Multimodal Item Scoring for Natural Language Recommendation via Gaussian Process Regression with LLM Relevance Judgments

Yifan Simon Liu[*1], **Qianfeng Wen**[*1], **Jiazhou Liang**[*1], **Mark Zhao**[*1], **Justin Cui**[1], **Anton Korikov**[1], **Armin Toroghi**[1], **Junyoung Kim**[2] **and Scott Sanner**[1,3]
[1]University of Toronto, Canada
[2]Sungkyunkwan University, South Korea
[3]Vector Institute of Artificial Intelligence, Toronto, Canada

Abstract

Natural Language Recommendation (NLRec) generates item suggestions based on the relevance between user-issued NL requests and NL item description passages. Existing NLRec approaches often use Dense Retrieval (DR) to compute item relevance scores from aggregation of inner products between user request embeddings and relevant passage embeddings. However, DR views the request as the sole relevance signal, resulting in a unimodal scoring function centered on the request embedding, which is often a weak proxy for true relevance. To better capture the multiple relevance modes that may arise in complex NLRec data, we propose **GPR-LLM**, which uses Gaussian Process Regression (GPR) to estimate the underlying relevance function from multiple LLM-judged anchor passages instead of treating the request as the sole relevance signal. Experiments on four NLRec datasets and three LLM backbones demonstrate that GPR-LLM consistently outperforms baseline methods including DR, cross-encoder, and pointwise LLM-based relevance scoring by up to 65%.

1 Introduction

Natural Language Recommendation (NLRec) (Kang et al., 2017) aims to generate item suggestions based on user-issued free-form textual NL requests. Unlike traditional recommender systems that rely on historical interaction data, NLRec assumes the request itself encodes the user's preferences and intent (Kang et al., 2017; Bogers and Koolen, 2017). Each item is typically associated with multiple descriptive passages such as summaries, reviews, or menus. To score item relevance, existing approaches commonly use Dense Retrieval (DR; Karpukhin et al., 2020), which first estimates passage relevance scores based on inner product between NL request and passage embeddings, and then aggregates these passage-level scores into item-level scores. The standard DR approach implicitly assumes that the relevance scoring function is unimodal with a peak centered around the NL request (Figure 1, top). However, in practice, relevance in NLRec is often more complex, since a single request may involve multiple semantic aspects, each supported by different relevant passages. As a result, passages relevant to the same request may reflect different aspects of the request and therefore lie in separated regions of the embedding space (Figure 1, bottom).

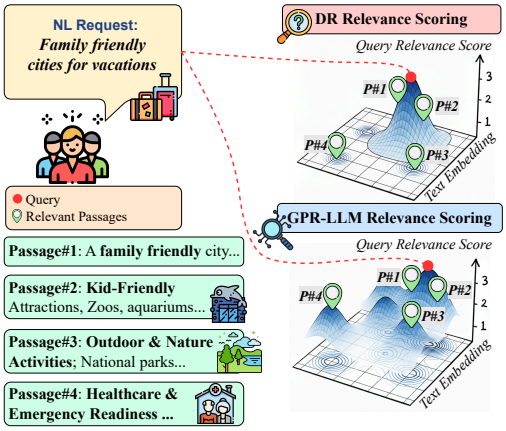

Figure 1: **(Top)** Dense retrieval (DR) methods assume a unimodal relevance scoring function concentrated near the user's NL request within the embedding space. **(Bottom)** In practice, the true relevance function is often multimodal, with relevant passages dispersed across distinct regions. The relevance scoring function for NL request "*Family friendly cities for vacations*" is multimodal as it covers multiple theme passages.

Large Language Models (LLMs) have recently emerged as a promising resource for reasoning about relevance between items or passages and NL queries, which offer more reliable relevance scoring than can be achieved with DR (Sachan et al., 2022; Qin et al., 2024; Sun et al., 2023; Ma et al., 2023; Zhuang et al., 2024; Shen et al., 2024). How-

[*]Equal contribution

Findings of the Association for Computational Linguistics: ACL 2026, pages 36859–36876

