

LLM-Based Re-Ranking for Real Estate Search

Nkateko Ntimane
nkateko.ntimane@quintoandar.com.br
QuintoAndar
Lisbon, Portugal

Tiago Cunha*
tiago.cunha@growthloop.com
Growthloop
Portugal

Rafael Guedes
rafael.guedes@quintoandar.com.br
QuintoAndar
Lisbon, Portugal

Pedro Nogueira
pedro.nogueira@quintoandar.com.br
QuintoAndar
Lisbon, Portugal

Abstract

QuintoAndar Group operates the leading housing marketplace in Latin America for both rentals and sales. The platform replaces traditionally paper-heavy workflows with a fully digital experience, making housing transactions faster and more accessible to tenants, buyers, and landlords in the region. Finding the ideal home in such a vast catalog is inherently difficult. At the same time, the widespread adoption of conversational assistants is reshaping user expectations: people increasingly want to express their needs through open, multi-turn dialog rather than rigid filter menus and faceted search. This shift is particularly pronounced in housing, where intent is multi-dimensional, context-dependent, and rarely reducible to a small set of structured constraints. To meet these expectations, we propose a Large Language Model (LLM) based re-ranker that augments a conversational recommendation system by reordering retrieved candidates according to the nuanced, context-rich intent expressed across the user's conversation. We additionally construct a large-scale offline evaluation dataset for conversational real-estate search, containing 960,000 query-item pairs constructed from both synthetic and production queries and annotated using an LLM-as-a-Judge framework with human validation. We validate our approach both offline, on this proprietary dataset, and online, through a production A/B test. Both evaluations show consistent improvements in ranking quality, including a statistically significant increase in production of +5.3% in click-through rate and +4.8% in scheduled visits, demonstrating the value of integrating conversational context into housing recommendations.

CCS Concepts

• **Computing methodologies** → *Machine learning*; • **Information systems** → **Recommender systems**; **Retrieval models and ranking**; **Users and interactive retrieval**.

*Work performed while the author was employed at QuintoAndar.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
RecSys '26, Minneapolis, USA

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
<https://doi.org/XXXXXXX.XXXXXXX>

Keywords

Conversational Information Retrieval, LLM-based Re-ranking, Agentic Workflows, Synthetic Data Augmentation, Real Estate Search

ACM Reference Format:

Nkateko Ntimane, Rafael Guedes, Tiago Cunha, and Pedro Nogueira. 2026. LLM-Based Re-Ranking for Real Estate Search. In *Proceedings of Twentieth ACM Conference on Recommender Systems (RecSys '26)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

The widespread adoption of LLM-powered conversational assistants is reshaping how users interact with online services [15]. Increasingly, people expect to express their needs in natural language, through open and incremental dialog, rather than through rigid forms or filter widgets. This shift has profound implications for recommender systems, particularly in domains where user intent is dense, qualitative, and discovered over time.

Real estate is one such domain. Finding a home is a high-stakes, infrequent decision in which preferences span lifestyle considerations (e.g., proximity to a workplace, sun exposure, neighborhood character), conditional trade-offs (e.g., willingness to exceed a budget for a specific amenity), and qualitative attributes that have no clean representation in a structured catalog. These signals are essentially invisible to traditional faceted search, leading systems to surface listings that satisfy the explicit constraints of a query but miss the user's underlying intent.

At QuintoAndar, this conversational shift has materialized as Concierge, a multi-agent assistant that handles property discovery, listing inquiries, and visit scheduling through natural-language dialog. However, a natural-language interface only results in better recommendations if the ranking stage can take advantage of the additional context provided by the user. Retrieval over structured filters can produce a reasonable candidate-set, but it cannot distinguish between properties that nominally match the same constraints but differ substantially in how well they align with the user's broader goals. Although conversational understanding can be injected at multiple stages of the recommendation pipeline, we focus on the re-ranking stage, which offers a pragmatic balance for production deployments: it operates on a small, latency-tolerant candidate-set, integrates seamlessly with existing high-throughput retrieval infrastructure, and can reason holistically over hard constraints, soft preferences, and relative value.

This work presents the design, deployment, and evaluation of an LLM-based re-ranker integrated into Concierge. The main contributions are as follows:

- (1) An **LLM-based point-wise re-ranker** that scores candidates using an LLM-based user profile and aggregate statistics over the full candidate-set, enabling relative comparisons under a strict production latency budget.
- (2) A **comprehensive empirical study** combining an offline evaluation on a proprietary dataset of synthetic and production queries, annotated via LLM-as-a-Judge with a production A/B test, both showing consistent gains in ranking quality.

In addition, we describe how we created a large-scale offline benchmark for conversational real-estate search, constructed from both synthetic and real-world production queries. The dataset captures diverse search behaviors, including conversational preferences, trade-offs, lifestyle requirements, and persona-driven constraints, enabling systematic evaluation of ranking models under realistic conditions.

Our work shows that LLMs can improve search quality in a production recommendation system without requiring end-to-end generative retrieval or a full rewrite of the serving stack. By applying the model only at the re-ranking stage over a compact candidate-set, the approach preserves the latency profile required by interactive search while improving both click-through rate and downstream engagement metrics.

The remainder of the paper is organized as follows. Section 2 surveys related work on LLM-based recommendation, with a focus on re-ranking and conversational systems. Section 3 describes the proposed user search profile and point-wise LLM re-ranker. Section 4 details the experimental setup, including dataset construction, evaluation protocols and results discussion. Section 5 concludes with a summary of findings and directions for future work.

2 Related Work

LLMs have expanded the design space of Recommender Systems (RSs) by enabling natural-language query understanding, richer user-intent modeling, and conversational interaction [18].

