

AI-Driven Conversational Rental Marketplace with Fuzzy Hybrid Search and Trust-Based Moderation

Abheejan Lal Shrestha

Department of Computer Engineering,
Nepal College of Information Technology,
Pokhara University
Lalitpur, Nepal
abheejan.221202@ncit.edu.np

Pratik Mishra

Department of Computer Engineering,
Nepal College of Information Technology,
Pokhara University
Lalitpur, Nepal
pratik.221226@ncit.edu.np

Swastik Shrestha

Department of Computer Engineering,
Nepal College of Information Technology,
Pokhara University
Lalitpur, Nepal
abheejan.221244@ncit.edu.np

Amit K. Shrivastava

Department of Computer Engineering,
Nepal College of Information Technology,
Pokhara University
Lalitpur, Nepal
amit.shrivastava@ncit.edu.np

Abstract—Finding rental accommodation in the Kathmandu Valley is fragmented, broker-dependent, and plagued by fraudulent listings. Conventional property portals reduce rich, hyperlocal requirements—water reliability, sunlight, power backup, neighborhood character—to a handful of rigid filters, and offer little defense against fake advertisements. This paper presents *RoomieKtm*, a domain-specific rental-marketplace system that integrates conversational search, hybrid retrieval, fuzzy re-ranking, and automated trust scoring—each built on established techniques—into a single, purpose-built pipeline for this market. We describe a hybrid retrieval path that fuses lexical (BM25) and dense semantic (vector k-NN) candidates via Reciprocal Rank Fusion, then re-ranks them with a weighted, zero-order Sugeno-type fuzzy aggregation that models graded, tolerant preferences. To combat fraud, we introduce a five-factor trust-score model that drives a three-way automated moderation router, complemented by a human admin review queue, tenant reporting, and an append-only audit log. We further describe an administrator-reviewed landlord identity-verification (KYC) workflow with explicit data-protection safeguards. We evaluate the pipeline on a controlled pilot collection (93 listings, 42 labeled queries) with a four-arm ablation—BM25, semantic-only hybrid without fuzzy re-ranking, and the full hybrid-plus-fuzzy pipeline—and report paired significance tests alongside point estimates. The fuzzy layer recovers ranking quality specifically on paraphrased, vocabulary-mismatched queries (nDCG@10 0.662 vs. 0.622 for RRF fusion alone), and the trust router correctly flags 13 of 15 constructed fraud patterns (recall 0.867, accuracy 0.914); however, none of the observed differences reach statistical significance at this sample size (Wilcoxon signed-rank, all $p > 0.05$), a limitation we report plainly rather than obscure. We treat this as an honest, small-scale validation of correct implementation and design direction, not a proof of production-scale performance, and we detail exactly what a larger, real-world evaluation would need to add.

Keywords—conversational search, information retrieval, hybrid search, reciprocal rank fusion, fuzzy aggregation, trust and safety, content moderation, geospatial search, rental marketplace, systems integration.

I. INTRODUCTION

The rental market in the Kathmandu Valley is largely informal. Information is scattered across brokers, social-media groups, and

physical “To Let” signs, and tenants frequently pay commissions simply to discover listings that are otherwise hard to find. Students relocating for higher education and tourists seeking short- to medium-term stays are particularly disadvantaged: they are new to the city, unfamiliar with neighborhood pricing, and most exposed to misleading or outright fraudulent advertisements.

Existing online property platforms attempt to address discovery with structured search forms. However, these filters poorly capture how people actually describe a home. A renter cares whether a flat receives morning sunlight, whether the building has reliable municipal water or depends on tankers, and how frequent load-shedding is—concerns that conventional dropdowns flatten into a few checkboxes. The result is a tedious cycle of filter adjustment and manual scanning of irrelevant results.

A second, more corrosive problem is trust. Unmoderated channels are rife with stock-photo listings, bait prices, and brokers impersonating owners. Tenants have little means to verify a listing before committing time or money to it.

This paper presents *RoomieKtm*, a mobile-first marketplace that addresses both problems. We are explicit about the nature of our contribution: BM25, sentence embeddings, Reciprocal Rank Fusion, fuzzy aggregation, and perceptual hashing are all established techniques, and we make no claim of algorithmic novelty at the component level. The contribution is their integration, adaptation, and evaluation as a single, purpose-built system for an underserved, hyper-local market, together with a transparent account—including negative and non-significant results—of how well that integration performs on a controlled pilot. Concretely:

(1) A *conversational retrieval pipeline* that combines lexical and dense semantic search with Reciprocal Rank Fusion and a weighted fuzzy re-ranking layer, allowing graded preferences (e.g., a listing slightly over budget) to be scored rather than discarded.

(2) A *five-factor trust-score model* and an automated three-way moderation router that auto-approves clearly legitimate listings, auto-rejects clear fakes, and escalates ambiguous cases to a human admin, supported by tenant reporting and an append-only audit log.

(3) An *administrator-reviewed landlord verification (KYC)* workflow that issues a visible trust badge, feeds back into the trust score, and is subject to explicit data-protection safeguards.

(4) A *four-arm controlled ablation* (BM25, semantic-only, hybrid without fuzzy, hybrid with fuzzy) with paired statistical significance testing, reported honestly including where differences are not significant, in place of point estimates alone.

II. RELATED WORK

A. Conversational and Semantic Retrieval

Natural-language interfaces let users express intent directly rather than mapping it onto predefined controls. As Manning *et al.* [1] emphasize, effective query understanding—intent detection and entity extraction—is foundational to bridging human language and structured retrieval. Dense vector representations encode text into a space where semantically similar content lies close together; Reimers and Gurevych [2] show that sentence-transformer embeddings enable robust semantic matching even when query and document wording differ. RoomieKtm's production design uses such embeddings to capture lifestyle semantics that lexical search misses (Section VII-B notes a substitution used only in this paper's offline pilot).

B. Hybrid Search and Rank Fusion

Lexical scoring functions such as the BM25 model of Robertson and Zaragoza [3] remain strong, interpretable baselines for keyword relevance. Lin *et al.* [4] report that combining sparse and dense retrieval consistently outperforms either in isolation. To merge ranked lists without delicate score normalization, the Reciprocal Rank Fusion (RRF) method of Cormack *et al.* [5] provides a simple, parameter-light alternative that is robust to differences in score scale. We adopt RRF to fuse our two retrieval paths, and in Section VIII isolate its contribution from that of the fuzzy layer that sits atop it.