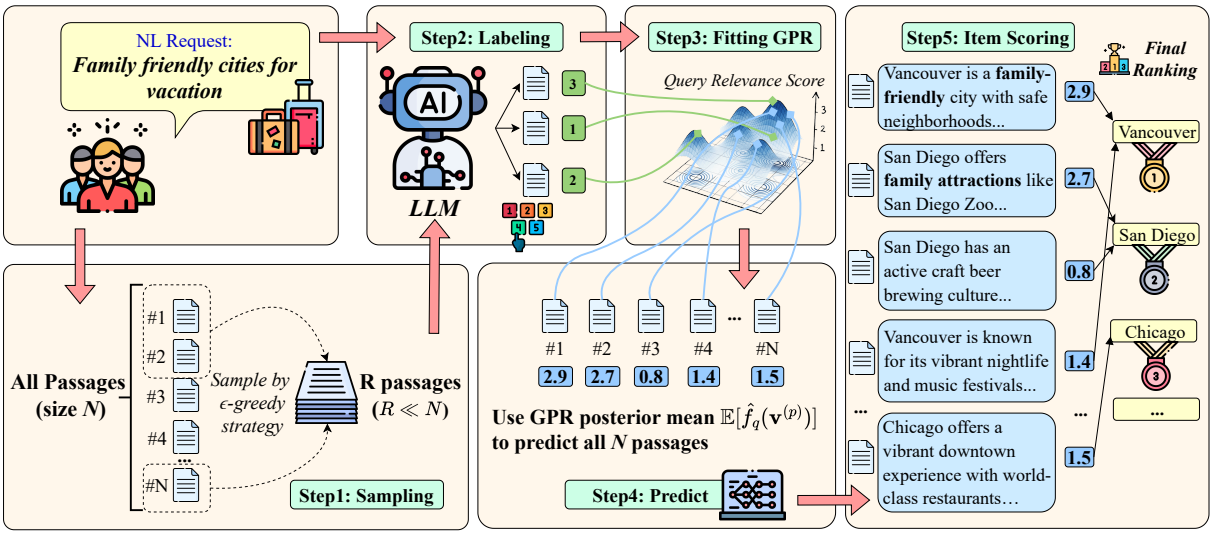

Figure 2: Overview of GPR-LLM. A small subset of $R \ll N$ passages is sampled from the full set of N passages using an ϵ-greedy sampling strategy and labeled using an LLM. A query-specific GPR model is then trained on this labeled subset to estimate relevance scores for *all* passages. Finally, item-level relevance scores are aggregated from the passage-level scores using Equation 3, and items are ranked accordingly to generate the recommendation list.

ever, exhaustively prompting LLMs for relevance judgments of all passages associated with an item is expensive, especially with large item collections and numerous NL description passages.

In order to improve the reliability of relevance scoring for NLRec while working within a minimal budget of LLM labeling calls, we propose **GPR-LLM** (cf. Figure 2), which uses Gaussian Process Regression (GPR) (Rasmussen and Williams, 2006) to estimate the underlying relevance function from LLM relevance judgments on a small subset of candidate passages.

As GPR is defined by a kernel function, GPR-LLM naturally generalizes DR. Under a linear kernel and with the NL request as the only relevance signal, it reduces exactly to standard DR. GPR-LLM becomes more flexible once additional relevance signals from LLM passage judgments are incorporated to estimate the underlying relevance function. In particular, these LLM-judged passages act as anchor relevance signals, while kernels with stronger locality, such as RBF, allow each anchor to primarily influence nearby passages, thereby capturing multiple relevance modes.

In summary, we make the following key contributions:

- We propose GPR-LLM, which uses GPR to estimate the relevance scoring function from multiple LLM-judged anchor passages, addressing the unimodal assumption of DR.
- We empirically show that GPR-LLM with an RBF kernel consistently outperforms linear kernels, as its stronger locality allows different LLM-judged anchor passages to influence different neighborhoods and better capture multiple relevance modes in complex NLRec data.
- GPR-LLM consistently outperforms all baselines, including BM25, DR, Cross-encoder, and LLM-based relevance scoring under the same LLM labeling budget, and achieves comparable performance to the best baselines even with a significantly smaller budget.

2 Preliminaries

2.1 Gaussian Process Regression

Gaussian Process Regression (GPR) is a non-parametric Bayesian regression method that models an unknown real-valued function through a Gaussian process prior (Rasmussen and Williams, 2006). Let $f : \mathbb{R}^d \to \mathbb{R}$ denote the latent function of interest. In GPR, f is assumed to follow

$$f \sim \mathcal{GP}(0, k(\cdot, \cdot)),$$

where $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a kernel function.

Given a set of inputs $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times d}$ and corresponding observed outputs $\mathbf{y} \in \mathbb{R}^n$, GPR defines the posterior predictive distribution at a new input $x_* \in \mathbb{R}^d$ as

$$f(x_*) \mid X, \mathbf{y}, x_* \sim \mathcal{N}(\mu_*, \sigma_*^2),$$

with posterior mean and variance

$$\mu_* = \mathbf{k}_*^\top (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{y},$$

$$\sigma_*^2 = k(x_*, x_*) - \mathbf{k}_*^\top (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{k}_*,$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the kernel matrix with entries $\mathbf{K}_{ij} = k(x_i, x_j)$, $\mathbf{k}_* \in \mathbb{R}^n$ is the vector of kernel values between x_* and the training inputs, i.e., $\mathbf{k}_* = [k(x_*, x_1), \ldots, k(x_*, x_n)]^\top$, and $\alpha > 0$ is the observation noise variance. The function value at x_* is estimated by the posterior mean μ_*, while σ_*^2 quantifies predictive uncertainty.

Kernels for GPR The kernel function in GPR determines how similarity between inputs is measured. In this work, we consider the following kernels:

- **Dot Product (Linear):**

$$k(x, x') = x^\top x'$$

  This kernel measures similarity by the inner product between two inputs.

- **Cosine Similarity:**

$$k(x, x') = \frac{x^\top x'}{\|x\| \cdot \|x'\|}$$

  This kernel measures similarity by the angle between two inputs.

- **Radial Basis Function (RBF):**

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right)$$

  This kernel defines similarity as an exponential decay of squared Euclidean distance between two inputs.

2.2 Natural Language Recommendation

We formulate Natural Language Recommendation (NLRec) as the task of ranking a set of items in response to a user-issued NL request. We denote such a NL request by a query $q \in \mathcal{Q}$. Let $\mathcal{I}$ represent a collection of total M items. Each item $i \in \mathcal{I}$ is associated with a set of textual passages. We denote the complete collection of N passages as $\mathcal{P}$, and their corresponding text embeddings as $\mathcal{V}$, where $\mathbf{v}^{(p_j)} \in \mathbb{R}^D$ represents the embedding of passage $p_j \in \mathcal{P}$. Each passage $p_j \in \mathcal{P}$ is uniquely linked to item i. We define the set of item i's passages as

$$\mathcal{P}^{(i)} = \{p_j \in \mathcal{P} \mid p_j \text{ is associated with item } i\}. \quad (1)$$

The objective is to recommend the item $i \in \mathcal{I}$ that is most relevant to q based on its associated passages $\mathcal{P}^{(i)}$. Thus, we decompose the task into two subproblems: passage-level relevance estimation and item-level relevance aggregation.