2.1 From Conversational RS to Agentic RS

Traditional Conversational RSs were defined by multi-turn dialogues where the system explicitly asked questions to elicit user preferences, provide explanations or process the user feedback on the provided recommendations [14]. Earlier approaches focused mainly on critique-based systems, where critiques ("cheaper" or "closer to the center") were used to refine recommendations [11]. Other methods used slot-filling systems that guided users through predefined dialogue paths to acquire specific attributes for recommendation, often constrained by rigid taxonomies [27]. More recently, neural architectures have been devised to move away from manual dialogue engineering to models trained on large corpora leveraging Reinforcement Learning [29].

Despite these advances, a gap remains between traditional task-oriented chatbots and systems capable of truly conversational recommendation. This has led to the emergence of Agentic RSs, where

the LLM acts as an autonomous reasoning entity, manages the dialogue state implicitly and orchestrates calls to various tools to assist users [7].

Recent literature has explored the integration of LLMs into the recommendation pipeline with meaningful success. We categorize this integration into four key stages:

- **Query Understanding and Expansion:** At the entry point of the pipeline, LLMs serve to bridge the vocabulary mismatch between user natural language and item metadata. By transforming ambiguous queries into structured filters or expanded semantic queries, LLMs improve the precision of initial candidate retrieval [4, 17, 21, 23, 34].
- **User Profiling and Representation:** LLMs act as sophisticated feature extractors that distill user history and multi-modal behaviors into compressed, latent representations. This includes generating explicit and detailed user profiles in natural language [2, 16, 19] or generating dense embeddings from conversational traces to capture evolving preferences [9, 22, 30, 33, 37].
- **LLM Re-ranking:** A two-stage approach where a traditional retriever handles candidate generation, and an LLM acts as a ranker to refine the top- k results. This enables the integration of complex user intent and cross-attribute reasoning that exceeds the capabilities of standard retrieval functions [3, 5, 8, 12, 20].
- **Agentic Frameworks and Orchestration:** A broader form of integration, where the LLM orchestrates the entire process, deciding when to search, when to ask for clarification, and how to present the final recommendation [13, 25, 28, 32, 35].

Our work focuses on the re-ranking stage within a conversational real-estate assistant, showing that LLM-based scoring can use conversational context, structured filters, property metadata, and candidate-set statistics to improve ranking quality under production constraints.

2.2 LLM-Based Re-Ranking

Using LLMs in the re-ranking stage allows the system to leverage their deep semantic understanding to refine candidate-sets generated by traditional retrievers [18]. Unlike first-stage retrieval, which prioritizes efficiency, LLM-based re-ranking focuses on complex reasoning and cross-attribute evaluation.

Early work on LLM re-ranking focused on assessing the zero-shot capabilities of these models, i.e. without explicit training on recommendation data. Research has formalized the task as a conditional ranking problem where the LLM evaluates a list of candidates retrieved by an external model [12]. These studies identify three primary paradigms: point-wise, pair-wise, and list-wise ranking [5]. While LLMs demonstrate promising alignment with traditional Information Retrieval (IR) ranking, they are often susceptible to the order of items in the prompt or the item's frequency in training data [5, 12].

To overcome the limitations of zero-shot models, recent work emphasizes instruction tuning to align LLMs with specific recommendation objectives. The TALLRec framework demonstrates that efficient tuning can bridge the gap between general language tasks and the recommendation domain [3]. More specialized approaches

like RecRanker introduce instruction tuning specifically for the top- k list-wise re-ranking task, utilizing importance-aware sampling and position-shifting strategies to eliminate biases [20]. The LLM4Rerank framework abstracts different re-ranking objectives (such as accuracy, diversity, and fairness) into a graph-based structure, using Chain-of-Thought (CoT) reasoning to automatically balance these competing factors [8].

3 Methodology

This section presents the design of Concierge and its integration with an LLM-based re-ranker. The pipeline follows a two-stage architecture in which a retrieval stage produces a set of candidate properties, and a re-ranking stage refines the candidate set ordering through a point-wise LLM scoring function applied to query-property pairs.

3.1 Conversational Recommendation System

Concierge is implemented as a conversational, multi-agent framework that processes natural-language queries and returns personalized property recommendations. The framework comprises several domain-specific agents, including a recommendation agent for property discovery, a property details agent for retrieving listing-specific information, and a visits agent for handling scheduling workflows.

An LLM-based planner acts as the entry point for each user interaction: it classifies the user's intent and dispatches the request to the most appropriate domain-specific agent.

When the planner routes a query to the recommendation agent, the request enters the retrieval pipeline. A core component of this pipeline is Text2Filter, which converts free-text user queries into structured search filters. The supported filter types include location, price range, number of bedrooms, and many others. The service is implemented as a two-stage LLM pipeline based on GPT-4o (2024-11-20) [24]. In the first stage, an LLM identifies which filter attributes are present in the user query. In the second stage, structured values are generated for each identified attribute. Both stages operate against a predefined schema, ensuring the consistency and validity of the extracted filters.

The resulting filters are then used to retrieve and rank candidate properties using the existing search system, which serves as the baseline retrieval stage (Figure 2). To keep responses concise and easy to interpret in the conversational interface, only the top five ranked properties are returned.

3.2 LLM-based User Search Profile

The user search profile is a natural-language representation of the user's search intent. It is constructed by an LLM from three complementary sources: (i) a previously generated profile, when available, providing continuity across sessions; (ii) the most recent conversation with Concierge, capturing explicitly stated requirements; and (iii) a structured behavioral profile derived from historical interactions with listings [36]. The behavioral profile aggregates implicit preference signals over key attributes, such as location, price, property type, and amenities. It does so by modeling interaction events with different relevance weights according to their importance. For example, clicks are treated as weak engagement signals, whereas

favorites and scheduled visits are considered stronger indicators of user preference.