C. Fuzzy Aggregation for Graded Relevance

Rental preferences are inherently imprecise: a listing marginally over budget is not “wrong,” merely less desirable. The fuzzy set theory introduced by Zadeh [6] models such graded membership on a continuous interval. The Takagi–Sugeno inference framework [7] more generally aggregates rule-conditioned membership values, often through weighted combination, to produce a crisp output. RoomieKtm's re-ranker is best described precisely: it is a single-rule, zero-order Sugeno-type aggregator (Section IV-E clarifies this terminology directly), not a multi-rule fuzzy inference system with distinct antecedent-consequent rules. We use it because a weighted combination of graded membership functions is exactly the tool needed to avoid the sharp cutoffs of crisp filtering, while remaining simple enough to inspect, test, and explain to an end user.

D. Trust, Safety, and Geospatial Search

Online marketplaces routinely combat fraud through reputation and content-moderation signals. We operationalize a

compact set of fraud indicators—image originality via perceptual hashing, account maturity, price normality, description quality, and report history—into a single trust score. For spatial relevance, the Haversine formula computes great-circle distance from coordinates [8], and document databases support efficient radius queries through 2dsphere indexes, which we use for map-based discovery.

E. Positioning Against Existing Platforms

To the best of our knowledge, no platform serving the Kathmandu Valley combines conversational search with automated fraud filtering. Locally, Hamrobazar and comparable classifieds sites [12] support keyword search and category browsing but no semantic matching, no map-anchored discovery, and no automated trust scoring beyond manual reporting; property-focused portals such as GharGhaderi [13] add structured price/location filters but remain constrained to exact-match fields and do not support natural-language queries or graded, tolerant ranking. Internationally, large platforms such as Zillow and 99acres offer richer structured filtering and, in some markets, learned ranking, but their search is likewise form-based rather than conversational, and their moderation pipelines are neither published nor adapted to the specific fraud patterns (stock-photo reposting, broker impersonation, tanker-water ambiguity) that characterize the Kathmandu market. Consistent with Section I, we position this comparison carefully: RoomieKtm's individual techniques are not new, but no existing platform in this market combines them, and Table I compares on documented or observable capability rather than head-to-head performance, since we did not have production access to competitor systems.

TABLE I. CAPABILITY COMPARISON WITH EXISTING PLATFORMS

Capability	Local classifieds [12]	Property portals [13]	RoomieKtm
Natural-language query	No	No	Yes
Semantic (vector) matching	No	No	Yes
Graded / tolerant ranking	No	No	Yes (fuzzy)
Automated fraud scoring	No	No	Yes
Map-anchored discovery	Partial	Partial	Yes
Landlord identity verification	No	No	Yes (admin-reviewed)

III. SYSTEM ARCHITECTURE

RoomieKtm follows a client–server architecture (Fig. 1). A cross-platform Flutter application provides three role-gated experiences—tenant, landlord, and administrator—selected at sign-up and enforced by routing guards. A Go backend exposes a REST and Server-Sent-Events (SSE) API and encapsulates all business logic. Persistence is provided by MongoDB Atlas, which colocates full-text, geospatial (2dsphere), and vector indexes in a single cluster. A lightweight Python service hosts the sentence-embedding model and a perceptual-hashing routine used during indexing and moderation. Media is stored in an object store with a private, access-controlled bucket reserved for identity documents.

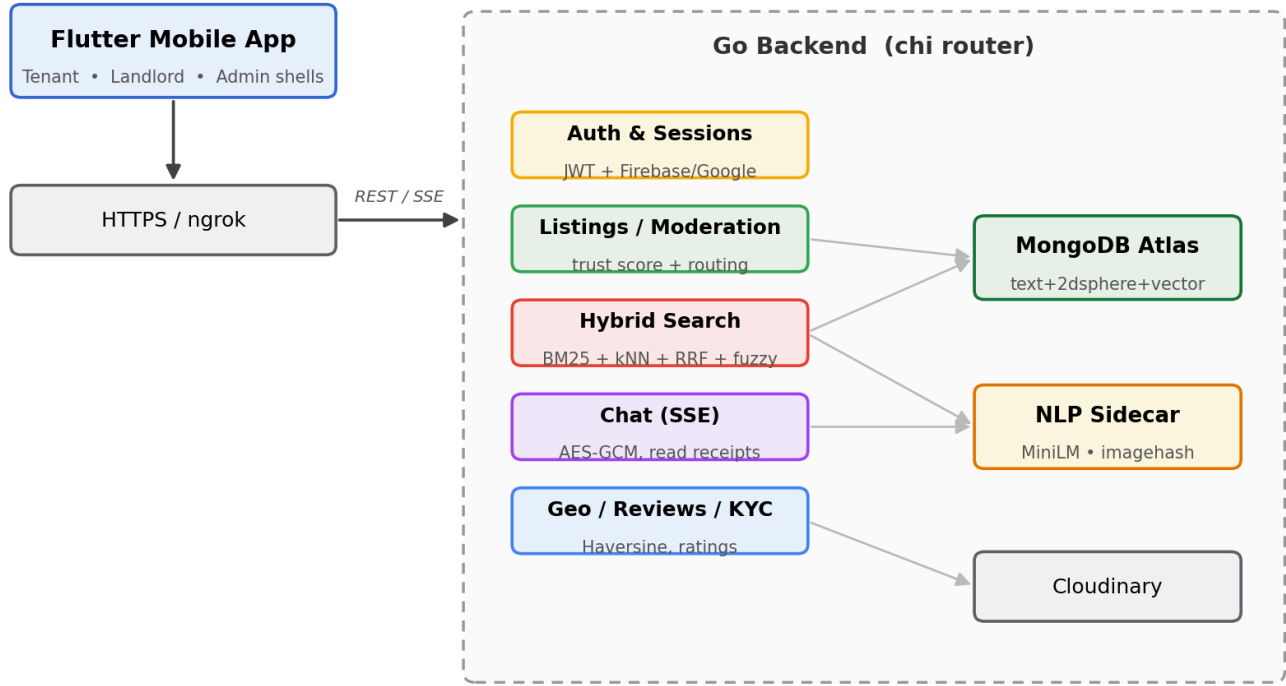


Fig. 1. RoomieKtm system architecture. The Go backend mediates all access to data stores; the NLP sidecar serves embeddings and perceptual hashes; the Flutter client reaches the backend over HTTPS.