Passage-Level Relevance Let $f_q^* : \mathcal{P} \to \mathbb{R}$ be a query-specific passage relevance scoring function that assigns a ground-truth real-valued relevance score to passage $p_j \in \mathcal{P}$ with respect to query q.

Let s_j^* denote the ground-truth relevance score of passage $p_j \in \mathcal{P}$ with respect to query q, defined as:

$$s_j^* = f_q^*(p_j) \quad \forall p_j \in \mathcal{P}. \quad (2)$$

However, in real-world applications, $f_q^*(p_j)$ is often unknown; thus, we employ $\hat{f}_q : \mathcal{P} \to \mathbb{R}$ to estimate these scores as $s_j = \hat{f}_q(p_j)$ for all $p_j \in \mathcal{P}$. For notational convenience, we group passage scores by item, i.e., $\mathcal{S}_{i,j} = s_j \; \forall p_j \in \mathcal{P}^{(i)}, i \in \mathcal{I}$.

Item-Level Relevance Given the estimated passage-level relevance scores, item-level scores can be derived by aggregating $\mathcal{S}_{i,j}$ by item. Specifically, for each item i, we first select the top-T passages from $\mathcal{P}^{(i)}$ with highest $\mathcal{S}_{i,j}$. We denote this list as $\mathcal{P}^{(i)}_{\text{top-T}}$.

The item-level relevance score $\mathcal{S}_i$ for an item i is computed as follows:

$$\mathcal{S}_i = \phi\left(\left[\, s_j \mid p_j \in \mathcal{P}^{(i)}_{\text{top-T}} \,\right]\right) \quad \forall i \in \mathcal{I}, \quad (3)$$

where $\phi : \mathbb{R}^T \to \mathbb{R}$ is an aggregation function (e.g., mean or max). The top-K NLRec items are ranked by $\mathcal{S}_i$ in descending order.

The effectiveness of NLRec relies heavily on the quality of estimated passage-level scores. Thus, we focus on the formulation of $\hat{f}_q$, aiming to closely approximate the true relevance function f_q^* and capture multimodal relevance patterns, thereby enabling better item-level relevance estimation for NLRec.

3 GPR-LLM: Gaussian Process Regression with LLM Relevance Judgments

To estimate the underlying relevance function f_q^* in complex NLRec settings, we use a query-specific GPR scorer $\hat{f}_q : \mathcal{V} \to \mathbb{R}$ that leverages LLM relevance judgments on a small, carefully sampled subset of passages as anchor relevance signals. This

allows GPR-LLM to move beyond a single query-centered relevance signal and better capture multiple relevance modes that may arise in $\mathcal{S}^*$ (cf. Figure 1, bottom). At a high level:

1. *Candidate Sampling:* from all N passages, construct a small labeled set $\mathcal{P}^{\text{GP}}$ of size $R \ll N$ via an ϵ-greedy strategy that mixes top DR-ranked passages with uniform random exploration (cf. Section 3.3).

2. *LLM Relevance Judgments:* Obtain an LLM relevance score s_j for each sampled passage $p_j \in \mathcal{P}^{\text{GP}}$(cf. Section 3.2).

3. *Fitting GPR:* fit a query-specific GP prior/posterior over $\hat{f}_q$(cf. Section 3.1).

4. *Scoring all passages:* use the GPR posterior mean $\mathbb{E}[\hat{f}_q(\mathbf{v}^{(p)})]$ to produce passage-level scores for all $p \in \mathcal{P}$, followed by standard item-level aggregation.

3.1 Query-specific GPR.

For each NLRec query q, we instantiate a query-specific GPR scorer $\hat{f}_q : \mathcal{V} \to \mathbb{R}$ to estimate passage-level relevance in the embedding space. Specifically, $\hat{f}_q(\mathbf{v}^{(p)})$ predicts the relevance score of passage p to query q. We place a GP prior over this query-specific relevance function:

$$\hat{f}_q(\mathcal{V}) \sim \mathcal{GP}(0, k(\mathcal{V}, \mathcal{V}')). \tag{4}$$

where $k(\cdot, \cdot)$ is the kernel function defined in Section 2.1. In this work, we consider three kernel choices: dot product, cosine similarity, and RBF.

To construct supervision for q, we use both the query embedding and a small set of sampled passages with observed relevance scores. In particular, we define

$$\mathcal{D} = \{(\mathbf{v}^{(q)}, s_{\max})\} \cup \{(\mathbf{v}^{(p_j)}, s_j)\}_{j=1}^{R}, \tag{5}$$

where $\mathbf{v}^{(q)}$ is the embedding of the query, $s_{\max}$ is the maximum relevance score assigned to the query anchor, and s_j is the observed relevance score of sampled passage p_j.

For a candidate passage p_* with embedding $\mathbf{v}^{(p_*)}$, the noisy-observation posterior is

$$p\Big(\hat{f}_q(\mathbf{v}^{(p_*)}) \mid \mathcal{D}\Big) \sim \mathcal{N}(\mu_*, \sigma_*^2),$$

with

$$\mu_* = \mathbf{k}_*^\top (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{y}, \tag{6}$$

$$\sigma_*^2 = k(\mathbf{v}^{(p_*)}, \mathbf{v}^{(p_*)}) - \mathbf{k}_*^\top (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{k}_*, \tag{7}$$

where

- $\mathbf{y} = [s_{\max}, s_1, \ldots, s_R]^\top \in \mathbb{R}^{R+1}$ is the vector of observed relevance labels,
- $\mathbf{K} \in \mathbb{R}^{(R+1)\times(R+1)}$ is the kernel matrix over the query anchor and the R labeled passages,
- $\mathbf{k}_* \in \mathbb{R}^{R+1}$ is the cross-kernel vector between p_* and the labeled query-passage set, and
- $\alpha > 0$ is the observation noise variance on the relevance labels.

