidate the measurement. This isolates the gain attributable to the conversation, closing the arc between the offline diagnosis presented here and interactive recommendation.

Acknowledgments

The authors would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Declaration of Generative AI in the writing process

During the preparation of this work, the authors used Claude and Gemini in order to improve the language, grammar, and flow of the manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Carraro, D. and Bridge, D. (2024). Enhancing recommendation diversity by re-ranking with large language models. *ACM Transactions on Recommender Systems*.

Cormack, G. V., Clarke, C. L. A., and Büttcher, S. (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proc. ACM SIGIR*, pages 758–759.

Gao, C., Lei, W., He, X., de Rijke, M., and Chua, T.-S. (2021). Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126.

Harte, J., Zorgdrager, W., Louridas, P., Katsifodimos, A., Jannach, D., and Fragkoulis, M. (2023). Leveraging large language models for sequential recommendation. In *Proc. 17th ACM Conference on Recommender Systems (RecSys)*, pages 1096–1102.

Hou, Y., Li, J., Fu, X., He, Z., Yan, A., Chen, X., and McAuley, J. (2026). Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proc. 64th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 3251–3265. Amazon Reviews 2023 dataset.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. *IEEE International Conference on Data Mining (ICDM)*, pages 263–272.

Jannach, D., Manzoor, A., Cai, W., and Chen, L. (2021). A survey on conversational recommender systems. *ACM Computing Surveys*, 54(5):1–36.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Lemdiasova, E. and Zmanovskii, N. (2026). Diagnosing LLM-based rerankers in cold-start recommender systems: Coverage, exposure and practical mitigations. arXiv:2604.16318.

Li, S., Wang, Y., Wang, J., Li, Y., Ghosh, J., and Cocos, A. (2026). LLM reasoning for cold-start item recommendation. In *Proc. ACM Web Conference (WWW)*.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proc. EMNLP-IJCNLP*, pages 3982–3992.

Santos Silva, F., Soares, J. L., Neto, S. F. G., dos Reis, J. C., and Reis, M. S. (2026). ARARA: A LLM-based multi-agent development framework for conversational recommender systems. In *Proc. 14th Int. Workshop on Engineering Multi-Agent Systems (EMAS), co-located with AAMAS*.

Shehmir, S. and Kashef, R. F. (2025). LLM4Rec: A comprehensive survey on the integration of large language models in recommender systems — approaches, applications and challenges. *Future Internet*, 17(6):252.

Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., Wen, Z., Wang, F., Zhao, X., Tang, J., and Li, Q. (2024). Recommender systems in the era of large language models (LLMs). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6889–6907.