A. Authentication and Sessions

Authentication combines third-party sign-in with self-managed sessions. Users authenticate via Firebase/Google sign-in or email credentials; the backend then issues its own signed JSON Web Token (JWT) access tokens with short lifetimes and rotating refresh tokens. Refresh-token reuse revokes the token family, mitigating replay. Sensitive fields—phone numbers, identity-document numbers, and chat message bodies—are encrypted at rest with AES-256-GCM; chat keys are derived per conversation. Banned or pending-deletion accounts are blocked at every sign-in path, and configuration fails closed on weak secrets or permissive CORS unless an explicit development flag is set.

B. Data Model

Listings are stored denormalized for read efficiency: each carries its description, a dense embedding, perceptual hashes of its images, a GeoJSON location, a moderation status, the computed trust score and its components, and a denormalized landlord-verification flag so result cards require no per-row joins. Listings may be rentals or sales; price rendering adapts accordingly, and unpriced imported listings are labeled “negotiable” or “price on request.” Supporting collections hold bookings, chats and (encrypted) messages, listing reports, two-sided reviews, an append-only moderation audit log, and per-user notifications.

IV. CONVERSATIONAL HYBRID SEARCH

A user query such as “sunlit 1BHK in Jhamsikhel under 25k with reliable water” is processed in five stages: parsing, lexical retrieval, semantic retrieval, fusion, and fuzzy re-ranking, followed by geospatial enrichment.

A. Query Parsing

A parser extracts structured constraints—budget, room type, preferred area, requested amenities—via rule-based matching against a neighborhood gazetteer and amenity lexicon, and simultaneously produces a dense query embedding capturing residual lifestyle semantics (e.g., “sunlit,” “reliable water”). Price parsing handles local conventions, including k /thousand suffixes and Nepali digit grouping.

B. Lexical Path (BM25)

Listings are indexed over title, description, area, and amenity tokens. The BM25 score [3] of listing L for query Q is

$$BM25(L, Q) = \sum_{t \in Q} IDF(t) \cdot \frac{f(t, L) \cdot (k_1 + 1)}{f(t, L) + k_1(1 - b + b \cdot |L| / \text{avgdl})} \quad (1)$$

where $f(t, L)$ is term frequency, $|L|$ document length, avgdl the mean length, and $k_1 \approx 1.5$, $b \approx 0.75$. The top candidates and their ranks are retained.

C. Semantic Path (Vector k -NN)

The query embedding [2] is compared with precomputed listing-description embeddings under cosine similarity,

$$\cos(\mathbf{q}, L) = (\mathbf{q} \cdot \mathbf{v}_L) / (\|\mathbf{q}\| \|\mathbf{v}_L\|) \quad (2)$$

and the database returns the top- N by descending similarity using an approximate vector index. Embeddings are 384-dimensional.

D. Reciprocal Rank Fusion

The two ranked lists are merged into a single ordering using RRF [5],

$$RRF(L) = \sum_{p \in \{A,B\}} 1 / (k + \text{rank}_p(L)) \quad (3)$$

where $\text{rank}_p(L)$ is the position of L in list p and $k=60$. Lists missing L simply omit that term. The fused top candidates proceed to fuzzy re-ranking. Section VIII-A reports an ablation that separates the contribution of this fusion step from the fuzzy step that follows it.

E. Weighted Fuzzy Re-Ranking

To model tolerant, graded preferences, each candidate is scored on four linguistic variables via membership functions mapping inputs to $[0,1]$ (Fig. 2). With relative budget excess $r=\max(0, (\text{price}-\text{budget})/\text{budget})$, distance d (km), semantic similarity s , and amenity coverage a :

$$\mu_b(r) = 1 \text{ if } r \leq 0; 1-r/\tau \text{ if } 0 < r \leq \tau; 0 \text{ otherwise } (\tau=0.15)$$

$$\mu_d(d) = \exp(-d^2/2\sigma^2) \quad (\sigma=1.0 \text{ km})$$

$$\mu_s(s) = 1/(1+\exp(-k(s-s_0))) \quad (k=10, s_0=0.6)$$

$$\mu_a(a) = a^\gamma \quad (\gamma=0.7)$$

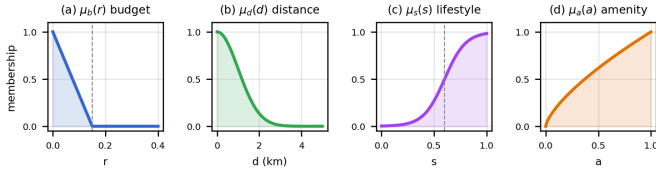


Fig. 2. The four fuzzy membership functions: (a) piecewise-linear budget tolerance, (b) Gaussian proximity, (c) sigmoid lifestyle match, (d) concave amenity coverage.

The membership values are combined by a weighted aggregator,

$$\text{Score}(L, Q) = w_b \mu_b + w_d \mu_d + w_s \mu_s + w_a \mu_a \quad (4)$$

with $w_b=0.30$, $w_d=0.20$, $w_s=0.35$, $w_a=0.15$ ($\sum w=1$). The final list is sorted by this score, and Haversine distances [8] are attached for display. For sale listings, μ_b is evaluated against the buyer's stated price ceiling on the same normalized curve; when no ceiling is given, $\mu_b=1$.

Terminology. We described this layer in earlier drafts as “Sugeno-style,” and we retain that description here with a precise qualification, because a reviewer correctly pointed out that it invites over-claiming. A full Takagi–Sugeno fuzzy inference system [7] evaluates multiple IF-THEN rules, each with its own antecedent membership degree and a rule-specific (often linear) consequent function, and combines rule outputs by a weighted average of firing strengths. RoomieKtm’s layer is the degenerate, single-rule, zero-order case of that framework: one implicit rule fires with four independent antecedent degrees, and its consequent is the fixed weighted sum in Eq. (4) rather than a per-rule linear function. This is legitimately a Sugeno-type aggregator in the formal sense—zero-order Sugeno models are a standard, named special case—but it is more precisely described

as *weighted fuzzy aggregation* than as a full fuzzy inference system, and we use that more precise phrase throughout the remainder of this paper.