When signals across sources conflict, the model gives precedence to the most recent and most explicitly stated information in a conversation. A central design principle is to treat behavioral signals as soft preferences rather than hard constraints. Instead of converting structured values directly into search filters, the model expresses them as natural-language preferences that capture varying levels of confidence or intensity. This enables the profile to meaningfully distinguish between a weak behavioral tendency and a strong, consistent requirement.

The textual component captures information that is absent from the property catalog. For example, this includes qualitative dimensions such as lifestyle or neighborhood character, as well as conditional trade-offs, such as a user's willingness to exceed a stated budget for a specific amenity. Such nuanced signals are particularly important in real estate search, where user intent is often multi-dimensional and context-dependent.

The final output is a concise natural-language summary of the user's search intent together with a list of explicitly mentioned points of interest (Figure 1). By consolidating implicit behavioral signals and explicit conversational cues into a unified textual representation, the profile furnishes the re-ranker with rich contextual grounding. This enables the re-ranker to surface properties that satisfy persistent or conditional preferences even when those preferences are not directly reflected in the current query.

User Search Profile Example

The user is looking for a house, studio, or apartment to rent in São Paulo, specifically in the South Zone, with a total rental value of less than 1000 reais. Preferred neighborhoods include Brooklin, Butantã, Chácara Santo Antonio, Panamby, and Santo Amaro. They are looking for a property for one person, with at least 1 bedroom and 1 bathroom, and a minimum area of 25 square meters. The user works at Shopping Morumbi and is looking for a property with easy access to the shopping center and near a subway or train station, preferably on the green line. They prefer properties that allow pets, have a glass shower stall, a laundry tub, a fridge, a microwave, kitchen cabinets, are on a silent street, and are faced east.

Points of Interest: Shopping Morumbi

Figure 1: Example of a structured user profile used for personalized real estate retrieval.

3.3 LLM-Based Re-ranker

The LLM re-ranker refines the ordering retrieved from the retrieval model to better align with the user's intent. It does this by applying context-aware scoring using textual and structured inputs together with aggregate statistics computed across the candidate-set. The LLM then produces a refined top- k ranking using a point-wise strategy.

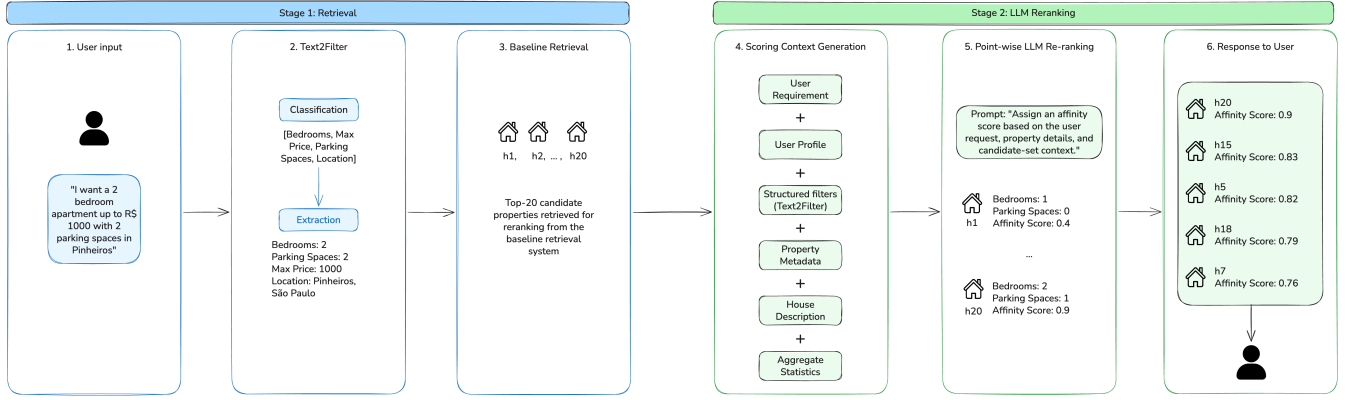


Figure 2: Overview of the LLM re-ranker pipeline.

3.3.1 Point-wise Ranking Strategy. Under this strategy, each candidate property is scored independently with an affinity score: a scalar estimate of how well the property matches the user’s intent and preferences. This independent scoring allows candidates to be evaluated in parallel, helping keep re-ranking latency low.

In contrast, list-wise methods evaluate the full candidate-set jointly and can capture richer inter-item dependencies, but they are less compatible with our production latency requirements. Pair-wise methods were also considered, but their comparison cost grows quadratically; for example, ranking 20 properties would require 190 pairwise comparisons. We therefore adopt a point-wise formulation as a practical trade-off between ranking quality, scalability, and latency.

3.3.2 Scoring Context Construction. For each request, a scoring context is constructed by combining the following inputs:

- **User requirement:** the free-text request captures the user’s primary intent, including explicit constraints, preferences, and qualitative needs that may not be represented in structured fields.
- **Textual user profile:** an LLM-generated summary of historical user interactions provides additional personalization context by capturing recurring behavioral patterns, lifestyle preferences, and implicit property interests that may not be explicitly stated in the current query.
- **Structured filters:** filters derived from Text2Filter provide normalized constraints, such as location, price range, and room requirements, which help the model distinguish hard eligibility criteria from softer preferences.
- **Property metadata:** candidate-level information, including amenities, installations, property attributes, and house descriptions, supplies the evidence used to assess whether each property satisfies the user’s stated and inferred preferences.
- **Candidate-set statistics:** aggregate summaries computed over the full candidate-set, such as price ranges, median values, room distributions, property-type distributions, and amenity prevalence, provide contextual baselines for point-wise scoring. Since each property is evaluated independently,

these statistics allow the model to judge whether a candidate is comparatively expensive, unusually well equipped, or otherwise distinctive within the retrieved set, improving score consistency across candidates.