The posterior mean $\mu_* = \mathbb{E}[\hat{f}_q(\mathbf{v}^{(p_*)})]$ is used as the estimated relevance score of passage p_* for query q, which is later aggregated to the item level.

3.2 LLM Relevance Judgments for GPR

Another crucial consideration for GPR-LLM is the source of relevance judgments used in $\mathcal{D}$. LLMs are capable of modeling complex relationships between queries and passages that require contextual reasoning, inference, or multi-step understanding (Sun et al., 2023; Qin et al., 2024; Jiao et al., 2026; Wen et al., 2025b; Liu et al., 2025a). This makes LLM relevance judgments a natural choice to estimate S^*.

We use the commonly adopted UMBRELA `prompt` (Upadhyay et al., 2024) and follow the procedure of Zhuang et al. (2024) to obtain the LLM relevance judgment between a query q and a passage p_j as follows:

$$\mathbf{z} = \text{LLM}(q, p_j, \texttt{prompt}) \tag{8}$$

where $\mathbf{z} = [z_0, z_1, \ldots, z_{K-1}]$ corresponds to the logit of a predefined discrete relevance label $r_k \in \{0, 1, \ldots, K-1\}$. We then compute the LLM-based relevance score using the *expected relevance*:

$$s_j = \sum_{k=0}^{K-1} \left(\frac{e^{z_k}}{\sum_{j=0}^{K-1} e^{z_j}} \right) \cdot r_k. \tag{9}$$

3.3 Sampling for GPR

Since obtaining LLM relevance judgments for all N passages is expensive, we instead construct a sampled subset of passages of size $R \ll N$, denoted as $\mathcal{P}^{\text{GP}}$. Ideally, $\mathcal{P}^{\text{GP}}$ should cover all relevant passages. While dense retrieval (DR) efficiently identifies potentially relevant passages, its unimodal bias in embedding space may overlook relevant passages that are distant from the query representation.

To mitigate this limitation, we propose an ϵ-*greedy sampling* strategy that balances exploitation of DR rankings with controlled exploration. Let $\{p_{(1)}, \ldots, p_{(N)}\}$ denote passages sorted by DR score in descending order. We construct $\mathcal{P}^{\mathrm{GP}}$ as:

$$\mathcal{P}^{\mathrm{GP}} = \mathcal{G}_q \cup \mathcal{U}_q,$$

where:

$$\mathcal{G}_q = \{p_{(1)}, \ldots, p_{(\lfloor (1-\epsilon)R \rfloor)}\}$$

is the greedy set consisting of the top-ranked passages, and

$$\mathcal{E}_q^{\tau} = \{p_{(1)}, \ldots, p_{(\tau)}\} \setminus \mathcal{G}_q$$

is the exploration pool formed by removing the greedy set from the top-τ DR-ranked passages. We then sample:

$$\mathcal{U}_q \sim \mathrm{UnifSample}(\mathcal{E}_q^{\tau}, \lceil \epsilon R \rceil),$$

i.e., $\lceil \epsilon R \rceil$ passages are drawn uniformly at random from $\mathcal{E}_q^{\tau}$ without replacement.

This sampling scheme thus introduces two parameters:

- $\epsilon \in [0, 1]$ controls the exploration–exploitation trade-off. When $\epsilon = 0$, the method reduces to purely greedy selection based on DR. When $\epsilon = 1$, all sampled passages are drawn uniformly from the top-τ DR-ranked pool.
- τ defines the size of the candidate pool for exploration, restricting sampling to high-ranking passages and avoiding low-relevance regions of the ranking.

Let $\mathcal{D}_q = \{(\mathbf{v}^{(p_j)}, s_j)\}_{p_j \in \mathcal{P}^{\mathrm{GP}}}$ denote the labeled dataset, where s_j is the LLM-derived relevance score for passage p_j, and $\mathbf{v}^{(p_j)}$ is its dense embedding. We then estimate relevance scores for all passages in $\mathcal{P}$ using Gaussian Process regression (cf. Equation 6), and aggregate them for item-level recommendation (cf. Equation 3). Figure 2 illustrates the overall GPR-LLM pipeline.

3.4 Complexity Analysis

Per query, GPR-LLM proceeds in four stages:

1. *Candidate sampling.* We construct the candidate pool via ϵ-greedy sampling that selects a fraction of passages from a DR ranking. The DR pass over N passages with D-dimensional embeddings costs $\mathcal{O}(ND)$, with negligible additional cost for the ϵ-greedy draw.

Method	Per-query Complexity	Latency (sec)
DR	$\mathcal{O}(ND)$	0.165 [0.161, 0.168]
	LLM-based Scoring	
PW	$\mathcal{O}(ND + R\,C_{\mathrm{LLM}})$	0.678 [0.671, 0.685]
	GPR–LLM	
Dot	$\mathcal{O}(ND + NRD + R\,C_{\mathrm{LLM}} + R^2D + R^3)$	0.782 [0.769, 0.795]
Cosine		0.774 [0.762, 0.787]
RBF		0.754 [0.730, 0.780]

Table 1: Per-query time complexity and latency (seconds) under R=50, N=100,000, and D=384. Latencies show 95% CIs in $[\cdot]$. All computations were performed on an NVIDIA GeForce RTX 4070 GPU.