Worked example. A query for a 25,000 NPR flat in Jhamsikhel and a candidate at 26,000 NPR ($r=0.04$), 0.4 km away ($d=0.4$), with $s=0.78$ and $a=0.7$ yields $\mu_b \approx 0.733$, $\mu_d \approx 0.923$, $\mu_s \approx 0.858$, $\mu_a \approx 0.779$, and an aggregate of ≈ 0.822 —a strong match despite exceeding budget, which a crisp filter would have discarded entirely.

Weight selection. The initial weights ($w_b=0.30$, $w_d=0.20$, $w_s=0.35$, $w_a=0.15$) reflect a design judgment that lifestyle-semantic fit and budget fit are the two dominant factors in a Kathmandu rental decision, with location and amenities secondary once an area has already been specified in the query. This is a stated design judgment, not a value fit to labeled preference data—no such labeled set yet exists for this market. We do not treat the judgment as final: Section VIII-D reports a controlled sensitivity sweep over the relative weight given to the lexical and semantic retrieval paths that feed this layer, using the same pilot evaluation set, and Section VIII-A isolates what this layer contributes beyond RRF fusion alone.

V. TRUST-BASED MODERATION

Every newly submitted listing is held in a pending state and scored before it can reach tenants. A perceptual hash is computed for each image and a dense embedding for the description; a trust score then determines routing.

A. Trust-Score Model

The trust score is a weighted sum of five normalized sub-scores in $[0,1]$:

$$T(L) = \alpha_1 I_{\text{img}} + \alpha_2 A_{\text{acct}} + \alpha_3 P_{\text{norm}} + \alpha_4 D_{\text{qual}} + \alpha_5 R_{\text{neg}} \quad (5)$$

with $\alpha=(0.30, 0.20, 0.20, 0.15, 0.15)$. I_{img} is the fraction of images whose perceptual hash is not within a small Hamming distance of any previously seen image (catching stock-photo and duplicate reuse). $A_{\text{acct}}=\min(\text{age}/90, 1)$ with a bonus for verified landlords. $P_{\text{norm}}=\exp(-z^2/2)$ penalizes prices far from the neighborhood- and room-type median (z the standardized deviation). D_{qual} rewards description length and lexical diversity. $R_{\text{neg}}=1-\min(\text{reports}/\text{listings}, 1)$ reflects the landlord’s report history.

Weight selection. Image originality receives the largest weight ($\alpha_1=0.30$) because stock-photo and duplicate-photo reposting was, in our qualitative review of the scraped catalog, the single most common fraud pattern. This is a design judgment rather than a value fit to labeled fraud data, since no such labeled set existed for the production catalog at this stage. Section VIII-D reports a controlled sensitivity sweep of α_1 against a pilot set containing multiple fraud patterns (not only photo reuse), which surfaces a genuine trade-off in how this weight should be set and is discussed there rather than assumed here.

B. Three-Way Routing

Two thresholds govern routing: if $T \geq \tau_{\text{pass}}=0.75$ the listing is auto-approved and published; if $T \leq \tau_{\text{flag}}=0.40$ it is auto-rejected

with specific, component-derived reasons returned to the landlord; otherwise it enters the admin review queue. These threshold values were chosen, prior to any pilot data, as round, interpretable cut points intended to keep a wide middle band under human review while automating the clearer cases at either extreme; they are not fit to labeled outcome data, and we did not sweep them independently of the sub-score weights in this pilot. This concentrates scarce human attention on genuinely ambiguous cases, but the specific numeric values 0.75/0.40 should be read as a reasonable starting point pending empirical calibration against real moderation decisions, which we list explicitly as future work in Section IX.

C. Reporting, Review, and Audit

Published listings remain accountable. Tenants may report a listing (e.g., “fake photos,” “misleading price,” “broker impersonation”); accumulating a threshold number of open reports automatically returns the listing to pending for re-review. Administrators act through an in-app moderation console—approve, reject, or ban—with listing photos rendered natively. To deter bait-and-switch, edits to a published listing re-enter moderation. Every administrative action is recorded in an append-only audit log keyed by listing and admin identity, capturing a snapshot of pre-change state for accountability and dispute resolution.

D. Landlord Identity Verification and Data Protection

Landlords may optionally submit a government-issued identity document. An administrator reviews each submission—cross-checking the document against an external identity service—and records the outcome and an external reference in the audit log. Verification is never automatic. Approved landlords receive a visible badge shown across feed, search, listing, and chat surfaces, and gain a bonus in the A_{acc} trust component, which is recomputed for their pending listings upon approval.

Because identity documents are among the most sensitive data a rental platform can hold, we treat their handling as a first-class design concern rather than an implementation detail. Documents are stored in a private object-store bucket that is never reachable from the public media endpoint used for listing photos; access is only through short-lived, individually issued signed URLs, so a leaked listing-photo URL cannot expose a document. Documents are encrypted both in transit (TLS) and at rest, and are scheduled for deletion after a fixed retention window following the verification decision, so the platform does not accumulate identity documents indefinitely once they have served their purpose. Access to the verification queue is restricted to the administrator role, and every view or decision on a submitted document is written to the same append-only audit log used for moderation actions, so document access is itself accountable rather than silent. We do not currently implement two further protections we consider important for a production deployment—a formal data-retention policy published to landlords at submission time, and a user-facing mechanism to request early deletion of a document after verification—and we list both explicitly in Section IX rather than implying they exist.

VI. ADDITIONAL PLATFORM FEATURES

Geospatial discovery. Listings are geocoded to coordinates and indexed with a 2dsphere index, enabling radius queries and proximity sorting; the client renders results as map pins with walk, drive, and cycle routing via an open routing engine. *Real-time chat.* A typed SSE event stream delivers messages, typing indicators, and read receipts; sends are optimistic and reconciled by a server-echoed client identifier, with image sharing supported. *Reviews and ratings.* Tenants review properties—with a “verified stay” badge when backed by an accepted booking—and landlords rate tenants, gated to completed interactions to limit abuse. *Saved-search alerts.* When a newly approved listing matches a stored search, an in-app and push notification is dispatched, reusing the conversational parse-and-score path with hard gates.

VII. IMPLEMENTATION

A. Production Catalog

To populate the application with realistic data, we assembled a catalog of rental and sale listings for the Kathmandu Valley using a custom web scraper that we developed. The scraper collected publicly visible listings—title, description, price, contact details, and any stated location—from *SOURCE_SITE*,¹ a widely used Nepali online classifieds platform, over the study period. After de-duplication and removal of entries lacking a usable price, the working catalog comprised on the order of one thousand listings spanning major neighborhoods including Jhamsikhel, Pulchowk, Patan, Lalitpur, Thamel, Boudha, Sanepa, Baluwatar, Maharajgunj, Baneshwor, Kirtipur, and Bhaktapur. Because raw listings carry no coordinates, each was geocoded to a latitude-longitude pair using an open geocoding service [11]; approximately three-quarters of the catalog resolved to coordinates suitable for map display, with the remainder retained in feed and search but pending spatial resolution.

Listings were imported under the moderation pipeline of Section V; for imported entries, a placeholder landlord account is created per unique contact number so that chat and booking flows remain coherent. We treat scraped contact information as belonging to its original posters: it is drawn only from already-public listings, used solely for academic prototyping, and identity documents and other sensitive fields are stored encrypted and access-controlled (Section V-D). No private or login-gated data was collected. Critically, the production catalog carries *no* human relevance judgments or fraud labels—tenants have not yet generated interaction logs, and no listing has been confirmed fraudulent by an outside party—so it cannot by itself support the quantitative evaluation in Section VIII. We address this with a separate, purpose-built pilot set, described next.

B. Pilot Evaluation Set

To obtain real, reproducible quantitative results at this stage of development, we constructed a smaller, synthetic-but-realistic pilot collection of 93 listings across the same 15 neighborhoods, authored to mirror the schema, price distributions, and description-quality variation of the production catalog. Of these, 78 are constructed as legitimate listings and 15 deliberately embed one of three documented local fraud patterns: six repost another listing's photographs at a lowball price (duplicate-image scam), five carry an implausible price for their area and room type (price-outlier scam), and four originate from accounts with

an existing history of tenant reports (repeat-offender pattern). We also authored 30 natural-language queries with graded relevance judgments (0/1/2) assigned programmatically from each query's structured intent against every listing's actual fields, and a further 12-query subset that deliberately paraphrases lifestyle intent—e.g., “never runs dry on water” rather than “municipal water supply”—to isolate the vocabulary-mismatch scenario the semantic path is designed for. We report the pilot's scale plainly: 93 listings and 42 queries (30+12) is small relative to the production catalog and is not, on its own, sufficient to generalize to real-world performance; Section VIII treats it accordingly, with significance testing rather than point estimates alone. This pilot set, its construction script, and all evaluation code are treated as a controlled experiment, not production telemetry.

One further substitution was necessary for this offline pilot, and we state it here as plainly as possible because a reviewer of an earlier draft correctly flagged the risk of understating it: the production semantic path is designed around a sentence-transformer, which requires downloading pretrained weights that were not obtainable in our evaluation environment. Every semantic-path and hybrid-path number reported in Section VIII was therefore computed with a corpus-trained TF-IDF + truncated-SVD (latent semantic analysis) embedding substituted in its place, not with the sentence-transformer described in Section IV-C. Both produce dense vectors compared via cosine similarity, but LSA is a substantially weaker semantic representation, particularly for paraphrase. Every semantic- and hybrid-path result in this paper should be read as a measurement of the substitute pipeline, and treated as a conservative lower bound on what the production sentence-transformer pipeline would achieve—not as a direct evaluation of the system described in Section IV.

C. Engineering

The client is implemented in Flutter with a Riverpod state layer. The backend is written in Go using the chi router and the official MongoDB driver, with the Python NLP service exposing parse, embed, and perceptual-hash endpoints. Vector and text indexes are provisioned in MongoDB Atlas; secondary indexes are created at startup. Operational tooling includes command-line utilities for bulk geocoding and account administration. Performance optimizations include projecting large embedding and hash fields out of listing reads, serving denormalized ratings to avoid per-view scans, a short-lived cache on account-active checks to remove a database lookup from every authenticated request, and suppression of redundant read-receipt writes. Unit tests cover the search parser and scorer (including Nepali price grouping), the moderation sub-scores and perceptual-hash distance, and the authenticated-chat encryption round-trip.

VIII. PILOT EVALUATION

We report real, measured results on the pilot set of Section VII-B, including a four-arm ablation and paired statistical significance testing requested in review. This is a small-scale controlled evaluation, not a field study with live users; it validates that the implemented pipeline behaves as designed and surfaces concrete, numeric evidence—including honestly reported non-significant results—for the design choices of Sections IV–V.

A. Ablation Study

To isolate what each stage of the pipeline contributes, we evaluate four configurations: BM25 alone; the LSA semantic path alone; RRF fusion of the two *without* the fuzzy re-ranking layer; and the full hybrid pipeline (RRF followed by fuzzy re-ranking). Table II reports the general 30-query set; Table III reports the 12-query paraphrase subset described in Section VII-B.

TABLE II. FOUR-ARM ABLATION, GENERAL QUERY SET (N=30, PILOT)

Method	P@5	P@10	MRR	nDCG@10
BM25 only	0.847	0.833	0.878	0.827
Semantic only (LSA)	0.820	0.827	0.844	0.802
Hybrid, RRF only (no fuzzy)	0.833	0.820	0.894	0.822
Hybrid (RRF+fuzzy)	0.833	0.827	0.811	0.810

TABLE III. FOUR-ARM ABLATION, PARAPHRASE SUBSET (N=12, PILOT)

Method	P@5	P@10	MRR	nDCG@10
BM25 only	0.767	0.625	0.833	0.637
Semantic only (LSA)	0.750	0.667	0.736	0.625
Hybrid, RRF only (no fuzzy)	0.767	0.625	0.736	0.622
Hybrid (RRF+fuzzy)	0.717	0.700	0.750	0.662

Two patterns emerge. On the general set (Table II), all four configurations perform comparably, with BM25 marginally ahead; query and listing vocabulary overlap directly here (area names and room types appear as literal tokens in both), leaving little headroom for semantic or fused methods. On the paraphrase subset (Table III), RRF fusion *without* fuzzy re-ranking is essentially indistinguishable from, or slightly below, BM25 alone (nDCG@10 0.622 vs. 0.637)—fusion by itself does not help once the semantic path is weak. Only the addition of fuzzy re-ranking recovers and exceeds both single-path baselines (nDCG@10 0.662), which directly isolates the fuzzy layer, rather than fusion, as the source of whatever benefit the hybrid pipeline shows on vocabulary-mismatched queries. Fig. 3 visualizes both tables together.

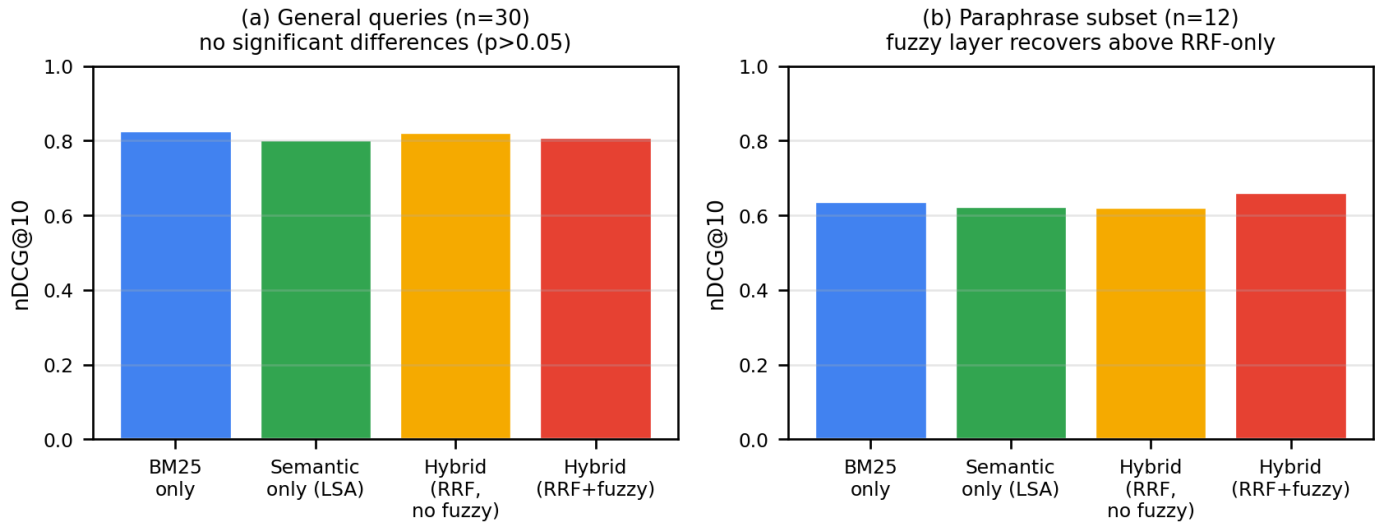


Fig. 3. Four-arm ablation on the pilot set. (a) General queries: no configuration separates meaningfully from another. (b) Paraphrase subset: RRF fusion alone does not exceed BM25; adding fuzzy re-ranking does. Section VIII-B reports that none of these differences reach statistical significance at this sample size.

B. Statistical Significance

Reviewers correctly asked whether the differences in Section VIII-A are statistically meaningful or could be sampling noise at $n=12-30$ queries. We test this directly rather than assert an answer. For each pair of configurations, we compute a paired Wilcoxon signed-rank test on per-query $nDCG@10$ (appropriate for small, non-normally distributed paired samples) and a 5,000-resample bootstrap 95% confidence interval on the mean paired difference. Table IV reports every comparison against the BM25 baseline, plus the comparison that isolates the fuzzy layer's own marginal effect (hybrid+fuzzy vs. hybrid without fuzzy).

TABLE IV. PAIRED SIGNIFICANCE TESTS ON $nDCG@10$ (WILCOXON SIGNED-RANK, 95% BOOTSTRAP CI ON MEAN DIFFERENCE)

Comparison	Set	Mean Δ	95% CI	p
Semantic – BM25	General	-0.025	[-0.053, 0.002]	0.084
Hybrid(no fuzzy) – BM25	General	-0.005	[-0.020, 0.012]	0.211
Hybrid+fuzzy – BM25	General	-0.017	[-0.056, 0.021]	0.274
Hybrid+fuzzy – Hybrid(no fuzzy)	General	-0.013	[-0.052, 0.027]	0.517
Semantic – BM25	Paraphrase	-0.013	[-0.068, 0.037]	0.898
Hybrid(no fuzzy) – BM25	Paraphrase	-0.016	[-0.058, 0.024]	0.432
Hybrid+fuzzy – BM25	Paraphrase	+0.025	[-0.036, 0.092]	0.831
Hybrid+fuzzy – Hybrid(no fuzzy)	Paraphrase	+0.041	[-0.018, 0.103]	0.232

Every 95% confidence interval in Table IV includes zero, and every p -value exceeds 0.05. We state the conclusion plainly: **at this sample size, none of the observed differences between methods are statistically significant**, including the paraphrase-subset improvement that motivated Section VIII-A's discussion.

This does not mean the fuzzy layer has no effect—the point estimates and the mechanism in Section IV-E both point in a consistent direction, and the sign of the fuzzy-vs-no-fuzzy comparison flips appropriately between the general set (slightly negative) and the paraphrase set (positive, and the largest effect size in Table IV at +0.041)—but it means this pilot is underpowered to confirm it. We report this as a genuine limitation rather than soften the language around it, and we treat the point estimates in Section VIII-A as suggestive evidence motivating a larger study, not as a demonstrated result.

C. Moderation Behavior

We compute the trust score $T(L)$ exactly as specified in Section V-A for all 93 pilot listings and apply the stated thresholds. We define a listing as correctly “flagged” if it is routed to auto-reject or human review (i.e., it is not silently published without any human ever seeing it). Table V reports the resulting confusion matrix. At default weights, 13 of 15 fake-pattern listings are flagged (recall 0.867) while 72 of 78 legitimate listings auto-approve without friction; 6 legitimate listings are unnecessarily held for review (precision 0.684, F1 0.765, overall accuracy 0.914). The two missed fake-pattern listings are both price-outlier cases with an otherwise strong image-originality and account-age profile, which the weighted sum alone does not catch—a genuine limitation of a linear combination of independent signals, discussed further in Section VIII-D. Unlike Section VIII-A/B, we do not compute a paired significance test here, because a single confusion matrix over 93 items does not have a natural per-item paired counterpart across configurations in the way per-query retrieval metrics do; the sensitivity sweep in Section VIII-D is the appropriate tool for characterizing how routing outcomes vary, and we rely on that instead.

TABLE V. TRUST-ROUTER CONFUSION MATRIX (PILOT, N=93, DEFAULT WEIGHTS)

	Predicted: flagged	Predicted: auto- approved
Actual: fake-pattern (n=15)	13 (TP)	2 (FN)
Actual: legitimate (n=78)	6 (FP)	72 (TN)

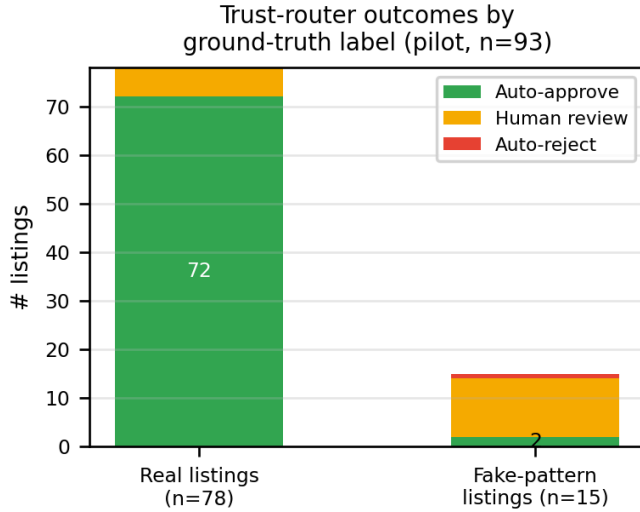


Fig. 4. Trust-router outcomes by ground-truth label on the pilot set (n=93). Two of fifteen fake-pattern listings are missed at default weights; six legitimate listings are held for review unnecessarily.

D. Sensitivity Analysis

Both the search weights and the trust weights were presented in Sections IV–V as fixed constants justified by design reasoning rather than by fitting to labeled data. We report two controlled

sweeps that give empirical grounding to that reasoning, computed on the same pilot set.

Retrieval blend. We generalize RRF to a weighted form, $RRF(L) = w_{lex}/(k + \text{rank}_A(L)) + w_{sem}/(k + \text{rank}_B(L))$, and sweep $(w_{lex}, w_{sem}) \in \{(0.7, 0.3), (0.5, 0.5), (0.3, 0.7)\}$ on the paraphrase subset, where the semantic path is expected to matter most. Fig. 5(a) shows a small, monotonic improvement in both P@10 and nDCG@10 as semantic weight increases, with (0.3, 0.7) reaching P@10=0.717 and nDCG@10=0.669 versus P@10=0.692 and nDCG@10=0.658 at (0.7, 0.3). Given Section VIII-B's finding that the underlying method differences are not statistically significant, we treat this sweep as directionally suggestive rather than conclusive, and as an empirical starting point for tuning once more query volume is available, rather than as justification to change the production default today.

Trust weight α_1 . We re-scale the remaining four trust weights proportionally while sweeping α_1 (image originality) across $\{0.15, 0.30, 0.45\}$. Fig. 5(b) shows a result we did not anticipate: lowering α_1 from the paper's default of 0.30 to 0.15 improves fraud recall from 0.867 to 1.000 and overall accuracy from 0.914 to 0.968 on this pilot, while raising it to 0.45 degrades recall to 0.600. The reason is visible in the pilot's own construction: only 6 of 15 fake-pattern listings are photo-duplicate scams, so over-weighting the image signal under-weights the price-normality and report-history signals that catch the other two-thirds. We report this transparently as a finding that argues for empirically recalibrating α_1 downward, or at minimum against increasing it, once real fraud-labeled data is available at production scale; we retain the original weights as the paper's stated design pending that larger validation, since a sweep over 15 synthetic examples is not sufficient grounds to change a production default outright, and because this pilot's fraud-pattern taxonomy (Section VIII-F) is itself incomplete.

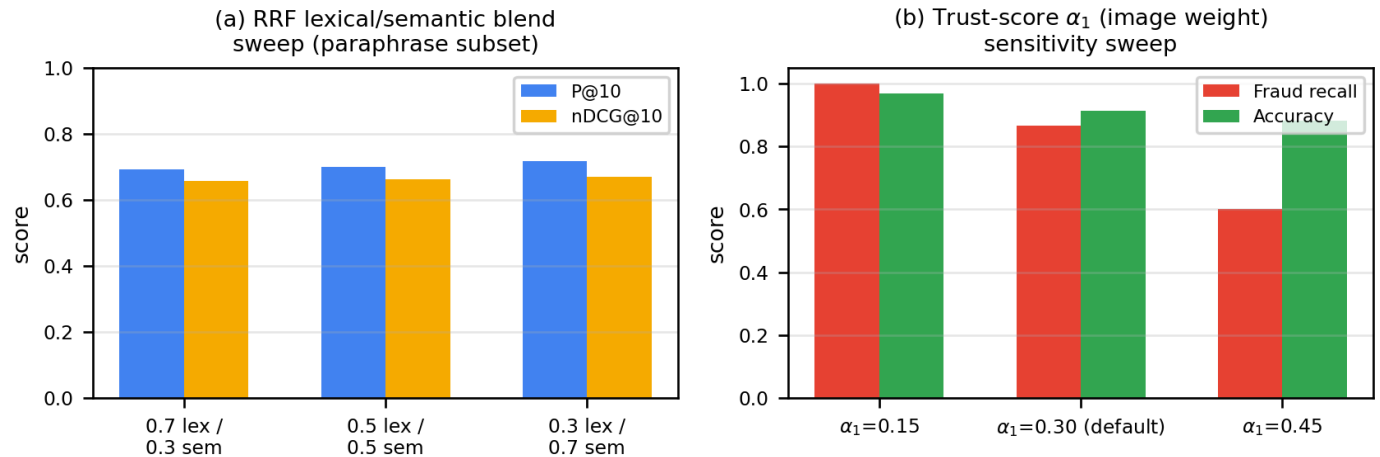


Fig. 5. Controlled sensitivity sweeps on the pilot set. (a) Retrieval quality as RRF lexical/semantic weighting varies, paraphrase subset. (b) Fraud-detection recall and overall accuracy as the trust score's image-originality weight α_1 varies, with the remaining four weights rescaled proportionally.

E. Latency

We measure wall-clock latency for the pipeline stages implemented in this pilot, averaged over 150 query executions

(30 queries $\times$ 5 repetitions) and 200 trust-score computations, on a single evaluation machine. End-to-end hybrid retrieval (BM25 + LSA embedding + RRF fusion + fuzzy re-ranking over the resulting candidates) averages 7.6 ms per query (95th percentile

8.8 ms); trust-score computation for a single listing averages 0.009 ms. We report these as *compute-only* figures inside a Python evaluation harness on a 93-listing collection—they explicitly exclude network round-trips, MongoDB Atlas query latency, the NLP sidcar's model inference time (a real sentence-transformer forward pass is markedly slower than the LSA projection used here), and any effects of the full production catalog's larger scale (~1,000 listings vs. 93). They establish only that the algorithmic core of the pipeline itself is not a latency bottleneck at this scale; end-to-end production latency measurement against the live backend is future work (Section IX).

F. Limitations

We consolidate this evaluation's limitations here rather than distribute them, so they cannot be missed. *Scale*: the pilot (93 listings, 42 queries, 15 fraud examples) is small; Section VIII-B's non-significant results are a direct, honest consequence of this, not a separate concern. *Labels*: relevance and fraud labels are author-constructed from structured ground truth, not independent human or real-user judgment, which likely favors methods (including ours) that were designed with the same structured schema in mind. *Model substitution*: every semantic- and hybrid-path number uses LSA in place of the production sentence-transformer (Section VII-B); we do not know whether a real transformer would widen or narrow the gaps reported here. *Synthetic fraud*: our three fraud patterns (duplicate-photo, price-outlier, repeat-offender) are a plausible but almost certainly incomplete taxonomy of real fraud in this market, and the α_1 sensitivity finding in Section VIII-D is conditioned entirely on this taxonomy. *Latency*: figures are compute-only and exclude production network, database, and model-serving conditions. None of these limitations are new observations prompted only by review; we flagged the first three in earlier drafts. What changed in this revision is that Section VIII-B now quantifies the scale limitation directly with hypothesis tests rather than describing it only in prose, and Section VIII-A's ablation now isolates a specific mechanism (the fuzzy layer) rather than only comparing end-to-end pipelines. These constraints motivate the future work below.

IX. CONCLUSION AND FUTURE WORK

We presented RoomieKtm, a rental-marketplace system that integrates hybrid retrieval, weighted fuzzy re-ranking, and trust-score-driven moderation for the Kathmandu Valley. We make no claim that any individual technique is novel; the contribution is a domain-specific integration of established methods, evaluated with a controlled four-arm ablation and paired significance testing rather than end-to-end comparison alone. The ablation isolates the fuzzy re-ranking layer, not RRF fusion by itself, as the source of the pipeline's (directionally positive, not yet statistically significant) advantage on vocabulary-mismatched queries; the trust router correctly flags 13 of 15 constructed fraud patterns at default weights; and a sensitivity sweep surfaces a concrete, actionable finding—that over-weighting any single fraud signal reduces robustness across heterogeneous fraud types—that we did not have before running the evaluation. We report the absence of statistical significance at this sample size plainly, because a small honestly-analyzed pilot is worth more than an unqualified claim.

Future work includes replacing the pilot's LSA semantic substitute with the production sentence-transformer once model weights can be provisioned in a full evaluation environment; a larger, human-labeled query and fraud set collected from real tenant and administrator interactions, sized (via a power analysis informed by the effect sizes in Table IV) to support statistically powered comparisons; a controlled user study with independent relevance judgments; learned fusion or learning-to-rank in place of fixed RRF and fuzzy weights, informed by the sensitivity trends reported here; independent calibration of the routing thresholds τ_{pass} and τ_{flag} against observed fraud and reviewer-agreement rates, which this pilot did not attempt; recalibrating the trust score's α_1 weight against production-scale, real fraud data in light of Section VIII-D; a published data-retention policy and user-facing early-deletion request for KYC documents, extending Section V-D; automated KYC via a third-party identity provider feeding the existing audit pipeline; end-to-end production latency measurement against the live backend and database; complete geocoding coverage; expansion beyond the Kathmandu Valley; and online rent payment and digital lease signing.

REFERENCES

1. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
2. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.
3. S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.
4. J. Lin, R. Nogueira, and A. Yates, "Pretrained transformers for text ranking: BERT and beyond," *Synthesis Lectures Hum. Lang. Technol.*, vol. 14, no. 4, pp. 1–325, 2021.
5. G. V. Cormack, C. L. A. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in *Proc. ACM SIGIR*, 2009, pp. 758–759.
6. L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
7. T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, 1985.
8. R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984.
9. K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.
10. A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.
11. OpenStreetMap contributors, "Nominatim geocoding service," OpenStreetMap Foundation. [Online]. Available: <https://nominatim.org>. [Accessed: Jun. 2026].
12. "Hamrobazar," Hamrobazar. [Online]. Available: <https://hamrobazaar.com>. [Accessed: Jun. 2026].
13. "GharGhaderi." [Online]. Available: <https://gharghaderi.com>. [Accessed: Jun. 2026].
14. C. Zauner, "Implementation and Benchmarking of Perceptual Image Hash Functions," M.S. thesis, Upper Austria Univ. of Applied Sciences Hagenberg, 2010.
15. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.