3.3.3 Affinity Scoring and Ranking. Each candidate is scored by the LLM, which assigns an affinity score between $[0, 1]$ reflecting its alignment with the user’s intent. The scoring prompt considers three complementary dimensions: satisfaction of hard constraints, alignment with stated and inferred user preferences, and relative value within the candidate-set. This formulation combines structured constraint checking with semantic reasoning over trade-offs and implicit preferences. A simplified version of the scoring prompt is shown in Figure 3.

Formally, given a user query q and a candidate property p_i , the LLM assigns an affinity score:

$$s_i = f_{\text{LLM}}(q, p_i, u, c)$$

where u denotes the textual user profile and c represents contextual candidate-set aggregate statistics. The resulting score $s_i \in [0, 1]$ estimates how well the property aligns with the user’s intent and preferences.

The final ranked list is then obtained by sorting candidate properties in descending order of affinity score:

$$R = \text{sort}(P, s)$$

where P denotes the set of retrieved candidate properties and R denotes the final re-ranked ordering.

Affinity scores are computed asynchronously across all candidates. Properties are then sorted in descending order of affinity score; ties are broken using the original retrieval score produced by the retrieval model. Only the top- k properties are surfaced to the user, where $k = 5$ in the production system.

4 Evaluation

We evaluate the LLM-based re-ranker along two complementary axes. First, offline experiments on a labeled search dataset measure intrinsic ranking quality under controlled conditions, enabling systematic comparison of configurations and ablation studies. Second, we conducted an online A/B test in production to measure

Affinity Score Generation Prompt

You are an Affinity Score Generator in a real estate recommender system. Your task is to assign an affinity score between 0.0 and 1.0 indicating how well the property matches the user's intent.

You are provided the following:

- (i) the user requirement;
- (ii) structured user filters;
- (iii) textual user profile;
- (iv) metadata and textual description for a single candidate property; and
- (v) aggregate statistics computed across the candidate-set.

Scoring should consider:

- alignment with explicit constraints (e.g., location, price, bedrooms, etc.) and user preferences;
- semantic information from property descriptions;
- value relative to other candidate properties;
- rarity and usefulness of amenities and installations; and
- consistency of scores across independently evaluated candidates.

The full scoring range should be used, where higher scores indicate stronger alignment with the user's preferences and contextual requirements.

Figure 3: Simplified affinity score generation prompt used for point-wise LLM-based re-ranking.

the downstream impact of the re-ranker on real user interactions. In this setting, behavioral signals such as clicks and downstream engagement serve as proxies for user satisfaction. Together, offline and online evaluations provide a comprehensive view of both ranking quality and real-world effectiveness.

4.1 Offline Search Dataset

To enable robust and scalable evaluation of LLM-based re-ranking strategies, we construct a large-scale, domain-specific offline dataset. Our approach combines synthetic query generation with real-world production queries, and leverages an LLM-as-a-Judge framework for scalable relevance annotation, complemented by human validation. This design approximates realistic search behavior while avoiding the considerable cost of fully manual labeling.

The dataset construction pipeline consists of four stages (Figure 4).

4.1.1 Data Selection and Synthetic Query Generation. We randomly sampled 10,000 active property listings from QuintoAndar's inventory. Structured fields provide core attribute signals, while descriptions contribute complementary information, such as proximity to points of interest and qualitative neighborhood characteristics.

Synthetic queries are generated with *Gemini 3.1 Flash Lite* [10] by treating each listing as an anchor item. We define a taxonomy of six

query types to capture diverse search behaviors: (i) explicit attribute queries, (ii) trade-off formulations, (iii) high-complexity constraint queries, (iv) conversational queries, (v) lifestyle and amenity driven queries, and (vi) persona-based scenarios [26]. One query per category is generated for each property, yielding 60,000 synthetic queries in total. This taxonomy ensures systematic coverage of complex and underrepresented patterns, such as multi-constraint or persona-driven queries, that are sparse in production logs but critical for evaluating advanced retrieval systems.

Query quality is validated through a human evaluation on a random sample of 1,000 query-property pairs, where annotators assign binary relevance labels. We observe 94% agreement with the LLM-generated labels, which confirms high synthetic fidelity.

4.1.2 Candidate Retrieval and Hard Negative Generation. For each anchor property, we retrieve semantically similar candidates to enable the inclusion of hard negatives rather than trivial random ones. Given the cross-lingual nature of the setting (English structured metadata and Portuguese descriptions), we concatenate both field types and compute dense embeddings using *intfloat/multilingual-e5-base* [31], which offers a favorable trade-off between retrieval performance and computational efficiency. Embeddings are indexed with FAISS [6] under an inner-product similarity metric. For each anchor we retrieve the top- k nearest neighbors ($k = 10$), expanding the dataset from 60,000 to 600,000 query-item pairs while keeping LLM-based annotation tractable [33].

4.1.3 Relevance Annotation via LLM-as-a-Judge. Binary relevance labels are assigned using *Claude Sonnet 4* [1] with a structured prompting strategy: the model is instructed to (i) identify all constraints expressed in the query, (ii) verify their presence in the candidate listing, and (iii) apply strict rejection criteria for mismatches on critical attributes such as location and price range. Another human evaluation was performed on a sample of 1000 pairs yielding to 96% agreement with LLM judgments, supporting the use of LLM-as-a-Judge as a scalable annotation alternative.

4.1.4 Integration of Production Queries. To complement synthetic data with realistic user behavior, we sampled 30,000 unique queries from production logs and incorporate them into the dataset. For each query, candidate listings are retrieved using the same embedding model and FAISS configuration from stage 2. To ensure a diverse evaluation set, candidate pools are constructed by combining the top-5 most similar items (likely positives and hard negatives) with the bottom-5 least similar items (easy negatives), capturing a broad spectrum of difficulty levels. Relevance annotation is performed using the same LLM-as-a-Judge framework.

Combining synthetic and production queries provides complementary benefits: synthetic data increases coverage and complexity, while real queries reflect authentic user intent.

4.1.5 Dataset Summary. The final dataset contains 960,000 query-item pairs. Each pair consists of a user query and a candidate property listing, represented by its identifier, structured metadata, unstructured description, and a binary relevance label. The relevance distribution is 33% positive and 67% negative, reflecting a moderately imbalanced setting that increases the difficulty of creating an accurate ranking.

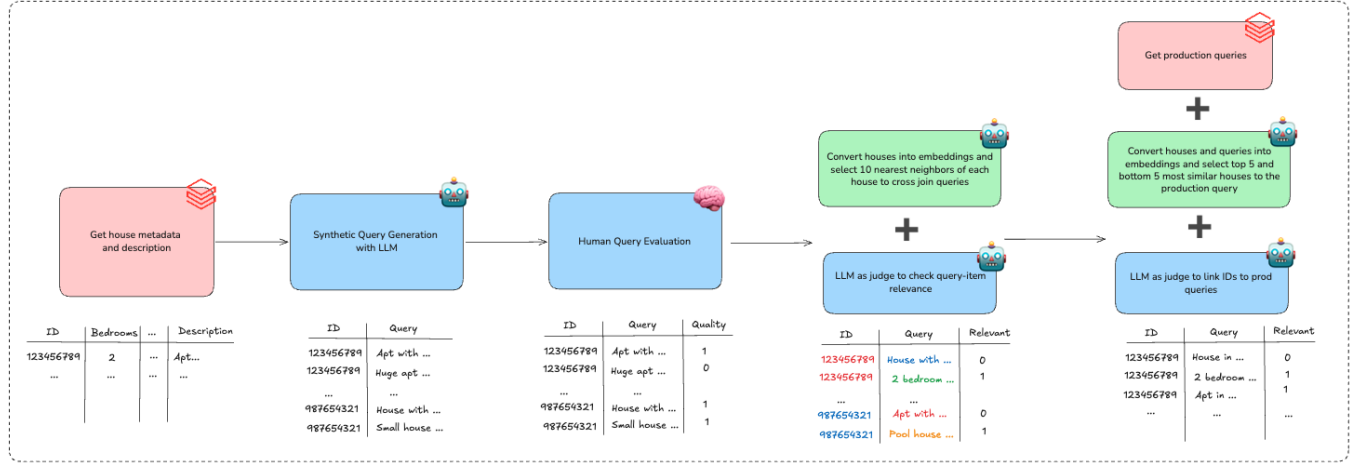


Figure 4: Overview of the dataset construction pipeline. (1) Property data is sampled from the data lake. (2) Candidate listings are retrieved using dense embeddings to construct hard negatives. (3) Relevance labels are assigned using an LLM-as-a-Judge. (4) The dataset is enriched with real-world queries.

4.2 Experimental Variations

All experimental variants are derived from the affinity-scoring prompt introduced in Section 3.3.3 and illustrated in Figure 3.

We first evaluate the baseline retrieval system without LLM-based re-ranking. We then incrementally introduce the proposed re-ranking components to measure their individual contributions to the ranking quality.

The evaluated variations include:

- **Baseline (B):** Existing retrieval system.
- **Base Re-ranker (BR):** Introduces the point-wise LLM re-ranking stage using structured property metadata and user query information.
- **Aggregated Statistics (AS):** Adds candidate-set statistics such as price ranges, room distributions, and amenity prevalence to provide contextual grounding for point-wise scoring.
- **Scoring Rules (SR):** Adds explicit scoring calibration instructions to encourage globally consistent affinity scores across independently evaluated candidates.
- **House Description (HD):** Adds unstructured property descriptions to evaluate the contribution of richer semantic property context.
- **Textual User Profile (TUP):** Adds an LLM-generated textual representation of historical user preferences to evaluate the effect of personalization signals.

4.3 Offline Evaluation and Results

All configurations are evaluated on the dataset described in Section 4.1.

4.3.1 Results. In table 1 we present the offline ranking performance across selected prompt configurations and experimental variations.

The LLM-based re-ranker substantially improved ranking quality over the baseline retrieval system across all evaluated metrics. The strongest improvements were observed in nDCG, indicating

Table 1: Offline ranking performance across prompt formulations and input configurations.

Configuration	Recall@5	nDCG@5
B + BR + AS + HD + TUP	0.814*	0.889*
B + BR + AS + HD	0.797	0.865*
B + BR + AS + SR	0.784	0.848*
B + BR + AS	0.790	0.857*
B + BR	0.784	0.850*
B	0.766	0.770

* Statistically significant compared to the baseline configuration ($p < 0.05$).

that the re-ranker is particularly effective at improving the ordering of relevant properties near the top of the ranked list. The base re-ranker improved nDCG@5 by 10.4% relative to the baseline retrieval system. Furthermore, all evaluated re-ranking configurations achieved statistically significant improvements over the baseline configuration in nDCG@5.

Adding candidate-set aggregate statistics yielded a consistent gain, with a 0.8% increase in nDCG@5 versus the matched configuration without these statistics. This suggests that candidate-level context helps the model compare relative value, rarity, and trade-offs during point-wise scoring.

In contrast, explicit scoring rules slightly degraded ranking performance, reducing nDCG@5 by 1.1% relative to the corresponding configuration without scoring calibration instructions. This suggests that rigid score calibration instructions may reduce the model’s ability to make nuanced semantic ranking decisions in the point-wise setting.

House descriptions also contributed positively to the ranking metrics, improving nDCG@5 by 0.9% relative to the corresponding configuration without descriptions. These results indicate that

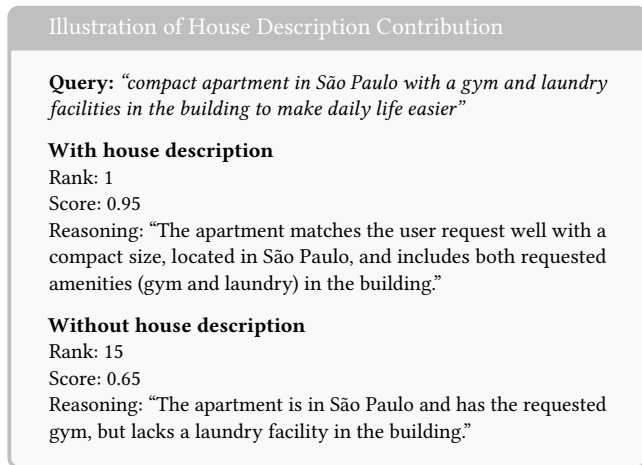


Figure 5: Example illustrating the contribution of house descriptions.

free-text property descriptions provide important semantic signals beyond structured metadata alone, particularly for lifestyle-oriented and qualitative queries. Figure 5 presents an illustrative example of this effect.

Textual user profiles produced the strongest overall performance improvements. The best-performing configuration combined aggregated statistics, house descriptions, and textual user profiles, improving nDCG@5 by 2.8% and Recall@5 by 2.1% relative to the strongest configuration without textual user profiles. These results suggest that combining richer semantic property context with historical preference representations provides stronger personalization signals for ranking semantically similar properties.

The second-best configuration, which excludes textual user profiles, corresponds to the version deployed in the online A/B experiment presented in Section 4.4. At the time of deployment, textual user profiles were available for offline evaluation but had not yet been integrated into the production serving pipeline.

Overall, the offline results show that the proposed LLM-based re-ranker improves both retrieval coverage and ranking quality relative to the baseline retrieval system. This suggests that the re-ranker is particularly effective at improving the ordering of relevant candidate properties near the top of the ranked list. These results further suggest that richer semantic context, including house descriptions, aggregated candidate-set statistics, and textual user profiles, enables the model to make more effective ranking decisions for semantically similar properties. These findings are consistent with the role of the re-ranker as a refinement stage operating over a strong retrieval baseline.

4.4 Online A/B Testing

The re-ranker was deployed in a randomized A/B experiment conducted between April 7 and May 8, 2026. The users were randomly assigned to the treatment and control groups with a 50/50 allocation. The experiment included more than 200,000 production recommendation traces collected from real user interactions with

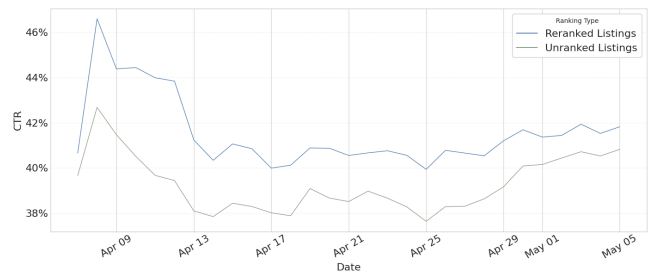


Figure 6: CTR Over Time by Ranking Type during the A/B test.

Concierge. The treatment group received LLM re-ranked results, while the control group received the baseline retrieval ordering. Key evaluation metrics included the Click-Through-Rate (CTR) and scheduled visits through the conversational system.

In the A/B experiment, the LLM-based re-ranker consistently outperformed the retrieval baseline in CTR throughout the evaluation period (Figure 6), achieving a statistically significant +5.3% relative CTR improvement. The impact also extended beyond clicks: the treatment produced a statistically significant +4.8% uplift in scheduled visits generated through the conversational system. Together, these results show that the re-ranker improved both immediate recommendation engagement and downstream real-estate outcomes in production, suggesting that better ranking quality translated into more meaningful user actions rather than only higher surface-level engagement.

From a systems perspective, the online deployment validated the practicality of the proposed point-wise architecture. Despite the additional LLM inference stage, the system maintained stable production latency throughout the experiment and remained within the operational response-time constraints of the conversational platform. Compared with the retrieval baseline, end-to-end response latency increased by an average of 4.2s, while operational inference costs increased by approximately +7%. Although these overheads represent a measurable deployment cost, they remained within acceptable production limits and were outweighed by the observed improvements in user engagement and scheduled visits.

4.5 LLM-as-a-Judge Evaluation

In addition to A/B testing, the re-ranker is evaluated on sampled production traces to assess ranking quality in a controlled setting. A LLM-as-a-Judge model compares the ordering of properties before and after re-ranking. For each trace, the user query and the top five properties from both the retrieval model and re-ranked lists are presented in anonymized form. The judge selects which ordering better satisfies the user intent.

The evaluation is formulated as a pairwise preference task rather than an absolute scoring problem. This design reduces score calibration issues and allows the judge to focus directly on comparative ranking quality. The judge is instructed to prioritize hard constraints such as location, budget, and number of bedrooms, while also considering softer preferences, semantic relevance, and

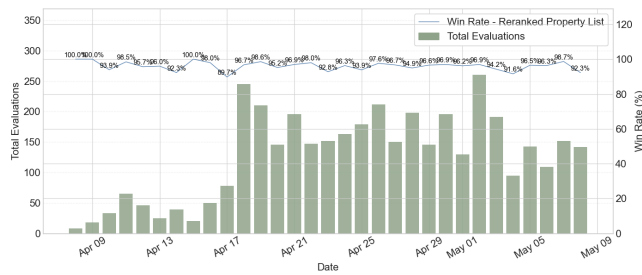


Figure 7: LLM-as-a-Judge win rate and evaluation volume

value-for-money trade-offs. To better reflect realistic recommendation behavior, properties that slightly violate constraints may still be preferred when they offer substantially better overall value.

The candidate lists are anonymized and presented without identifying information to reduce positional and presentation bias.

The LLM-as-a-Judge evaluation further supports the production results. Across 3,944 sampled traces, the judge model preferred the re-ranked property lists over the baseline retrieval ordering in 95% of evaluated cases. Figure 7 shows the daily win rate and evaluation volume throughout the evaluation period. Qualitative inspection of the judge rationales showed that the re-ranker more effectively prioritized:

- Properties satisfying explicit user constraints.
- Stronger value-for-money trade-offs.
- Semantically relevant lifestyle and amenity preferences.
- Higher-quality properties earlier in the ranked list.

The judge explanations also revealed that the re-ranker frequently improved the ordering of semantically ambiguous or borderline candidates. At the same time, it preserved the strongest matches identified by the retrieval system. These observations are consistent with the improvements in ranking quality observed in the offline metrics results.

5 Conclusion

This paper presented an LLM-based point-wise re-ranker for Concierge, a production conversational assistant for real-estate search. By combining conversational context, structured filters, property metadata, house descriptions, and aggregate statistics from the candidate-set, the proposed architecture improves the quality of the ranking while preserving the scalability benefits of parallel candidate scoring. Offline and online evaluations show gains in intrinsic ranking quality, click-through rate, and downstream engagement signals. Importantly, the production deployment maintained stable latency despite the additional LLM inference stage, suggesting that asynchronous point-wise scoring is an operationally feasible approach for LLM-based conversational recommendation systems.

6 Future Work

Future work will explore deeper agentic integration, where retrieval, ranking, and follow-up interactions are optimized jointly using the broader conversational state. This includes A/B testing the addition of the LLM-based user profile as a personalization signal, allowing

the assistant to ask clarifying questions when user intent is ambiguous, adapting the candidate-set dynamically across conversation turns, and incorporating feedback from downstream actions such as clicks, saved properties, and scheduled visits. Another promising direction is to investigate hybrid point-wise and list-wise strategies that preserve the latency advantages of independent scoring while adding richer inter-candidate reasoning when the candidate-set is small or the query requires more nuanced trade-off analysis.

7 AI Usage Disclosure

The authors used AI-assisted tools for grammar correction and minor editorial improvements. All scientific content, conclusions, and claims are solely the responsibility of the authors, who vouch for the accuracy, originality, and integrity of the work presented.

8 Authors

Nkateko is a Data Scientist at QuintoAndar, working on conversational search and recommendation systems. Previously, she worked across the fintech, consulting and insurance industries, developing machine learning solutions, recommendation models and large-scale data pipelines. Nkateko is also actively involved in technology mentorship and digital skills initiatives across Africa. She holds a BSc in Applied Statistics & Economics from the University of Cape Town.

Rafael is a Senior Data Scientist at QuintoAndar, working on conversational search and recommendation systems. Previously, he focused on recommendation challenges at a food delivery company. With more than 7 years of experience in the field, Rafael is passionate about applying machine learning to solve real-world business problems and build impactful data-driven solutions. He holds an MSc in Data Science & Engineering from the University of Porto.

Pedro Nogueira is a Head of Data Science at QuintoAndar working in search and recommendations. Previously he also tackled Information Retrieval problems at a luxury fashion ecommerce company. With more than 10 years experience under his belt, Pedro's passion is to build machine learning systems that have real impact in companies. He holds an MSc in Electrical and Computer Engineering from University of Porto.

Tiago is a Senior Staff AI/ML Researcher at Growthloop, working on Agentic Decisioning and Outcomes-based Marketing Optimizations. He has over 6 years of industry experience building data-driven decision systems in large-scale marketplaces and e-commerce environments. Previously, he worked on search and ranking problems in the travel and real estate domains. He holds a PhD in Computer Science from the University of Porto.

References

- [1] Anthropic. 2025. Introducing Claude 4. <https://www.anthropic.com/news/claude-4>
- [2] Honghui Bao, Wenjie Wang, Xinyu Lin, Fengbin Zhu, Teng Sun, Fuli Feng, and Tat-Seng Chua. 2025. Heterogeneous User Modeling for LLM-based Recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. Association for Computing Machinery, 145–154.

- [3] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. Association for Computing Machinery, 1007–1014.
- [4] Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-Tau Yih, and James Glass. 2023. Expand, Rerank, and Retrieve: Query Reranking for Open-Domain Question Answering. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 12131–12147.
- [5] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. Association for Computing Machinery, 1126–1132.
- [6] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. arXiv:2401.08281
- [7] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging large language models in conversational recommender systems. arXiv:2305.07961
- [8] Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025. LLM4Rerank: LLM-based Auto-Reranking Framework for Recommendations. In *Proceedings of the ACM on Web Conference 2025*. Association for Computing Machinery, 228–239.
- [9] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. Association for Computing Machinery, 299–315.
- [10] Google. 2026. Gemini 3.1 Flash-Lite. <https://deepmind.google/en/models/gemini/flash-lite/>. Accessed: 2026-05-15.
- [11] Kristian J. Hammond, Robin Burke, and Kathryn Schmitt. 1994. A Case-Based Approach to knowledge Navigation. In *AAAI'94*.
- [12] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large Language Models are Zero-Shot Rankers for Recommender Systems. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval*. Springer-Verlag, 364–381.
- [13] Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2025. Recommender AI Agent: Integrating Large Language Models for Interactive Recommendations. *ACM Trans. Inf. Syst.* 43 (6 2025). Issue 4.
- [14] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *ACM Comput. Surv.* 54, 5, Article 105 (May 2021), 36 pages.
- [15] Ada Jusić and Eleonora Lima. 2026. *How will AI reshape the news in 2026? Forecasts by 17 experts from around the world*. Reuters Institute for the Study of Journalism, University of Oxford. <https://reutersinstitute.politics.ox.ac.uk/news/how-will-ai-reshape-news-2026-forecasts-17-experts-around-world>
- [16] Sein Kim, Hongseok Kang, Seungyeon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large Language Models meet Collaborative Filtering: An Efficient All-round LLM-based Recommender System. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 1395–1406.
- [17] Jiahao Liu, Xueshuo Yan, Dongsheng Li, Guangping Zhang, Hansu Gu, Peng Zhang, Tun Lu, Li Shang, and Ning Gu. 2025. Improving LLM-powered Recommendations with Personalized Information. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2560–2565.
- [18] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, Prompt, and Recommendation: A Comprehensive Survey of Language Modeling Paradigm Adaptations in Recommender Systems. *Transactions of the Association for Computational Linguistics* 11 (2023), 1553–1571.
- [19] Qidong Liu, Xiangyu Zhao, Yuhao Wang, Yejing Wang, Zijian Zhang, Yuqi Sun, Xiang Li, Maolin Wang, Pengyue Jia, Chong Chen, Wei Huang, and Feng Tian. 2025. Large Language Model Enhanced Recommender Systems: Methods, Applications and Trends. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 6096–6106.
- [20] Sichun Luo, Bowei He, Haohan Zhao, Wei Shao, Yanlin Qi, Yinya Huang, Aojun Zhou, Yuxuan Yao, Zongpeng Li, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2025. RecRanker: Instruction Tuning Large Language Model as Ranker for Top-k Recommendation. *ACM Trans. Inf. Syst.* 43 (7 2025). Issue 5.
- [21] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting in Retrieval-Augmented Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 5303–5315.
- [22] Marco De Nadai, Francesco Fabbri, Paul Giglioli, Alice Wang, Ang Li, Fabrizio Silvestri, Laura Kim, Shawn Lin, Vladan Radosavljevic, Sandeep Ghael, David Nyhan, Hugues Bouchard, Mounia Lalmas, and Andreas Damianou. 2024. Personalized Audiobook Recommendations at Spotify Through Graph Neural Networks. In *Companion Proceedings of the ACM Web Conference 2024*. Association for Computing Machinery, 403–412.
- [23] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. arXiv:1904.08375
- [24] OpenAI. 2024. GPT-4o Model. <https://platform.openai.com/docs/models/gpt-4o-model-version: gpt-4o-2024-11-20>.
- [25] Enrico Palumbo, Marcus Isaksson, Alexandre Tamborrino, Maria Movin, Catalin Dinciu, Ali Vardasbi, Lev Nikeshkin, Oksana Gorobets, Anders Nyman, Poppy Newdick, Hugues Bouchard, Paul Bennett, Mounia Lalmas, Dani Doro, Christine Doig Cardet, and Ziad Sultan. 2025. You Say Search, I Say Recs: A Scalable Agentic Approach to Query Understanding and Exploratory Search at Spotify. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. Association for Computing Machinery, 1117–1121.
- [26] Enrico Palumbo, Gustavo Penha, Alva Liu, Marcus Eltscheminov, Jefferson Carvalho dos Santos, Alice Wang, Hugues Bouchard, Humberto Jesús Corona Pampin, and Michelle Tran Luu. 2025. AudioBoost: Increasing Audiobook Retrievability in Spotify Search with Synthetic Query Generation. arXiv:2509.06452
- [27] Francesco Ricci and Quang Nhat Nguyen. 2007. Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System. *IEEE Intelligent Systems* 22, 3 (May 2007), 22–29.
- [28] Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024. Large Language Models are Learnable Planners for Long-Term Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1893–1903.
- [29] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 235–244.
- [30] Jianling Wang, Ainur Yessenalina, and Roshan-Ghias Alireza. 2022. Exploring Heterogeneous Metadata for Video Recommendation with Two-tower Model. (2022). arXiv:2109.11059
- [31] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. arXiv:2212.03533
- [32] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024. RecMind: Large Language Model Powered Agent For Recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, 4351–4364.
- [33] Jiaqi Xi, Raghav Saboo, Luming Chen, Martin Wang, and Sudeep Das. 2026. Mine and Refine: Optimizing Graded Relevance in E-commerce Search Retrieval. arXiv:2602.17654
- [34] Adam Yang, Gustavo Penha, Enrico Palumbo, and Hugues Bouchard. 2025. Aligned Query Expansion: Efficient Query Expansion for Information Retrieval through LLM Alignment. (2025). arXiv:2507.11042
- [35] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024. AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems. In *Proceedings of the ACM Web Conference 2024*. Association for Computing Machinery, 3679–3689.
- [36] Joyce Zhou, Yijia Dai, and Thorsten Joachims. 2024. Language-Based User Profiles for Recommendation. arXiv:2402.15623
- [37] Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. 2021. TextGNN: Improving Text Encoder via Graph Neural Network in Sponsored Search. In *Proceedings of the Web Conference 2021*. Association for Computing Machinery, 2848–2857.