To RAG or Not to RAG? Rethinking Multilingual Spelling Correction at Global Scale

Naman Jain[1]

[1]Uber Technologies, Inc., Bangalore, Karnataka, India

Abstract

Query spelling correction is a foundational component of e-commerce search, yet whether retrieval augmentation still improves correction quality in the era of reasoning-capable language models remains unanswered. Recent work combines retrieval-augmented generation (RAG) with large language models to ground LLM predictions in catalog lookups, click logs, and external knowledge bases. We challenge this assumption through a systematic ablation across 30,000 queries spanning five languages (English, French, Japanese, Spanish, and Taiwanese Mandarin).

On English we find that a reasoning model without any retrieval augmentation outperforms every RAG variant we tested, including catalog grounding via Lucene fuzzy matching, brand-model predictions, click/conversion signals, and web-search retrieval. Furthermore, a single language-agnostic prompt generalizes across all five languages with no accuracy trade-off, and even improves English performance over language-specific prompts. Our findings suggest that, for linguistically grounded tasks in quick commerce search, investing in reasoning capability yields greater returns than investing in retrieval infrastructure, with implications for the design of agentic query understanding systems.

We also present an end-to-end set of practical considerations spanning dataset construction, ambiguity-aware evaluation, and deployment, including a hybrid human-plus-agentic-AI annotation workflow, a morphology-aware evaluation stack based on normalization, exact matching, and LLM-as-a-judge assessment, and an offline-first serving design.

Keywords

query spelling correction, e-commerce search, retrieval-augmented generation, large language models, reasoning models, multilingual information retrieval

1. Introduction

This paper studies *spelling correction*: detecting and rewriting misspelled search queries into their intended canonical form. The Uber Eats quick-commerce platform, which spans food delivery, grocery, and convenience retail, serves hundreds of millions of search queries daily across diverse global markets. The search vocabulary on these platforms differs substantially from general web or traditional product search. Users query for dish names (*pad thai*, *tonkatsu ramen*), ingredient terms (*cilantro*, *gochujang*), brand names (*Haagen-Dazs*, *Dr Pepper*), and category expressions (*boba near me*, *healthy lunch*), often mixing transliterations, abbreviations, and colloquial spellings. Misspelled queries account for approximately 15–20% of search traffic and directly reduce retrieval recall: catalog items are indexed under canonical spellings that do not match corrupted inputs, which makes accurate spelling correction a prerequisite for effective retrieval.

Industry has addressed this challenge through progressively complex architectures. Early systems relied on edit-distance algorithms such as SymSpell [1] and Levenshtein automata [2]. Subsequent generations added phonetic matching, including Soundex [3], Metaphone [4], and Double Metaphone [5], alongside keyboard-proximity heuristics.

Recent work extends this retrieval-centric paradigm by incorporating large language models with retrieval-augmented generation [6]. Guo et al. [7] combine BM25 and ColBERT retrieval from product catalogs with a fine-tuned Mistral-7B model, achieving 71.0 F1 on Amazon product search queries.

The author thanks Robert Saliba (Director of Engineering), Felix Fang (Staff Machine Learning Engineer), and Yuanhua Lv (Principal Machine Learning Engineer) for their thoughtful review of drafts of this paper.

ECOM'26: SIGIR Workshop on eCommerce, Jul 24, 2026, Melbourne, Australia

namajain@uber.com (N. Jain); nammanjain@gmail.com

Choudhury [8] describes a production system at Zepto using Llama-3 with vector database RAG for grocery and convenience search, reporting a 7.5% conversion lift. The foundational architectural premise of these systems is the necessity of retrieval augmentation to provide the domain-specific grounding required for LLM-based corrections.

This paradigm is mirrored in our baseline production pipeline, which identifies 1,000 potential candidates through Lucene fuzzy matching and BM25 ranking. This initial set is refined to 20 candidates using phonetic and keyboard-distance heuristics before a final correction is determined by an LLM judge.

We challenge this assumption. With the advent of reasoning models, that is, LLMs that employ extended chain-of-thought computation before producing outputs [9], we hypothesize that retrieval augmentation may no longer provide marginal value for tasks where the required domain knowledge is well represented in pretraining data. To test this hypothesis, we conduct a systematic ablation across four RAG strategies (brand catalog lookup, click/conversion signals, Lucene fuzzy matching, and web search), two model families (o3-medium, gpt-5.2), and five languages.

Our investigation addresses three research questions:

RQ1. To what extent does retrieval augmentation enhance spelling correction when utilizing a high-capacity reasoning model?

RQ2. Can a unified prompt maintain accuracy while generalizing across linguistically diverse languages?

RQ3. What practical design choices for data gathering, evaluation, and deployment best support production-ready generative spelling correction systems?

Contributions. This paper offers the following contributions.

(i) Evidence that reasoning-centric LLMs surpass RAG-based pipelines in query correction. Through a systematic ablation involving web search results, click/conversion data, and brand entities, we establish that optimal accuracy is reached without external retrieval context. On English benchmarks, the inclusion of brand RAG, click/conversion RAG, and web search RAG each results in accuracy declines. We explain mechanistically why each retrieval source fails.

(ii) A single unified prompt generalizes across five languages with minimal per-language tuning. The same prompt achieves 85–87% accuracy across English, French, Japanese, Spanish, and Taiwanese Mandarin, with $< 10\%$ of the prompt containing language-specific instructions. This prompt also improves English performance relative to English-specific prompts, reducing the historically expensive process of deploying spell correction for each new language.

(iii) A practical framework for building production-ready generative query correction systems. We present an end-to-end set of practical considerations spanning dataset construction, ambiguity-aware evaluation, and deployment. Our approach combines a hybrid human-plus-agentic-AI annotation workflow, a morphology-aware evaluation stack based on normalization, exact matching, and LLM-as-a-judge assessment, and an offline-first serving design that precomputes and persistently caches corrections for head and torso queries while routing the uncached long tail to lightweight online models. Together, these choices make reasoning-centric correction both measurable and operationally viable at quick-commerce scale.

2. Related Work

Classical Spell Correction. The noisy-channel framework [10, 11] underpins traditional spell correction. Edit-distance methods such as SymSpell [1] and phonetic algorithms like Soundex [3], Metaphone [4], and Double Metaphone [5] enable efficient matching of misspellings. These methods

are effective but rely on language-specific rules, limiting scalability. There is also work leveraging query logs [12], reformulation modeling [13], statistical machine translation [14], and noisy-channel heuristics popularized for the web by Norvig [15].

Neural Methods. Neural approaches include sequence-to-sequence and BERT-based models [16]. ByT5 [17] operates at the byte level, enabling strong multilingual performance, while Zhang et al. [18] use multi-teacher distillation for cross-lingual correction. These approaches typically require large training datasets and per-language tuning.

Retrieval-Augmented Spell Correction. Guo et al. [7] show that RAG significantly improves correction for smaller models using catalog retrieval and fine-tuning. Choudhury [8] reports similar gains in production using Llama3-8B with catalog retrieval. Both rely on retrieval to compensate for limited model capacity. In contrast, we evaluate reasoning-class models without fine-tuning and show that these gains do not transfer.

LLM-Based Evaluation. Zheng et al. [19] establish the LLM-as-a-judge paradigm with MT-Bench, demonstrating strong agreement between LLM judgments and human preferences for evaluating open-ended generation. We adapt this methodology for spell correction evaluation, where exact-match scoring systematically underestimates performance because multiple morphological variants may be equally valid corrections. Prompt-engineering techniques [20] also inform our universal-prompt design.

3. Problem Definition

3.1. Task Formulation

We conceptualize query spelling correction as a conditional generation objective. Given an input search query q, the system is tasked with producing a refined output q' such that:

- $q' = q$ when the query is orthographically correct (pass-through), or
- q' represents the most probable intended string when q contains corruption or noise.

This framework requires the model to simultaneously execute two sub-tasks: (1) *error detection*, that is, determining whether a correction is warranted, and (2) *error correction*, that is, generating the canonical form. Rather than bifurcating these into discrete pipeline stages, we treat them as a unified generative process.

3.2. Challenges in Quick Commerce Search

Query refinement within the quick commerce ecosystem presents unique complexities that diverge from standard web search or conventional product discovery.

Domain-specific vocabulary. Quick commerce inventories comprise a specialized lexicon of culinary terms, ingredients, and brand entities often absent from standard linguistic corpora. Lexical items such as *tonkatsu*, *gochujang*, *acai*, and *boba* constitute valid intents that traditional spell checkers would erroneously flag as anomalies.

Brand name ambiguity. Proprietary names frequently mirror the orthography of common nouns (e.g., “Lyft” vs. “lift,” “Cheez-It” vs. “cheese”) or employ non-standard spellings for differentiation. Accurate processing requires distinguishing these intentional variations from genuine errors [7].

Multilingual and transliterated queries. Users in global markets frequently oscillate between native scripts, English, and hybrid transliterations. In many markets, the catalog is indexed in the local language or native script, while users may issue the same intent in English or romanized form. For example, a Japanese catalog may index an item under its native-script form while the user searches for *ramen* or *sushi*; similarly, a Taiwanese Mandarin catalog may contain local-script item names while users search with English or romanized approximations. Error distributions therefore vary by language: French queries often involve diacritical omissions, Japanese requires matching across romaji and native-script forms, including cases where users do not consistently use katakana, kanji, and hiragana for the same item, and Taiwanese Mandarin involves both script and input-method variation, including simplified Chinese, traditional Chinese, and phonetic input through Bopomofo (Zhuyin). These inputs can also produce errors such as Zhuyin keystrokes selecting the wrong homophonic character or partially uncommitted Zhuyin sequences leaking into the final query as raw phonetic symbols.

High precision requirements. With approximately 80% of queries being orthographically sound, over-correction (the unintended modification of a correct query) significantly degrades the user experience. The architecture must preserve high precision for pass-through strings while maintaining robust recall for genuine corruptions.

Morphological variations. A single misspelling often admits several surface forms that all preserve the user's intent. "Laays," for example, can reasonably be normalized to either the brand "Lay's" or the lowercase plural "lays"; "ice cream" and "icecream" refer to the same item; and "cafe" and "café" are typically interchangeable in retrieval. Treating any one of these as the sole correct answer penalizes models that produce an equally valid alternative, so the evaluation framework must recognize these morphological variants as equivalent rather than as errors.

4. Dataset and Evaluation Framework

4.1. Golden Dataset Construction

We construct a golden dataset of 10,000 English queries and 5,000 queries each in French, Japanese, Spanish, and Taiwanese Mandarin, for a total of 30,000 queries. Queries are randomly sampled from live search traffic and include diverse error types: phonetic substitutions (e.g., "blutut" for "bluetooth"), keyboard-proximity errors, brand-name misspellings, transliterations, diacritical omissions, and character-level input-method errors.

English Annotation Protocol. We constructed the English dataset using a three-annotator framework following a $2L + 1J$ design. Two independent human labelers (L_1 and L_2) first annotated each query using shared guidelines, with initial calibration rounds to align task understanding. In cases of disagreement, a third human annotator (J) acted as a judge to produce the final label, yielding a single adjudicated dataset.

In the second stage, we benchmarked several LLM-based annotation setups against this human-labeled dataset. To evaluate annotation quality, we conducted a manual review of 200 queries in collaboration with the product team, comparing human and LLM annotations side by side. The best-performing configuration, GPT-4O-SEARCH-PREVIEW (an agentic model with web search access and the same guidelines as human annotators), achieved 86.5% accuracy on this sample, while the adjudicated human annotations achieved 77.5%. This corresponds to a $+9.0$ percentage point improvement, driven by more consistent handling of ambiguous cases and reduced susceptibility to knowledge gaps, such as correctly resolving "sendy's" to "wendy's."

In the final stage, we scaled annotation to the full dataset using a second $2L + 1J$ framework. Here, one "labeler" was the adjudicated output from the initial human $2L + 1J$ process, and the second labeler was the LLM. In cases of disagreement between these two, an expert human judge provided the final

decision. This hybrid human-plus-LLM adjudication pipeline produces a high-fidelity dataset while minimizing the need for repeated multi-round manual labeling and significantly reducing annotation cost and turnaround time.

International Annotation. Extending the full English annotation pipeline to international markets presents practical challenges, including limited availability of qualified annotators for non-English e-commerce domains, lower baseline annotation quality (often below 70% for complex scripts such as Japanese and Taiwanese Mandarin), and significantly longer turnaround times of typically one to two weeks per labeling batch.

To address these constraints, we adapt the framework by replacing one human labeler in the multi-rater setup with an LLM, effectively implementing a hybrid $2L + 1J$ scheme with one human annotator and one LLM acting as labelers. Disagreements are resolved by a judge, preserving the structure of multi-rater adjudication while substantially reducing latency from weeks to days.

In addition, we introduce an LLM-based consensus score as a quality control mechanism. For each annotator, we compute the fraction of labels that match the LLM outputs. This per-annotator score is then compared against the average consensus across all annotators. Annotators whose agreement with the LLM falls significantly below the cohort average are flagged for review. This process surfaced several low-quality annotators whose outputs introduced systematic errors, particularly in Taiwanese Mandarin, where recognizing Bopomofo input-method errors requires domain-specific expertise.

4.2. Evaluation Methodology

Generative spell correction admits multiple valid outputs. Exact-match evaluation systematically underestimates performance and can lead to incorrect architectural conclusions by penalizing models that produce valid but non-identical corrections. We implement two evaluation protocols.

Programmatic normalization (English). Lemmatization, special character and accent removal, and whitespace normalization, followed by exact string matching. This handles cases such as "lay's" vs. "lays" without requiring LLM inference at evaluation time.

LLM-as-a-judge (Non-English). The judge evaluates whether the model's output is semantically and intentionally equivalent to the ground truth correction. This is particularly important for languages where programmatic normalization rules are brittle or nonexistent (e.g., CJK scripts where morphological variation is expressed through different characters rather than inflectional suffixes).

A natural concern is whether an LLM judge exhibits systematic biases that mirror the evaluated model's own tendencies, inflating perceived performance. The judge model is therefore distinct from the correction model to reduce self-evaluation bias. We validate the LLM-as-a-judge protocol by comparing its assessments against human judgments on a held-out calibration subset [19]. This validation confirms that the judge reliably identifies semantically equivalent corrections that differ only in surface form (e.g., singular vs. plural, presence vs. absence of special characters and filler words).

We note that LLM-as-a-judge evaluation is not noise-free: residual disagreement with human judgment introduces measurement variance into the non-English numbers. We expect this noise to be small relative to the accuracy gaps we report between configurations, so it is unlikely to alter the qualitative conclusions; nonetheless, the absolute scores should be read with this caveat in mind, and a fuller quantification of judge noise is left to future work.

5. Experimental Setup

All experiments use OpenAI o3 and GPT-5.2 as the primary reasoning models, evaluated across multiple reasoning-effort settings. These models are trained with large-scale reinforcement learning on chains of thought [9]. We additionally include GPT-4.1 and GPT-5.2 (reasoning disabled) as non-reasoning

baselines, operating with default decoding parameters (temperature, top-p, and max tokens) to reflect standard generation behavior without explicit reasoning optimization. All models are used without task-specific fine-tuning. Prompts are hand-crafted through iterative refinement by domain experts, with a focus on enforcing high precision, minimizing over-correction, and preserving user intent across multilingual inputs.

5.1. RAG Configurations

The study examines three RAG strategies, each contributing a unique form of domain knowledge. These augment the production model's existing lexical retrieval by providing candidate corrections grounded in actual catalog entries, similar to the BM25 retrieval methodology used by Guo et al. [7]. We opted for fuzzy BM25 over an Approximate Nearest Neighbor (ANN) approach for the catalog due to several considerations: (i) the primary focus of this ablation was to compare *knowledge sources* rather than retrieval mechanisms themselves; (ii) given our existing complex Lucene pipeline, the additional effort required to maintain a separate ANN architecture was not justified by the anticipated incremental value; (iii) ANN retrieval is susceptible to the same catalog noise and availability issues as lexical methods, particularly since catalogs often contain numerous misspelled items that weaken their reliability as a ground truth for error detection.

Beyond standard catalog lookups, we evaluated the following industry techniques for providing grounding context to LLM-driven spelling correction:

Brand Prediction RAG. Integrates predicted brand entities directly into the correction prompt. These entities are obtained from our most effective independent LLM-based brand prediction model.

Behavioral RAG (Click/Conversion). Context is provided by extracting the five stores and products most frequently clicked in historical behavioral logs associated with a given query.

Web-Augmented RAG. We use an agentic formulation via OpenAI's Responses API, providing the LLM access to a web search tool which it can call when needed. This supplies external knowledge beyond the platform's own catalog, potentially covering terms and brands not in the internal index.

In all RAG configurations, retrieval is keyed on the original user query as typed, including its misspellings, rather than on a pre-corrected form. This mirrors the production setting, where the correction is precisely what we are trying to produce and is therefore not yet available at retrieval time. An alternative design that first generates high-probability corrections and retrieves on those is possible, but it presupposes a correction step and is out of scope for this study.

To ensure a fair comparison, each RAG configuration is evaluated on an English subset where outputs from our top-performing brand model are available.

5.2. Prompt Strategies

We evaluate multiple prompt variants:

- **Language-specific prompts (English v1, v2):** iteratively refined for English, incorporating progressively more explicit correction rules and formatting constraints.
- **International prompts (i18n v1, v2):** designed for non-English languages with emphasis on precision over recall.
- **Universal prompt:** a single prompt combining the strongest elements of both English and international prompt designs, with language-agnostic instructions focused on intent preservation and minimal language-specific rules for output normalization.

Each prompt begins with task context that constrains all inputs to food, grocery, restaurant, and retail queries, ensuring domain-relevant interpretations; it then outlines key failure modes, explicitly

Table 1
RAG ablation on the English evaluation subset (historical p_{95} queries). Every RAG configuration reduces accuracy.

Model Configuration	Accuracy	Precision	Recall
GPT-5.2 medium (Brand RAG)	89.92%	84.91%	79.42%
GPT-5.2 medium (Click RAG)	89.18%	85.21%	76.22%
GPT-5.2 medium (Web RAG)	89.44%	82.35%	81.70%
GPT-5.2 medium (No RAG)	**90.63%**	83.83%	82.83%
o3 medium (Brand RAG)	91.07%	84.19%	85.83%
o3 medium (Click RAG)	90.44%	83.93%	84.28%
o3 medium (Web RAG)	90.18%	82.43%	85.32%
o3 medium (No RAG)	**91.11%**	84.40%	85.83%

Table 2
Comparison of prompt, model, and reasoning strategies on the complete English golden dataset.

Model	Prompt	Accuracy	Precision	Recall
GPT-4.1	prompt_en_1	75.77	61.43	67.67
GPT-5.2	prompt_en_1	77.43	67.33	63.70
GPT-5.2_xhigh	prompt_en_1	80.61	71.28	70.49
GPT-5.2_medium	prompt_en_1	81.51	75.49	70.00
o3_high	prompt_en_1	83.77	77.52	75.83
o3_low	prompt_en_1	84.27	79.07	75.43
o3_medium	prompt_en_1	84.52	78.52	76.93
o3_medium	prompt_en_2	85.08	81.06	76.44
o3_medium	prompt_universal	**85.61**	**83.56**	76.71

separating overcorrection, undercorrection, and proper-noun handling, along with language-specific rules for handling accents, phonetics, scripts, and transliteration. Detailed decision guidelines specify how to assign error presence, correction types, reasoning categories, and confidence levels. Finally, curated examples and edge cases provide calibration signals, illustrating correct handling of brand names, abbreviations, ambiguous queries, and multilingual inputs.

6. Results

6.1. RAG Ablation

Table 1 presents the central result of this study. We evaluate o3 and GPT-5.2 at medium reasoning effort with and without each retrieval source.

Every RAG configuration reduces accuracy. Brand RAG causes a marginal decline; click/conversion RAG and web search RAG introduce progressively larger degradations.

6.2. Prompt, Model, and Reasoning Comparison

Table 2 presents results across prompt, model, and reasoning variants on the full English golden dataset (10K queries) using different models and reasoning efforts.

The universal prompt demonstrates superior performance with an English accuracy of 85.61% and precision of 83.56%, surpassing specialized English prompts. Combined with the cross-lingual evaluation below, this directly addresses RQ2: not only does the universal prompt generalize across languages, it actually improves English performance.

Comparing o3_low (84.27%), o3_medium (84.52%), and o3_high (83.77%) on the same prompt (`prompt_en_1`), the key takeaway is that *having some reasoning is important, but increasing reasoning beyond that does not help and can even hurt.* Medium and low reasoning perform similarly, while high

Table 3
Cross-lingual performance with multiple prompt iterations. Numbers are Accuracy/Precision (%). Minimal language-specific engineering is applied.

Model	Prompt	Web Search	French	Japanese	Spanish	Taiwanese
o3_med	i18n_1	True	85.74/83.78	86.11/73.25	85.43/79.11	86.03/74.91
o3_med	i18n_1	False	84.44/79.91	85.13/72.62	84.36/77.50	82.05/62.50
o3_med	i18n_2	True	86.48/87.32	87.08/75.40	86.63/84.77	84.76/76.64
o3_med	i18n_2	False	85.68/87.50	86.99/75.85	87.17/85.22	83.23/75.25
o3_med	universal	True	86.05/90.26	86.99/80.56	83.29/82.07	83.57/72.69
o3_med	universal	False	84.07/87.25	87.96/81.88	84.36/84.49	83.49/76.29

reasoning slightly degrades accuracy, suggesting that once a baseline level of reasoning is present, additional computation introduces over-analysis rather than gains.

The model may correct to unrelated intent when given too much reasoning budget. One such example is when medium reasoning tried to correct "press hingham" and decided not to correct it, as both words are individually correct even though they seem odd together. High reasoning went a step ahead to consider search queries like "fresh ginger," "frozen ham," "pressed ham," and "pre-sliced ham" before finally correcting to "pressed hingham." Similarly, high reasoning incorrectly corrects *michmichi* to *kimchi* and *don't nos* to *donuts*.

This pattern is consistent across models. In the GPT-5.2 family, GPT-5.2_medium (81.51%) outperforms both the no-reasoning configuration (77.43%) and GPT-5.2_xhigh (80.61%), reinforcing that adding reasoning improves performance relative to no or minimal reasoning, but more reasoning does not translate to better outcomes. Overall, the results suggest that the *presence* of structured reasoning is more important than the *degree* of reasoning effort, with diminishing and sometimes negative returns beyond a moderate level.

6.3. Cross-Lingual Evaluation

Table 3 reports cross-lingual results for the universal prompt with o3 at medium reasoning effort. The setup uses one shared core prompt across languages, with only minimal ($< 10\%$ of overall prompt) language-specific guidance for normalization and guardrails. We do not translate the prompt or introduce separate retrieval logic per language at inference time.

Performance is stable across all languages. Under the universal prompt, accuracy remains tightly clustered, ranging from 83.29% to 87.96% across the evaluated language and web-search settings. More broadly, the results suggest that a largely shared prompt is sufficient for robust multilingual query correction in this task, and that only light language-specific adaptation is needed to handle script and orthographic differences.

Web search produces only modest and inconsistent effects, with small gains in some settings and regressions in others. This indicates that, for multilingual spell correction, the main performance driver is the reasoning-capable model together with a well-designed shared prompt, rather than heavy per-language engineering or retrieval augmentation.

6.4. Production Comparison

Table 4 compares the reasoning-only approach against the production Lucene RAG pipeline (Lucene fuzzy matching + keyboard/phonetic distance rankers + LLM judge) on the English evaluation set.

The production pipeline's accuracy of 72.76% indicates that only about three in four queries receive the correct output, that is, a correct pass-through for clean queries or a correct rewrite for misspelled ones. The reasoning model raises overall accuracy to 85.61%, reducing the end-to-end error rate from roughly 27% to 14%, which directly translates into fewer broken search sessions and more relevant retrieval results downstream.

Table 4
Production comparison on the English evaluation set. The reasoning-only model substantially outperforms the Lucene RAG pipeline across all metrics.

System	Accuracy	Precision	Recall	F1
Production pipeline	72.76%	58.97%	60.74%	59.84%
o3 + universal prompt	85.61%	83.56%	76.45%	79.85%
Delta	+12.85 pp	+24.59 pp	+15.71 pp	+20.01 pp
Relative lift	+17.66%	+41.70%	+25.86%	+33.43%

This production pipeline also serves as our practical baseline for classical techniques: its candidate generation combines Lucene fuzzy matching with keyboard-distance and phonetic rankers in the same family as edit-distance and phonetic methods such as SymSpell [1] and Metaphone [4]. Purely classical correctors and older neural sequence models are expected to underperform the reasoning-only approach by at least the margin shown here, since they lack its parametric domain knowledge; a direct head-to-head benchmark against standalone implementations of these systems would further quantify the gap and is left to future work.

7. Analysis and Discussion

7.1. Why Retrieval Does Not Help Reasoning Models

The central finding is that none of the RAG configurations improve over a reasoning-only baseline. In practice, retrieval not only fails to help but often introduces systematic errors. The production pipeline illustrates this clearly: it relies on Lucene fuzzy matching over the product catalog, which over-corrects queries when the intended item is absent. In such cases, the candidate set is constrained to whatever exists in the catalog, even if it is semantically unrelated. The LLM is then anchored to these irrelevant candidates. For example, "cat showr" is corrected to "cat chow" because the catalog contains *Cat Chow* but not "cat shower." We attempt to mitigate this through prompt tuning, instructing the model to fall back on its own judgement and ignore the retrieved candidates when none of them appear to be a good fit; in practice, however, the model still defaults to the provided candidates more often than not, and the anchoring effect persists. A reasoning model, operating without this constraint, instead recovers "cat shower" by relying on learned linguistic patterns, avoiding catalog-induced bias.

This pattern holds across all retrieval sources we evaluated. Each fails for a distinct but predictable reason.

Catalog grounding introduces bias from noise and incomplete coverage. Product catalogs contain misspellings, inconsistent naming, and gaps in coverage. As a result, the candidate set may exclude the correct term or include incorrect variants. This leads to error reinforcement when catalog misspellings are treated as valid, and coverage gaps when the intended item is missing, pushing the model toward the closest but wrong alternative.

Click and conversion logs encode failure rather than intent. Misspelled queries often lead to poor or empty results, so user interactions are sparse or reflect frustration-driven clicks. This produces weak or misleading supervision signals, leading to a measurable drop in accuracy.

Brand RAG is redundant. Brand prediction systems already rely on LLM-based retrieval over brand catalogs. Passing these predictions downstream adds no new information, as the reasoning model can independently recover the same brand knowledge from its parameters.

Web search introduces noise and cost. External results often surface irrelevant or overly specific matches to misspelled strings, which distract from the correct interpretation. While there is a small gain in some international settings, it comes at significant latency and monetary cost, making it impractical for production.

These results challenge the assumption that retrieval augmentation is necessary for domain-specific NLP tasks. The gap with prior work such as Guo et al. [7] is largely explained by model capacity. Smaller models benefit from retrieval because they lack sufficient parametric knowledge. In contrast, modern reasoning models already encode extensive knowledge of brand names, food terminology, and orthographic conventions, making retrieval largely redundant.

More fundamentally, spell correction is not a retrieval problem. It is a linguistic reasoning task centered on local edits. Correcting "chiken tikka" to "chicken tikka" requires understanding character-level error patterns, phonetic similarity, and word structure. Retrieved candidates may provide possible targets, but they do not help determine whether a correction is needed or how to transform the input. Reasoning models handle this directly by evaluating multiple hypotheses about the input and selecting the most plausible form.

Finally, retrieval-based approaches face a structural limitation: they depend on the very signal that is corrupted. The misspelled query must be used to retrieve candidates, but misspellings degrade retrieval quality across all methods. Fuzzy matching and ANN increase false positives, click logs are sparse or biased, and web search may reinforce the incorrect form. We tried extensive prompt tuning to mitigate this, but noisy input remains the main source of precision drop. Reasoning models avoid this circularity by relying on internal knowledge rather than external signals derived from noisy input. In sum, spell correction aligns closely with the strengths of large reasoning models: implicit knowledge, robust handling of noisy inputs, and the ability to perform fine-grained linguistic inference. Retrieval, instead of grounding the task, often constrains it in ways that degrade performance.

7.2. Why Unified Prompts Outperform Specialized Ones

The finding that a universal prompt outperforms an English-specific prompt (85.61% vs. 85.08%) is counterintuitive and points to the role of cross-lingual transfer. By not assuming a fixed language, the universal prompt encourages the model to consider a broader hypothesis space, which benefits even English queries. This is particularly important in our domain, where queries frequently include foreign-origin terms such as "ramen," "croissant," or "taco," or exhibit mixed-language patterns. A language-specific prompt can implicitly bias the model toward strictly English interpretations, whereas a universal prompt better aligns with the inherently multilingual nature of user input, leading to more accurate corrections. This finding has practical implications: maintaining a single prompt rather than n language-specific prompts substantially reduces engineering overhead and eliminates a source of prompt-engineering errors.

7.3. Practical Considerations for System Architecture

Offline-first strategy driven by query distribution. Query traffic follows a steep power law: head and torso queries account for roughly 95% of volume. These queries are also often repeated; we set a query-repeat-rate threshold with a one-month lookback to determine which percentile of queries to target. This makes offline computation the cornerstone of system design. High-quality corrections for frequent queries can be precomputed using a reasoning model and cached persistently, converting expensive inference into a one-time cost. At serving time, most queries are resolved via low-latency lookups, with only the long tail requiring online handling.

Cost and stability advantages over retrieval systems. The offline-first approach enables persistent reuse of corrections for high-frequency queries, making reasoning economically viable. In contrast, retrieval-based systems depend on dynamic sources such as product catalogs, requiring continuous

re-computation and limiting cacheability. This increases both operational cost and system complexity over time.

Global scalability. A universal prompt, combined with offline processing, enables rapid expansion across markets. New languages can be supported by running the same pipeline on localized query logs, without requiring language-specific prompt engineering or retrieval infrastructure.

Online serving for the long tail. For tail traffic which does not repeat often, offline computation is not feasible and invoking a large reasoning model is impractical due to latency and cost. Instead, a distilled seq2seq model with sub-20 ms latency provides a better trade-off. This model handles uncached and long-tail queries efficiently, while lightweight heuristics or classifiers gate invocation by detecting whether a query likely contains an error. This ensures that only a small fraction of queries incur model cost, keeping latency within production constraints.

7.4. Generalizability and Limitations

Several factors influence the scope and applicability of our findings. Our evaluation is centered on high-volume and recurring head and torso queries, covering roughly the 95th percentile of total traffic in a food delivery and quick commerce domain. In this traffic slice, vocabulary such as cuisine names, restaurant brands, and ingredient terms is well represented in pretraining data, which likely contributes to the strong performance of reasoning-only approaches. However, long-tail queries involving rare or newly emerging brand names, novel slang, or heavily code-mixed input may still benefit from retrieval, particularly in low-resource languages.

The extent to which these results transfer across domains depends on vocabulary coverage and error characteristics. Domains with highly specialized or rapidly evolving terminology, such as niche electronics or pharmaceuticals, may see greater gains from retrieval. While we evaluate several common RAG strategies, including catalog-based, click-based, and web-search augmentation, this is not an exhaustive exploration. Other knowledge sources or hybrid retrieval architectures may yield different outcomes. That said, the core failure modes we observe, namely redundancy with parametric knowledge, noise from weak behavioral signals, and distraction from irrelevant context, are likely to generalize to other settings where spelling errors are primarily orthographic or phonetic.

Regarding the mitigation of retrieval noise, we explored prompt tuning as a solution, though more sophisticated techniques like Retrieval-Aware Fine-Tuning (RAFT) [21] exist. Applying RAFT to the reasoning models we study is non-trivial for two reasons. First, proprietary reasoning models from providers like OpenAI do not currently support fine-tuning, so implementing RAFT would require self-hosting. Second, reasoning capability in these models is induced by large-scale reinforcement learning on chain-of-thought [9]; RAFT-style supervised fine-tuning on retrieval-conditioned labels targets a different objective and can clash with this reasoning training, blunting the very chain-of-thought behavior we rely on. Transitioning to self-hosted, smaller 7B-class models would further result in a loss of world knowledge and parametric depth. Consequently, both RAFT implementation and the hosting of larger-scale models were excluded from this analysis due to resource investment constraints and this potential objective mismatch.

There are also important methodological and system-level limitations. First, our use of LLM-as-a-judge, while validated against human annotations, may introduce bias, especially in low-resource languages where both the correction and evaluation models have weaker representations. Second, reasoning models are evolving rapidly, and our results reflect the capabilities at the time of evaluation. While the absolute performance may change, we expect the broader trend, that stronger reasoning reduces reliance on retrieval, to remain consistent. Finally, we do not conduct a detailed latency analysis due to our offline caching strategy. Reasoning models typically incur higher computational and latency cost, which may pose challenges for real-time systems if not mitigated through strategies such as caching or model distillation.

8. Conclusion

This paper examines a central design question for query spelling correction in quick commerce search: does retrieval augmentation still add value when the correction model is already a strong reasoning-class LLM? Across a systematic ablation over brand retrieval, behavioral signals, catalog fuzzy matching, and web search, we find that the answer is generally no. On English queries, every RAG variant either underperforms or matches the reasoning-only baseline, with failures that are explainable in mechanistic terms. Catalog retrieval is constrained by incomplete and noisy inventory coverage, behavioral signals are weak proxies for user intent, brand retrieval is largely redundant with parametric knowledge, and web search introduces both noise and operational cost.

A second finding is that a single universal prompt is sufficient across five typologically diverse languages. Rather than requiring separate prompt engineering for each locale, the shared prompt maintains strong performance in English, French, Japanese, Spanish, and Taiwanese Mandarin, while even improving English accuracy relative to language-specific prompts. This suggests that multilingual generalization, when paired with a reasoning-capable model, can simplify deployment without sacrificing quality.

Our third contribution is an evaluation framework for generative query correction. By combining programmatic normalization, exact matching, and LLM-as-a-judge evaluation [19], together with a hybrid human and LLM annotation workflow, we address the core challenge that multiple surface forms can represent equally valid corrections. This makes evaluation more faithful to the task and more scalable than strict string matching alone.

Taken together, these results suggest a shift in the architecture of query correction systems. For tasks dominated by local orthographic and phonetic reasoning, stronger model reasoning appears to provide more value than additional retrieval infrastructure. In our production comparison, the reasoning-only approach substantially outperforms the Lucene RAG pipeline across accuracy, precision, recall, and F1, showing that simplification can improve both quality and system complexity. More broadly, our findings indicate that query correction may be approaching a new design point: not retrieval-first, but reasoning-first, with retrieval reserved only for the long tail of genuinely novel or out-of-distribution entities.

Acknowledgments

The author thanks Robert Saliba (Director of Engineering, `rsaliba@uber.com`), Felix Fang (Staff Machine Learning Engineer, `felixfang@uber.com`), and Yuanhua Lv (Principal Machine Learning Engineer, `yuanhua@uber.com`) for their thoughtful review of drafts of this paper.

Declaration on Generative AI

During the preparation of this work, the author used GPT-class large language models in order to: paraphrase and reword authored passages, improve writing flow and style, and perform grammar and spelling checks. The initial content was drafted by the author; these tools were not used to generate original text. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication's content. The LLM systems studied as the subject of this work are described in the Experimental Setup section and are distinct from any tools used to prepare the manuscript.

References

[1] W. Garbe, SymSpell: Symmetric delete spelling correction algorithm, https://github.com/wolfgarbe/SymSpell, 2012.

[2] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady 10 (1966) 707–710.
[3] M. K. Odell, R. C. Russell, Soundex coding system, U.S. Patent 1,261,167, 1918.
[4] L. Philips, Hanging on the metaphone, Computer Language 7 (1990) 39–43.
[5] L. Philips, The double metaphone search algorithm, C/C++ Users Journal 18 (2000) 38–43.
[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 9459–9474.
[7] X. Guo, R. Patki, D. Everaert, C. Potts, Retrieval augmented spelling correction for e-commerce applications, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP), 2024.
[8] K. D. Choudhury, Lost in translation? how we fix misspelled multilingual queries with LLMs, Zepto TechXPress, 2025.
[9] OpenAI, o3 system card, https://openai.com/research/o3-system-card, 2025.
[10] M. D. Kernighan, K. W. Church, W. A. Gale, A spelling correction program based on a noisy channel model, in: Proceedings of the 13th Conference on Computational Linguistics (COLING), 1990, pp. 205–210.
[11] E. Brill, R. C. Moore, An improved error model for noisy channel spelling correction, in: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL), 2000, pp. 286–293.
[12] Y. Duan, B.-J. P. Hsu, Online spelling correction for query completion, in: Proceedings of the 20th International Conference on World Wide Web (WWW), 2011, pp. 117–126.
[13] Y. Li, S. Liu, P. Li, Spelling correction for query reformulation in e-commerce search, in: SIGIR Workshop on E-commerce (SIGIR eCom), 2021.
[14] S. A. Hasan, O. Farri, A. J. Yepes, Spelling correction of user search queries through statistical machine translation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 451–460.
[15] P. Norvig, How to write a spelling corrector, https://norvig.com/spell-correct.html, 2007.
[16] S. M. Jayanthi, D. Pruthi, G. Neubig, NeuSpell: A neural spelling correction toolkit, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), 2020, pp. 158–164.
[17] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, C. Raffel, ByT5: Towards a token-free future with pre-trained byte-to-byte models, Transactions of the Association for Computational Linguistics 10 (2022) 291–306.
[18] J. Zhang, X. Guo, S. Bodapati, C. Potts, Multi-teacher distillation for multilingual spelling correction, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP), 2023, pp. 142–151.
[19] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with MT-Bench and chatbot arena, in: Advances in Neural Information Processing Systems (NeurIPS), 2023.
[20] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large language models are human-level prompt engineers, in: International Conference on Learning Representations (ICLR), 2023.
[21] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, J. E. Gonzalez, RAFT: Adapting language model to domain specific RAG, arXiv preprint arXiv:2403.10131 (2024).