2. *LLM relevance judgments;* We obtain relevance judgments for R passages at cost $\mathcal{O}(R\,C_{\mathrm{LLM}})$, where C_{LLM} is the per-call cost.
3. *GPR fitting.* Building the dense kernel on the R labeled embeddings costs $\mathcal{O}(R^2D)$; the exact GP Cholesky solve costs $\mathcal{O}(R^3)$.
4. *Scoring all passages.* Scoring all N passages requires forming the cross-kernel $K_{N\times R}$ in $\mathcal{O}(NRD)$ and computing the posterior mean in $\mathcal{O}(NR)$; overall this stage is $\mathcal{O}(NRD)$.

Putting it together, the per-query runtime is

$$\mathcal{O}(ND + R\,C_{\mathrm{LLM}} + R^2D + R^3 + NRD).$$

With $R \ll N$, the dominant terms are typically the DR pass and the GP inference over all passages, i.e., $\mathcal{O}(ND + RC_{\mathrm{LLM}})$, while GPR fitting is inexpensive. In practice, the additional overhead of GPR–LLM compared to pointwise LLM-based scoring with a budget of R passages is minor. See Table 1 for details.

4 Experiments

4.1 Experimental Setup

We compare GPR-LLM against the following baseline relevance scoring methods[1][2]:

- BM25 (Robertson et al., 1994): Computes relevance scores using BM25.
- Dense Retrieval (DR) (Karpukhin et al., 2020): Computes relevance scores using inner products between query and passage embeddings.
- Cross-Encoder (CE) (Nogueira et al., 2019): Computes relevance scores by jointly encoding query-passage pairs.

[1]Code for reproducing experiments is available at https://github.com/QianfengWen/Bandit_Retrieval.git.

[2]See Appendix B for detailed baseline implementations.

Dataset	# Queries	# Items	# Passages	# Qrel
TravelDest	50	774	126,400	3,721
POINTREC	28	59,553	592,818	498
TripAdvisor Hotel	100	589	133,759	4,887
Yelp Restaurant	100	1,152	283,658	11,726

Table 2: Statistics of our benchmark datasets. Qrels indicates the total number of relevant items summed over all queries.

- Pointwise LLM-based Relevance Scoring (PW) (Zhuang et al., 2024): Computes relevance scores by prompting an LLM with query–passage pairs and computing the expected relevance score from the model's predicted label distribution.

We conduct experiments on four publicly available benchmark datasets:

- `POINTREC` (Afzali et al., 2021): Point-of-interest recommendation using reviews and structured descriptions.
- `TravelDest` (Wen et al., 2024): Travel city recommendation using city-level descriptions.
- `TripAdvisor Hotel` (Wen et al., 2025a): Hotel recommendation using user reviews.
- `Yelp Restaurant` (Wen et al., 2025a): Restaurant recommendation using user reviews.

These datasets allow us to assess the generalizability of our approach across diverse NLRec scenarios. We selected them because their queries are typically complex, broad, and multi-aspect natural language requests, while their items are represented by diverse collections of textual passages, such as descriptions and reviews. This makes them suitable for evaluating whether a method can estimate multiple relevance modes rather than relying on a single dominant similarity pattern. See Table 2 for dataset statistics.

We experiment with three LLM backbones:

- GPT-4o (OpenAI, 2024)
- Qwen3-Next-80B-A3B as an open-source model (Yang et al., 2025)
- Qwen3-8B as a smaller open-source model (Yang et al., 2025)

We use the UMBRELA prompt (Upadhyay et al., 2024) to obtain LLM relevance judgments across various LLM budgets R. For embeddings, we use the `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019) and the `msmarco-distilbert-base-tas-b` (Hofstätter et al., 2021; see Appendix E). We set the number of top passages to aggregate as $T = 3$ and define the aggregation function ϕ (cf. Equation 3) as the mean $\mathrm{mean}(\cdot)$ over passage scores. The observation noise variance α is set by default to 10^{-3}. Finally, the length scale ℓ in the RBF kernel is optimized using the L-BFGS-B (Zhu et al., 1997) algorithm.[3]

Evaluation is conducted using NDCG and Precision@10 and @30, where NDCG measures the quality of the ranked list by accounting for both relevance levels and item positions, while Precision measure the accuracy of the top-ranked recommendations at practically relevant cutoffs.

We address the following research questions:

RQ1 (Kernel Choice): Does the RBF kernel, whose stronger locality allows it to estimate multiple relevance modes, outperform linear kernels in passage-level relevance scoring for complex NLRec data?

RQ2 (Sampling Strategy): Does including a fraction ϵ of randomly sampled exploratory passages with a cap τ lead to better relevance scoring compared to only selecting top-ranked passages from DR?

RQ3 (Performance Comparison): Does GPR-LLM consistently outperform baseline methods across different LLM backbones given the same labeling budget?

RQ4 (Multimodal Relevance Scoring): Do empirical results support that GPR-LLM with the RBF kernel more effectively captures multimodal relevance compared to other methods?

4.2 Results

RQ1 (Kernel Choice) To address **RQ1**, Figure 3 compares different kernel functions under greedy sampling ($\epsilon = 0$) with τ set to include all passages. The RBF kernel consistently and significantly outperforms the linear dot product and cosine similarity kernels across all datasets and LLM backbones.

[3] Additional experiments analyzing the impact of different embeddings and hyperparameter choices are provided in Appendix E, Appendix F, and Appendix G

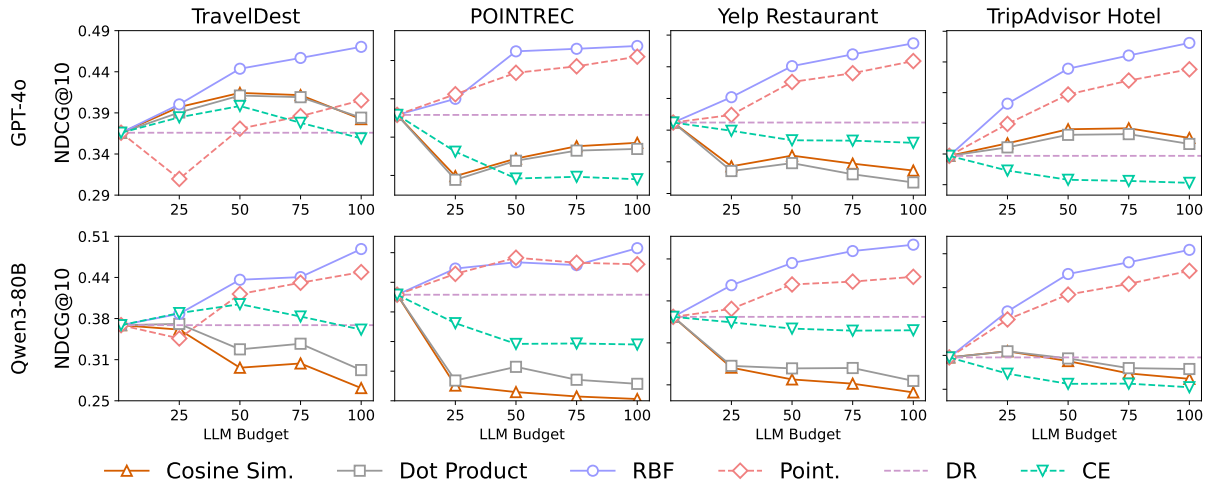

Figure 3: **RQ1:** Kernel choices comparison with Greedy Sampling ($\epsilon = 0$) under varying LLM labeling budget.

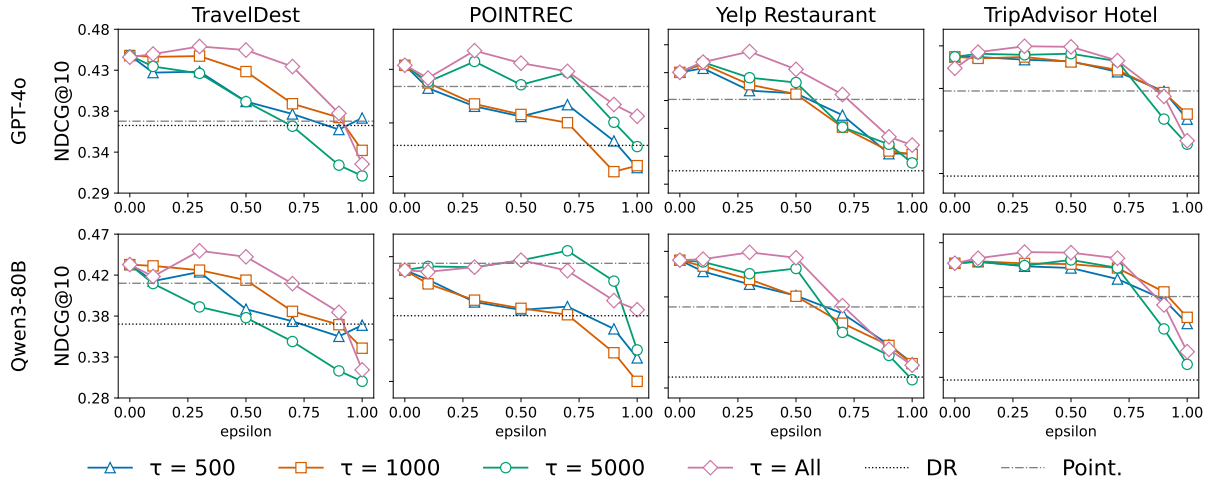

Figure 4: **RQ2:** Performance of GPR-LLM under $R = 50$ with varying sampling sets. Each configuration labels an ϵ fraction of random passages from the remaining top-τ DR-ranked passages and a $(1 - \epsilon)$ fraction from top DR-ranked passages.

These results suggest that kernels with stronger locality are better suited to GPR-LLM, since they allow different labeled anchors to primarily influence different neighborhoods and thereby better capture multiple relevance modes in the passage-level relevance function. This is further examined in **RQ4**. Additionally, Figure 3 demonstrates that GPR-LLM with the RBF kernel outperforms all baseline methods under the same labeling budget, even without tuning sampling parameters.

RQ2 (Sampling Strategy). We next investigate whether tuning the sampling strategy beyond purely greedy selection can further improve GPR-LLM. Figure 4 shows the effect of different sampling configurations that mix a fraction ϵ of randomly selected passages from the remaining top-τ DR-ranked passages with $(1 - \epsilon)$ passages drawn from the top DR rankings. Across all datasets and LLM backbones, a small exploratory fraction ($\epsilon = 0.3$) consistently improves GPR-LLM performance when the sampling range is sufficiently large ($\tau \geq 5000$), whereas no such gain is observed for smaller τ. Under most sampling configurations, GPR-LLM also outperforms the pointwise LLM-based scoring baseline (dot–dash line). These results suggest that exploration enables GPR-LLM to better capture the multimodal relevance distribution by incorporating a more diverse subset of anchor passages, but only when the exploration remains balanced and the sampled passages are sufficiently distinct from the top DR-ranking.

Budget	Backbone	Method	TravelDest				POINTREC				Yelp Restaurant				TripAdvisor Hotel			
			P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30
Budget = N/A																		
N/A	N/A	BM25	0.234	0.238	0.237	0.239	0.025	0.032	0.025	0.038	0.309	0.327	0.236	0.283	0.205	0.257	0.153	0.325
	N/A	DR	0.360	0.366	0.314	0.332	0.164	0.179	0.104	0.182	0.346	0.362	0.282	0.331	0.231	0.297	0.166	0.365
Budget = 25																		
25	N/A	CE	0.384	0.385	0.316	0.336	0.121	0.131	0.087	0.137	0.332	0.349	0.275	0.321	0.227	0.273	0.168	0.336
	GPT-4o	PW	0.294	0.309	0.238	0.219	**0.175**	**0.206**	**0.106**	**0.158**	0.324	0.374	0.237	0.260	0.251	0.349	0.178	0.332
		GPR-LLM	**0.376***	**0.401***	**0.340***	**0.364***	0.157	0.200	**0.106**	**0.158**	**0.360***	**0.402***	**0.273***	**0.340***	**0.282***	**0.382***	**0.182**	**0.426***
	Qwen3-80B	PW	0.308	0.345	**0.286**	0.309	0.175	0.214	**0.111**	**0.206**	0.336	0.381	0.266	0.331	0.258	0.353	0.165	0.398
		GPR-LLM	**0.366***	**0.384***	0.285	**0.316**	**0.179**	**0.223**	0.082	0.177	**0.395***	**0.437***	**0.277**	**0.353***	**0.270**	**0.365**	**0.176**	**0.418**
	Qwen3-8B	PW	0.282	0.313	**0.288**	0.303	0.150	0.174	**0.106**	**0.185**	0.322	0.352	0.261	0.314	0.234	0.316	**0.164**	0.377
		GPR-LLM	**0.328***	**0.357***	0.273	**0.304**	**0.154**	**0.176**	0.096	0.182	**0.355***	**0.394***	**0.274**	**0.336***	**0.252**	**0.338**	0.163	**0.384**
Budget = 50																		
50	N/A	CE	0.390	0.399	0.325	0.348	0.093	0.096	0.077	0.110	0.314	0.334	0.272	0.315	0.216	0.258	0.167	0.329
	GPT-4o	PW	0.356	0.371	0.248	0.281	0.196	0.234	0.105	0.208	0.386	0.426	0.254	0.332	0.289	0.397	0.169	0.417
		GPR-LLM	**0.432***	**0.445***	**0.362***	**0.389***	**0.236***	**0.262***	**0.113**	**0.222**	**0.408**	**0.451***	**0.304***	**0.377***	**0.324***	**0.439***	**0.197***	**0.482***
	Qwen3-80B	PW	0.392	0.415	0.273	0.315	**0.211**	**0.242**	**0.108**	**0.213**	0.401	0.439	0.266	0.347	0.296	0.390	0.169	0.417
		GPR-LLM	**0.414***	**0.437***	**0.327***	**0.363***	0.200	0.234	0.102	0.202	**0.439***	**0.490***	**0.308***	**0.398***	**0.314**	**0.420***	**0.196***	**0.472***
	Qwen3-8B	PW	0.348	0.382	0.261	0.302	0.182	0.205	**0.101**	0.189	0.343	0.371	0.254	0.316	0.252	0.333	0.164	0.384
		GPR-LLM	**0.370***	**0.408***	**0.315***	**0.350***	**0.188**	**0.211**	0.099	**0.192**	**0.401***	**0.429***	**0.300***	**0.367***	**0.278***	**0.367***	**0.184**	**0.426**
Budget = 100																		
100	N/A	CE	0.364	0.359	0.312	0.324	0.089	0.095	0.069	0.104	0.320	0.330	0.270	0.309	0.204	0.253	0.161	0.322
	GPT-4o	PW	0.398	0.406	0.296	0.328	0.225	0.255	0.124	0.231	0.418	0.459	0.298	0.379	0.322	0.438	0.189	0.472
		GPR-LLM	**0.448***	**0.472***	**0.380***	**0.412***	**0.246***	**0.269**	**0.130**	**0.239**	**0.444***	**0.487***	**0.334***	**0.413***	**0.358***	**0.481***	**0.218***	**0.529***
	Qwen3-80B	PW	0.418	0.449	0.323	0.366	0.207	0.230	0.121	0.218	0.424	0.457	0.303	0.381	0.319	0.425	0.192	0.467
		GPR-LLM	**0.472***	**0.486***	**0.362***	**0.399***	**0.225***	**0.258***	**0.123**	**0.226**	**0.491***	**0.534***	**0.355***	**0.443***	**0.345***	**0.456***	**0.220***	**0.520***
	Qwen3-8B	PW	0.378	0.403	0.299	0.333	0.182	**0.198**	0.106	0.190	0.356	0.384	0.274	0.334	0.260	0.344	0.172	0.403
		GPR-LLM	**0.414***	**0.442***	**0.344***	**0.376***	**0.191**	0.196	**0.111**	**0.192**	**0.401***	**0.438***	**0.314***	**0.383***	**0.295***	**0.388***	**0.193**	**0.450***

Table 3: **RQ3**: Comparison of DR, Cross-Encoder (CE), Pointwise LLM-based Relevance Scoring (PW) and GPR-LLM (RBF kernel, $\epsilon = 0.3$, τ set to include all passages) across four datasets and three LLM backbones at varying LLM label budgets. Metrics: Precision@10 (P@10), NDCG@10 (N@10), Precision@30 (P@30), NDCG@30 (N@30). Bold values indicate the best-performing method between PW and GPR-LLM for each backbone. Statistically significant improvements over PW (paired t-test, $p < 0.05$) are indicated by an asterisk (*).

RQ3 (Performance Comparison) Table 3 compares GPR-LLM (with the best-performing RBF kernel, $\epsilon = 0.3$, and τ including all passages) against all baseline methods across various LLM backbones and labeling budgets. GPR-LLM consistently outperforms baselines at same LLM budgets with only a few exceptions, and specifically achieves up to 65% improvements over the pointwise LLM relevance scoring. Also, GPR-LLM is often able to outperform baselines that use twice as many labels. The performance is consistent across different NLRec datasets and LLM backbones, which highlights the robustness and effectiveness of our proposed method.

RQ4 (Multimodal Relevance Scoring) Next, we investigate the underlying factors contributing to these performance gains. Figure 5 visualizes the distribution of top-ranked passages for a representative query from each dataset using both DR and GPR-LLM in the reduced embedding space using t-SNE (van der Maaten and Hinton, 2008).

DR's top-ranked passages tend to cluster tightly around the query across all datasets, reflecting its unimodal relevance assumption with a single peak at the query (i.e., passages closer to the query are inherently more relevant). In contrast, GPR-LLM retrieves passages distributed across multiple distinct regions in the embedding space. This distribution aligns naturally with our multimodal relevance hypothesis illustrated in Figure 1, where each region represents a local relevance peak within the overall multimodal relevance surface. By capturing distant yet relevant passages, GPR-LLM overcomes a key limitation of DR. This multimodal retrieval pattern is consistently observed across all four datasets.

5 Related Work

5.1 Natural Language Recommendation

Natural Language Recommendation (NLRec) aims to generate item recommendations based on user-issued textual requests (Kang et al., 2017). NLRec systems typically exhibit two key properties. First, they utilize natural language requests to encode user intent and preferences, contrasting with traditional recommender systems that mainly rely on structured interaction history (e.g., clicks, ratings) as their primary source of preference signals (Bogers and Koolen, 2017; Bogers et al., 2018, 2019; Afzali et al., 2021; Liu et al., 2026). Second, NLRec leverages collections of textual sources,

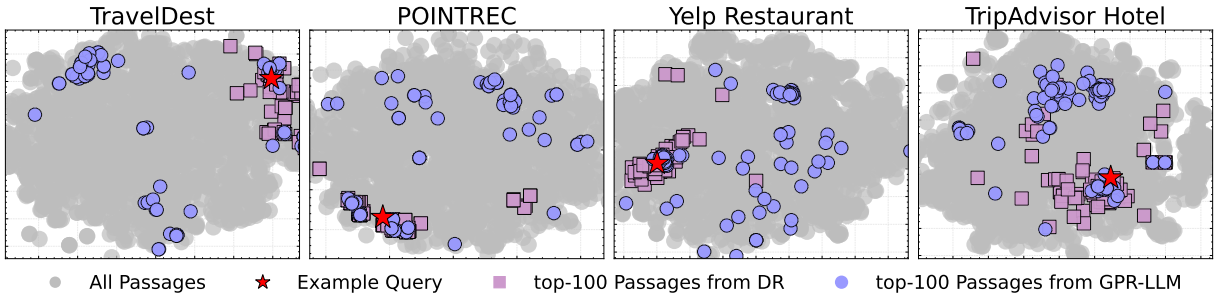

Figure 5: **RQ4:** Distribution of the top-100 passages from DR and GPR-LLM for example queries across all datasets. The embedding space is reduced to two dimensions using t-SNE for visualization. Passages ranked highest by DR (purple squares) cluster tightly around the query (red star), reflecting DR's unimodal assumption. In contrast, GPR-LLM (blue circles) identifies passages dispersed across multiple regions, highlighting its ability to capture the multimodal relevance distribution hypothesized in Figure 1.

such as item descriptions, reviews, or menus, to represent items and aggregate them into item-level representations (Afzali et al., 2021; Zhang et al., 2023; Wen et al., 2025a; Liang et al., 2026).

As NLRec tasks grow in complexity, they often exhibit *multimodal relevance*, characterized by multiple regions of high relevance scores in the dense embedding space (cf. Figure 1; Liu et al., 2025b; Wen et al., 2025a). To effectively address this multimodal relevance, we propose GPR-LLM, which aims to estimate the multimodal relevance scoring function using Gaussian Process Regression from multiple LLM-judged anchor passages.

5.2 LLM Relevance Judgment

Recently, Large Language Models (LLMs) have emerged as promising resources for determining the relevance between items or passages and natural language queries due to their strong contextual understanding and reasoning capabilities (Sachan et al., 2022; Qin et al., 2024; Sun et al., 2023; Ma et al., 2023; Zhuang et al., 2024; Shen et al., 2024).

LLM-based relevance judgments generally fall into three categories: pointwise, listwise, and pairwise. Pointwise methods independently compute relevance scores for each query–passage pair (Sachan et al., 2022; Zhuang et al., 2024; Upadhyay et al., 2024). Listwise methods prompt LLMs with a query and multiple candidate passages simultaneously, often using a sliding window approach to facilitate passage comparisons and directly generate a ranked list (Sun et al., 2023; