Universal Library of Engineering Technology Research Article

ISSN: 3064-996X | Volume 3, Issue 2

Open Access | PP: 110-117

DOI: https://doi.org/10.70315/uloap.ulete.2026.0302017

# Relevance Optimization in Neighborhood-Scale Search Using Proximity and Freshness Signals: A Hybrid AI Retrieval Architecture For Hyperlocal Community Platforms

**Venkata Karunakara Reddy Revunuru**

Senior Software Engineer – Search & AI Platforms Independent Technology, Phoenix, Arizona, USA.

## Abstract

*The article examines the design of geo-aware information-retrieval pipelines for hyperlocal community platforms, where ranking quality depends on small-radius distance, rapid content turnover, and sparse, noisy texts. The relevance problem is framed as a hybrid retrieval task that unifies lexical matching and embedding-based similarity while incorporating proximity and freshness as first-class ranking signals. The work targets an architecture that reduces engineering friction by consolidating sparse and dense retrieval within a single operational search stack and by utilizing fusion methods that remain stable under shifting query intent. The study outlines scoring functions for distance and time decay, candidate-generation strategies under geographic constraints, and ranking fusion options to balance precision and recall. Source materials cover hybrid rank fusion, integration of Lucene-based dense retrieval, geo-tagged vector querying, and spatial–keyword indexing. The article provides practical guidance for platform teams building neighborhood-scale discovery systems that rank results based on user intent, local time, and real-time availability, rather than proximity and freshness alone. Safety and moderation are treated as low-latency eligibility gates integrated into both sparse and dense retrieval paths. A concrete WingBud-style example illustrates intent-driven matching and contextual ranking for short-duration neighborhood interactions.*

**Keywords:** *Hybrid Information Retrieval, Geo-Aware Search, Neighborhood-Scale Ranking, Intent-Aware Ranking, Availability-Aware Scoring, Real-Time Safety And Moderation.*

## INTRODUCTION

Neighborhood-scale discovery differs from city-scale search in three properties that directly affect ranking construction. First, the geographic radius becomes small enough that minor distance differences meaningfully change user utility, forcing explicit distance decay rather than coarse "near me" filtering. Second, content churn becomes rapid: posts, classifieds, and offers lose utility quickly, so time decay must compete with textual relevance rather than act as a secondary sorting mechanism. Third, textual fields are short and heterogeneous, so pure keyword matching under-represents intent, while pure embedding similarity can over-generalize and surface semantically related but locally irrelevant items. Under these conditions, the search stack requires a combined retrieval-and-ranking workflow where proximity and freshness operate alongside lexical and semantic evidence.

A fourth property becomes decisive in short-duration hyperlocal interactions: intent is often situational and time-bounded, and ranking must reflect it explicitly. In products where users act within minutes (e.g., "open now", "available tonight", "within a 10-minute walk"), proximity and freshness alone are insufficient: two items at the same distance can differ radically in utility if one is closed, unavailable, or mismatched to the user's immediate intent. Therefore, neighborhood-scale ranking benefits from intent-driven scoring that conditions relevance on inferred intent type (transactional vs. informational vs. social coordination), local time features (time-of-day, day-of-week, near-term horizon), and availability signals (open/closed, remaining capacity, response likelihood). This motivates treating intent, time, and availability as first-class ranking signals integrated into the same scoring layer as lexical/semantic evidence and geo-temporal decay.

The article aims to justify and specify a hybrid AI retrieval architecture for hyperlocal platforms that integrates BM25-style retrieval, embedding-based retrieval, and geotemporal

**Citation:** Venkata Karunakara Reddy Revunuru, "Relevance Optimization in Neighborhood-Scale Search Using Proximity and Freshness Signals: A Hybrid AI Retrieval Architecture For Hyperlocal Community Platforms", Universal Library of Engineering Technology, 2026; 3(2): 110-117. DOI: https://doi.org/10.70315/uloap.ulete.2026.0302017.

ranking signals within a single production pipeline. The tasks are:

1) to define an operational ranking function that unifies proximity decay and freshness decay with lexical and semantic relevance;

2) to specify candidate-generation and index design choices that stay efficient under geographic constraints and sparse local data;

3) to compare fusion strategies and derive design rules for stable relevance under shifting query intent and content churn.

## MATERIALS AND METHODS

The study's materials comprise recent publications on hybrid rank fusion, Lucene-based dense retrieval integration, spatial–keyword indexing, and geo-tagged vector querying. Bruch et al. analyze rank-fusion functions for hybrid retrieval and describe differences in stability between fusion operators across varying retrieval quality [1]. Chaoji et al. propose range-filtering approximate nearest neighbor search structures, relevant to filtered vector retrieval under ordered attributes such as time or price [2]. Dong et al. study continuous top-k spatial–keyword retrieval over dynamic objects, grounding grid-based spatial indexing choices for moving or frequently updated local items [3]. Liu and Zhang propose a fusion framework that applies modified reciprocal-rank fusion across parallel retrieval routes (original vs expanded query), thereby enabling robust fusion under language variation [4]. Ma et al. integrate Lucene HNSW indexes into Anserini and discuss the practical tradeoffs of combining dense and sparse retrieval in a single stack [5]. Martins et al. outline system-level requirements for geo-temporal retrieval and synthesis, motivating explicit handling of spatial and temporal constraints in advanced retrieval pipelines [6]. Sager et al. report competitive hybrid retrieval and re-ranking results for scientific paper retrieval, providing empirical evidence that hybrid candidate merging, combined with a more potent final scorer, can improve early precision [7]. Xian et al. study top-k spatial-range-constrained approximate nearest neighbor search over geo-tagged vectors, describing workload-aware indexing for combined spatial filters and vector similarity [8]. Xu et al. analyze efficient processing of top-k frequent spatial keyword queries, supporting frequency-aware optimization for recurring local query patterns [9].

The methodological approach is analytical and architectural. The work employs a structured literature analysis and comparative synthesis of retrieval operators, deriving scoring formulations for proximity and recency with calibration constraints. It also constructs an architectural pattern that maps retrieval stages (candidate generation, fusion, re-ranking) to operational constraints (latency, update rates, sparsity). Evaluation is discussed through standard retrieval effectiveness metrics (e.g., MRR, nDCG, recall) and system metrics (latency, index size, update throughput).

To operationalize intent-driven ranking, the architecture assumes a lightweight intent inference step that classifies queries and interaction traces into intent families (e.g., "need now", "plan soon", "browse"). Features include query terms ("open now", "nearby", "available"), session time horizon, recent user actions (click depth, dwell, message initiation), and local temporal context (night vs. daytime). Availability is represented as structured attributes (open hours, inventory/capacity, last-seen recency, reply-rate priors) that can be filtered or incorporated as multiplicative utilities in the final scorer. The intent classifier is treated as a latency-bounded component whose output steers feature weights rather than triggering heavy re-ranking by default.

## RESULTS

A neighborhood-scale search pipeline can be formalized as a staged retrieval process with explicit geo-temporal gating and hybrid evidence fusion. The first stage performs coarse eligibility filtering: geographic bounds (circle or polygon), real-time safety/moderation eligibility rules (policy, abuse-risk, reputation, interaction constraints). On safety-sensitive community platforms, eligibility filtering is not limited to visibility flags; it serves as a real-time safety and moderation gate that precedes retrieval to prevent harmful exposure and protect users during interactions. The gate enforces complex policy rules (banned categories, location-based restrictions, user blocks, age/consent constraints where applicable), automated content screening (toxicity/harassment classifiers, scam and impersonation detectors, URL risk scoring), and behavioral rate limits (message bursts, repeated outreach, account reputation degradation). Signals are evaluated at query time and at interaction time: an item can be searchable but interaction-disabled if risk increases (e.g., sudden report spikes). Architecturally, the gate should be implemented as a low-latency policy service with auditable decisions, producing a candidate-eligibility mask that is consumed by both lexical and vector channels, so that unsafe candidates never enter rank fusion.

In sparse local ecosystems, this stage is not merely a performance optimization; it also protects relevance by preventing semantically similar but geographically irrelevant content from dominating candidate results.

The second stage performs hybrid candidate generation, where two retrieval channels run in parallel:

i) lexical retrieval using an inverted index (BM25-family scoring) for precision on named entities, categories, and exact attributes;

ii) embedding retrieval using approximate nearest neighbor search over dense vectors for intent tolerance and paraphrase robustness.

Consolidating these two channels inside a unified stack reduces synchronization overhead between separate stores. It simplifies consistent deletion and refresh of short-lived

local items, which aligns with the integrated Lucene HNSW approach described for dense retrieval within an established IR toolkit [5] and with Lucene-based embedding search demonstrations [3]. For hyperlocal data, the operational benefit is amplified because frequent updates and deletions are common in community feeds and classifieds.

A central design decision concerns filtered vector search: neighborhood retrieval typically applies a tight spatial filter first, then computes similarity within that region. If the system instead runs a global vector search and filters afterward, it wastes computation and risks candidate drift toward popular but non-local items. Work on range-filtering ANNS addresses a structurally similar problem: retrieving nearest neighbors under constraints on an ordered attribute (e.g., timestamp, price), where naive approaches degrade when selectivity shifts [9].

The cost of "global ANN then post-filter" can be approximated as the number of wasted similarity computations. If an ANN query returns K candidates globally, and the spatial predicate selectivity is p (fraction of items inside the neighborhood window), then the expected survivors after post-filtering are K·p. To retain N usable candidates, the system must request K≈Np, inflating the ANN's work by a factor of 1/p. For p=0.05 (tight radius in a dense city), obtaining N=200 neighborhood candidates implies K≈4000 global candidates, which materially affects latency and cache behavior. This back-of-the-envelope calculation motivates range-filtering and spatial-range-constrained ANN formulations as engineering responses to selectivity variability [2].

Geo-tagged vectors generalize the constraint to spatial windows; k-RANNS explicitly models a query vector along with a spatial range and targets stable performance across selectivity regimes by utilizing workload-aware indexing under memory limits [7]. Translated to neighborhood search, these results support designing the vector channel to respect tight spatial predicates rather than treating them as an afterthought.

After candidate generation, the third stage performs fusion.

A practical advantage of reciprocal-rank style fusion is that it avoids cross-channel score calibration by operating on ranks. For two channels s∈{lex, dense}, the RRF score for a document d is (1):

$$RRF(d) = \sum_{s\in\{lex,\, dense\}} \frac{1}{k+rank_s(d)}. \quad (1)$$

For example, with k=60, if an item is ranked 2nd in BM25 candidates and 10th in ANN candidates, then (2):

$$RRF(d) = \frac{1}{(60+2)} + \frac{1}{(60+10)} = \frac{1}{62} + \frac{1}{70} \approx 0.01613 + 0.01429 \approx 0.03042. \quad (2)$$

This explicit arithmetic clarifies why RRF remains stable when lexical and embedding scores live on incompatible scales, a property emphasized in fusion-function comparisons [1]. Modified RRF over parallel retrieval routes (original vs expanded query) preserves the same scale-invariance while increasing route diversity; its operational cost is dominated by the expansion path and the need to cap expansion fan-out to protect latency budgets [4].

Hybrid fusion must handle asymmetric failure modes: lexical retrieval can miss semantically phrased intent; embedding retrieval can over-match semantically related content that does not satisfy local intent. Fusion functions behave differently under these conditions. A systematic analysis of hybrid fusion functions reveals that the choice of fusion affects robustness when one channel becomes noisier or when the rank distributions shift [1]. In a neighborhood setting, such shifts occur naturally: daytime versus nighttime queries, event-driven bursts, and locality-driven vocabulary variation. Practical fusion, therefore, benefits from operators that remain stable when one channel temporarily underperforms, complemented by calibration on short-text corpora typical of community posts.

The ranking stage then integrates proximity and freshness. Proximity can be modeled as a monotone distance-decay term that is smooth near zero (to avoid tie explosions inside tiny radii) and has a controllable tail (to prevent marginally farther items from disappearing when the radius expands slightly).

For short-duration hyperlocal tasks, the final stage benefits from intent-conditioned weighting and an explicit availability utility. Let I(q) denote the inferred intent family for query q (e.g., immediate-need vs. planned vs. exploratory). Let savail(d, q, t) be an availability score computed from item state at local time t (open/closed, remaining capacity, response-likelihood prior, and near-term time-window match). A practical formulation keeps fusion as the base relevance signal while applying intent-specific weights:

$$S(d|q,t) = \alpha(I(q))R_{fuse}(d|q) + \beta(I(q))s_{dist}(d|q) + \gamma(I(q))s_{time}(d|t) + \delta(I(q))s_{avail} + (d,q,t) \quad (3)$$

Here, α(I), β(I), γ(I), and δ(I) are selected per intent family rather than fixed globally. For example, "open now" intent increases δ and γ while reducing β when the user accepts a slightly longer walk for guaranteed availability; navigational intent increases α and reduces δ to prevent availability heuristics from overriding exact-name relevance. This preserves the hybrid retrieval core while aligning ranking with minute-scale decision-making.

Distance and time decays become tunable controls when expressed as half-life parameters that product teams can align with user expectations for each content class. A convenient parameterization uses an exponential decay with a half-life (4; 5):

$$s_{dist}(d) = exp(-\frac{d}{\lambda}), \text{where } \lambda = \frac{d1/2}{ln2}; \quad (4)$$

$$s_{time}(t) = exp(-\frac{t}{\tau}), \text{where } \tau = \frac{t1/2}{ln2}. \quad (5)$$

If neighborhood utility halves every $d_{1/2}$=800 m, then λ≈800/0.693≈1154 m, giving $s_{dist}$(200 m)=exp(−0.173)≈0.841 and $s_{dist}$(1600 m)=exp(−1.386)≈0.250.

If content utility halves every $t_{1/2}$=24 h for alerts, then τ≈24/0.693≈34.6 h, giving $s_{time}$(6 h)≈0.841 and $s_{time}$(48 h)≈0.250.

This numeric grounding complements geo-temporal system framing by making "freshness" and "nearby" concrete calibration levers rather than qualitative requirements [6].

Freshness can be modeled as a time-decay term defined over item age, with a half-life tuned to the content class (alerts, classifieds, or evergreen businesses). The architectural requirement is not the existence of these decays, but rather their interaction with retrieval evidence: proximity and freshness should modulate the final score without overriding textual relevance when the user's intent is clearly specific (e.g., a named place). Spatial–keyword indexing research provides operational grounding for efficiently retrieving candidates where keywords and spatial predicates co-govern result eligibility [2], and frequent spatial keyword query processing supports caching and optimization when a subset of local intents repeats (e.g., "pharmacy near me", "open now") [8].

Figure 1 summarizes the resulting geo-aware hybrid retrieval pipeline used for neighborhood-scale discovery, aligning unified dense–sparse retrieval practices [3; 5] with constrained similarity retrieval under geo/temporal predicates [7; 9] and spatial–keyword indexing considerations [2].

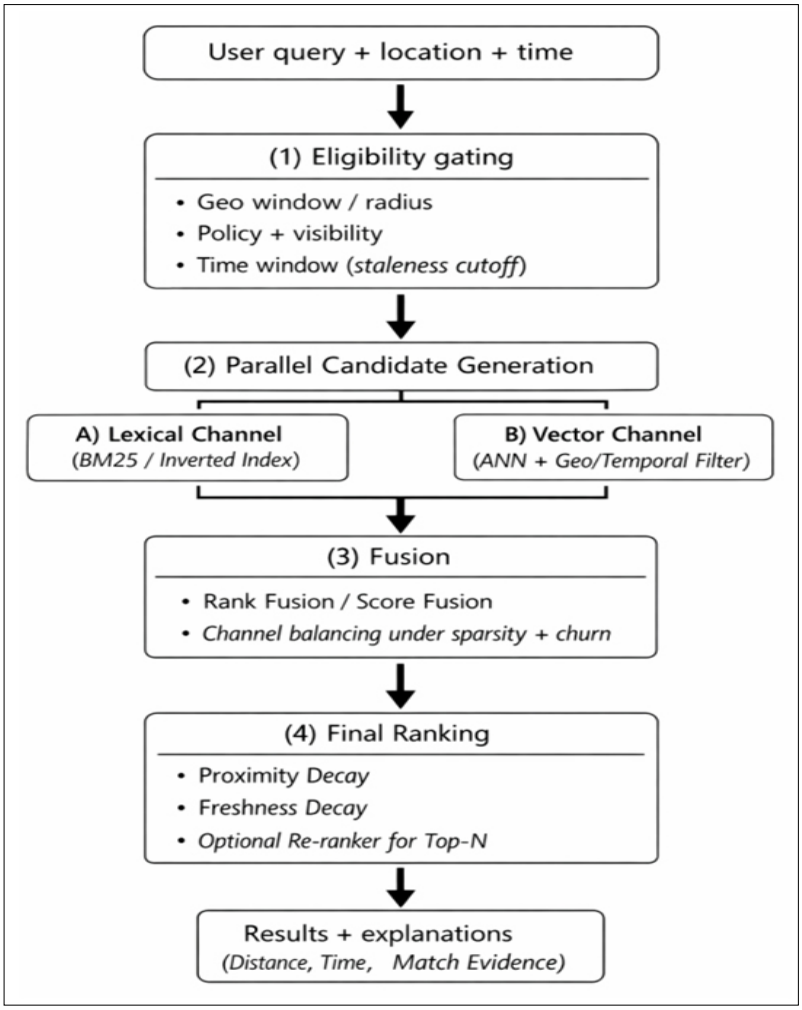


**Figure 1.** Geo-aware hybrid retrieval and ranking pipeline for neighborhood-scale search (compiled by the author from his own research)

Consider WingBud as a neighborhood-scale social coordination feature where users seek short-duration meetups ("find a buddy for a quick walk", "coffee now", "pickup game tonight") and where the primary constraint is immediate feasibility rather than long-term content relevance.

Example query A (immediate need): "coffee buddy near me open now".

1) Intent inference classifies the query as immediate-need with a near-term horizon (≤30 minutes), triggered by "open now" and interaction patterns (shallow query, rapid re-queries).

2) Safety/moderation gate filters candidates by user blocks, reputation thresholds, harassment signals, and scam/ impersonation risk; interaction privileges can be restricted even if profiles remain viewable.

3) Candidate generation runs two channels inside the neighborhood window: lexical retrieval captures explicit terms ("coffee", "open now"), while embeddings retrieve paraphrases ("grab a latte", "meet for a quick café").

4) Fusion (e.g., RRF) preserves diversity between exact-term matches and paraphrase matches.

5) Contextual ranking applies availability and time: savail rewards users who are currently active, recently responsive, and aligned with the time window; stime penalizes stale "availability" posts; sdist keeps walking-time reasonable but does not override feasibility. In this intent family, $\delta(I)$ and $\gamma(I)$ increase to prioritize immediate feasibility.

Example query B (plan soon): "pickup basketball tonight".

1) Intent inference marks plan-soon with a horizon of hours.

2) The gate enforces safety (spam outreach throttling, age/consent constraints if applicable, blocklists) and can downrank high-risk accounts even when not entirely removed.

3) Candidate generation retrieves both posts ("game at 8 pm") and user profiles with semantically similar interests, within the neighborhood radius.

4) Ranking shifts weight from "open now" availability to scheduled-time match, using a time-window overlap feature: items whose start time aligns with tonight receive higher availability, while pure recency receives lower weight than in Example A.

Impact: In WingBud, the same distance and freshness signals produce different orderings depending on intent and availability. Two candidates at equal proximity can swap ranks when one is responsive and available within the required time window, while the other is offline or misaligned with the session horizon. This concretely links the architecture to short-duration, context-sensitive interactions rather than to static "nearby feed" ranking.

A final re-ranking step can be applied to the fused top-N list to refine early precision, especially when user intent is implicit or when texts are short. Evidence from a competitive hybrid retrieval setting shows that combining BM25 candidates with dense retrieval candidates and applying a stronger re-ranking stage yields high early-rank effectiveness (reported MRR@5 values on development and hidden test sets) [7]. While that evaluation domain differs from hyperlocal discovery, the structural lesson transfers: hybrid candidate merging followed by a higher-capacity scorer is a reliable pattern when raw retrieval scores are not directly comparable across channels.

For operational deployment, the results imply three implementable constraints. First, geo-filtrering must be pushed as early as possible in both lexical and vector channels to prevent candidate drift and to control latency, consistent with constrained similarity search formulations over geo-tagged vectors [9]. Second, fusion must be selected and tuned for robustness to channel-quality fluctuations, consistent with empirical comparisons of fusion operators [1] and with fusion frameworks that explicitly combine alternative retrieval routes (original vs expanded) via a modified RRF [4]. Third, update-heavy local inventories demand indexing patterns that minimize reindex cost while preserving search correctness, aligning with dynamic spatial–keyword processing techniques in continuous settings.

## DISCUSSION

The proposed neighborhood-scale architecture can be interpreted as an attempt to make three scoring families commensurable: lexical matching, semantic similarity, and geo-temporal utility.

To operationalize commensurability under short-duration hyperlocal behavior, the final score is more stable when its weights are conditioned on inferred intent rather than fixed globally. A convenient representation is (6):

$$S(d|q,t) = \alpha(I(q))R_{fuse}(d|q) + \beta(I(q))s_{dist}(d|q) + \gamma(I(q))s_{time}(d|t) + \delta(I(q))s_{avail} + (d,q,t) \quad (6)$$

where I(q) is the intent family (immediate-need, plan-soon, exploratory, navigational). This makes a concrete constraint visible: proximity and freshness are not universal utilities; their contribution depends on whether the session reflects urgent action ("available now"), planning ("this weekend"), or navigation ("Starbucks on X Street"). Conditioning weights reduces drift across time-of-day and availability regimes without discarding the hybrid fusion backbone.

For an illustrative setting α=0.60, β=0.25, γ=0.15, and for an item with Rfuse=0.72, sdist=0.84, stime=0.25, one obtains (7):

$$S = 0.60 \cdot 0.72 + 0.25 \cdot 0.84 + 0.15 \cdot 0.25 = 0.432 + 0.210 + 0.0375 = 0.6795 \quad (7)$$

The arithmetic highlights a design constraint that is easy to miss in prose: even strong proximity cannot compensate for weak fused relevance unless β is set aggressively, and aggressive β increases drift risk for specific/navigational queries. This links directly to fusion-operator stability findings [1] and to the engineering decision to treat filtered retrieval as a first-order mechanism rather than a late correction [9].

The literature on hybrid fusion warns that rank fusion is not a neutral glue; different fusion operators encode distinct assumptions about channel reliability and score comparability [1]. On hyperlocal platforms, channel reliability varies by topic and time: lexical search tends to dominate for navigational intent and exact queries, while embeddings dominate for conversational, paraphrased, or multilingual phrasing. A stable design, therefore, benefits from a fusion layer that can be calibrated per vertical (posts vs businesses vs. classifieds) and per intent family, with fallback logic when one channel yields shallow or noisy ranks. Query expansion introduces an additional robustness lever: Exp4Fuse shows that fusing ranks from original and LLM-augmented query routes can improve sparse retrieval without requiring complete dense retrieval in the expansion loop, which is attractive when operational budgets constrain embedding throughput [4].

Textual sparsity and locality introduce a second tradeoff: aggressive freshness boosting improves feed satisfaction but can erode relevance when the query is specific. Spatial–keyword research supports index structures and query processing that keep spatial predicates and keyword predicates jointly active, thereby avoiding late-stage corrections that can inflate computation [3; 8]. For dense retrieval under constraints, range-filtering ANNS and geo-tagged vector search highlight that selectivity variability can destabilize latency; designs that adapt to selectivity or compress multiple filtered indexes into compact structures address that instability at the data-structure level [9], while workload-aware indexes for spatial-range-constrained ANN pursue memory-bounded stability [8]. This supports a practical engineering stance: filtered vector retrieval should be treated as a first-order capability rather than a bolt-on.

Fusion strategy selection benefits from an explicit trade-space view, because neighborhood-scale workloads combine frequent channel-quality swings (lexical dominance for navigational/exact queries vs. embedding dominance for paraphrase and multilingual phrasing), and strict geo-temporal predicates that constrain candidate sets unevenly across queries. To make the comparison operational rather than purely descriptive, the strategies are positioned on two normalized axes: robustness to channel-quality fluctuations and implementation/operational cost. Robustness is computed as a weighted composite of score-scale invariance, tolerance to one-channel degradation, and short-text robustness; cost aggregates engineering integration effort, latency impact at peak QPS, and maintenance burden (re-training, feature drift, monitoring).

Let the robustness score be (8; 9):

$$R = 0.35I_{scale} + 0.35T_{degrade} + 0.30U_{short}, \quad (8)$$

Moreover, the cost score is (8):

$$C = 0.40E_{eng} + 0.35L_{latency} + 0.25M_{maint}, \quad (9)$$

where each component is normalized to [0, 1]. Under the assumptions suggested by rank-fusion analyses (robustness when score scales differ) [1], query-route fusion for sparse retrieval with LLM expansion [4], and the operational overhead of learned re-rankers (higher-capacity scorer on top-N) [7], one illustrative parameterization yields the plotted points:

• RRF: R≈0.82, C≈0.45 (high scale-invariance; modest engineering and latency).

• Modified RRF over parallel routes (original vs expanded): R≈0.86, C≈0.62 (added robustness via route diversity; higher cost from expansion and routing).

• Hybrid + learned re-ranking: R≈0.74, C≈0.78 (strong peak effectiveness potential; higher maintenance and latency budgeting).

Figure 2 Fusion strategy trade-space for neighborhood-scale hybrid search (normalized robustness vs. implementation/ operational cost; rubric-based positioning compiled by the author).

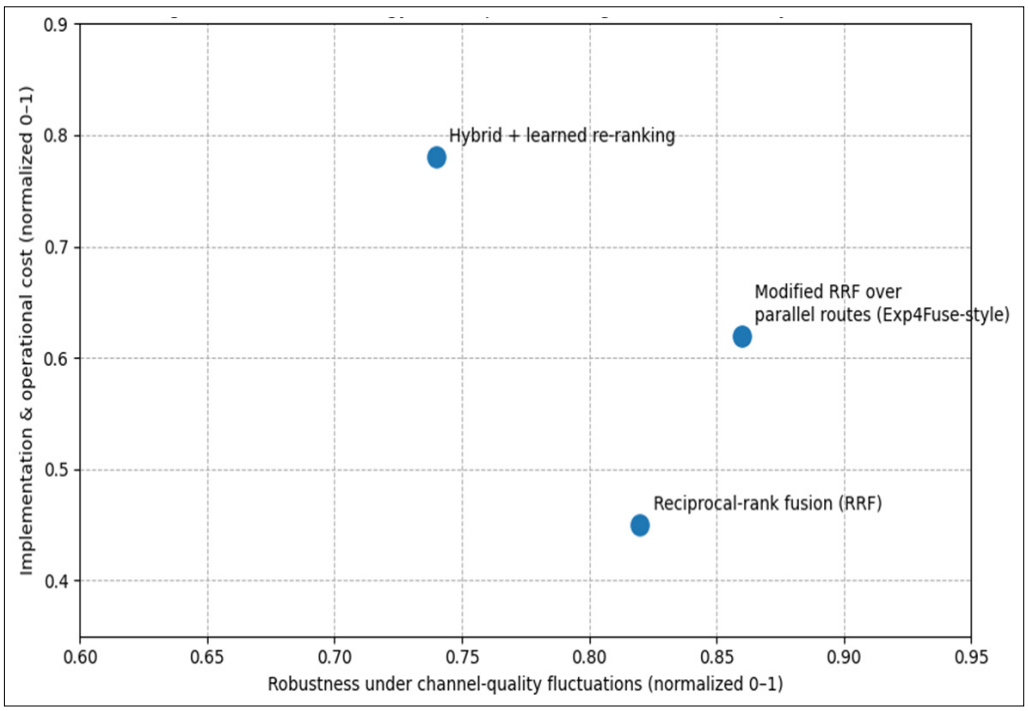


**Figure 2.** Fusion strategies for hybrid neighborhood search and their operational consequences [1; 4; 7]

Figure 2 reframes fusion choice as a constrained engineering decision. When strict geo-filters and churn shrink candidate pools, a fusion operator that remains stable under score-scale mismatch is often preferable to directly mixing raw scores. Route-level fusion (original vs expanded) adds robustness when lexical mismatch is the dominant failure mode, while learned re-ranking is better treated as an optional refinement layer that consumes a fused, diversity-preserving top-N list rather than replacing fusion. This aligns the operator choice with the dominant instability source: channel reliability variance across time of day, neighborhood vocabulary drift, and bursty event traffic [1; 4].

The design implication is that fusion and re-ranking are not interchangeable: fusion stabilizes candidate diversity, while re-ranking refines the ordering within that diversified set. A hyperlocal system that skips fusion diversity can overfit to either exact terms or semantic similarity, depending on the traffic segment, creating visible drift for users moving across neighborhoods.

Real-time moderation distinguishes neighborhood-scale community search from conventional web search and static recommendations. In hyperlocal social discovery, ranking directly influences who can contact whom, under time pressure and a small geographic radius. Therefore, safety constraints act as dynamic eligibility predicates coupled with ranking, not as a post-hoc content-removal workflow. This coupling changes system design choices: policy decisions must be latency-bounded, explainable, and integrated into retrieval so that fusion never amplifies risky candidates; monitoring must track not only relevance metrics (nDCG/MRR) but interaction-risk outcomes (report rate, block rate, repeat-contact anomalies) as first-order operational signals.

Indexing and data-model choices are summarized for sparse local inventories with frequent updates and tight geo constraints (Table 1).

**Table 1.** Indexing and retrieval patterns for sparse, short-lived neighborhood data [2; 5; 7–9]

| Pattern | Rationale for hyperlocal data | Evidence base |
|---|---|---|
| Grid-based spatial partitioning for dynamic items | Limits update scope and supports continuous maintenance of top-k results when locations/ keywords change | Continuous spatial keyword tracking using a grid index |
| Frequency-aware optimization for recurring local intents | Repeated local queries benefit from specialized processing and caching around frequent spatial keyword patterns | Efficient processing of frequent spatial keyword queries |
| Spatial-range-constrained ANN over geo-tagged vectors | Dense retrieval remains usable when candidates must fall inside a spatial window, with stability across selectivity regimes | k-RANNS formulation and workload-aware index under memory constraints |

| | | |
|---|---|---|
| Range-filtering ANNS structures for ordered constraints | Supports constrained vector retrieval under changing selectivity for attributes such as time windows (freshness gating) | Segment-graph compression of many constrained indexes |
| Unified dense–sparse retrieval in one stack | Reduces operational friction from dual stacks and simplifies synchronized updates | Lucene HNSW integration for dense retrieval alongside inverted indexes |

Two limitations emerge. First, academic-constrained ANN work often optimizes for ordered constraints or spatial windows in isolation; real hyperlocal products, however, frequently require compound predicates (e.g., geo, time, category, and policy). This suggests an engineering need for composable filtering strategies that preserve ANN efficiency. Second, hybrid retrieval studies often evaluate on web-scale corpora; neighborhood corpora exhibit stronger churn and shorter texts, so calibration and offline evaluation must incorporate temporal splits and locality-aware sampling, echoing geo-temporal evaluation concerns raised for retrieval systems that handle spatial and temporal constraints explicitly.

## CONCLUSION

The article presents a geo-aware hybrid retrieval architecture for hyperlocal community platforms and advocates for treating proximity and freshness as integrated ranking signals, rather than post-filters. The first task is addressed by defining a staged ranking design in which proximity and time decay modulate a fused relevance signal, without collapsing lexical and semantic evidence into a single, unreliable scale. The second task is addressed through index and candidate-generation patterns grounded in spatial–keyword processing for dynamic objects and in constrained similarity search over geo-tagged vectors, ensuring feasibility under sparse local data and frequent updates. The third task is addressed by comparing fusion and re-ranking strategies and by deriving selection rules that prioritize robustness under fluctuating channel quality and shifting query phrasing.

## REFERENCES

1. Bruch, S., Gai, S., & Ingber, A. (2024). An analysis of fusion functions for hybrid retrieval. ACM Transactions on Information Systems, 42(1), Article 20. https://doi.org/10.1145/3596512
2. Chaoji, Z., Qiao, M., Zhou, W., Li, F., & Deng, D. (2024). SeRF: Segment graph for range-filtering approximate nearest neighbor search. Proceedings of the ACM on Management of Data, 2(1), Article 69. https://doi.org/10.1145/3639324
3. Dong, Y., Xiao, C., Chen, H., et al. (2021). Continuous top-k spatial–keyword search on dynamic objects. The VLDB Journal, 30, 141–161. https://doi.org/10.1007/s00778-020-00627-4
4. Liu, L., & Zhang, M. (2025). Exp4Fuse: A rank fusion framework for enhanced sparse retrieval using large language model-based query expansion (arXiv:2506.04760). arXiv. https://arxiv.org/abs/2506.04760
5. Ma, X., Teofili, T., & Lin, J. (2023). Anserini gets dense retrieval: Integration of Lucene's HNSW indexes. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (pp. 5366–5370). https://doi.org/10.1145/3583780.3615156
6. Martins, B., & Gramacki, P. (2025). A vision for geo-temporal deep research systems: Towards comprehensive, transparent, and reproducible geo-temporal information synthesis (arXiv:2506.14345). arXiv. https://arxiv.org/abs/2506.14345
7. Sager, P. J., Kamaraj, A., Grewe, B. F., & Stadelmann, T. (2025). Deep retrieval at CheckThat! 2025: Identifying scientific papers from implicit social media mentions via hybrid retrieval and re-ranking (arXiv:2505.23250). arXiv. https://arxiv.org/abs/2505.23250
8. Xian, J., Teofili, T., Pradeep, R., & Lin, J. (2024). Vector search with OpenAI embeddings: Lucene is all you need. In Proceedings of the 17th ACM International Conference on Web Search and Data Mining (pp. 1090–1093). https://doi.org/10.1145/3616855.3635880
9. Xu, T., Xu, A., Mango, J., Liu, P., Ma, X., & Zhang, L. (2022). Efficient processing of top-k frequent spatial keyword queries. Scientific Reports, 12(1), Article 7352. https://doi.org/10.1038/s41598-022-10648-4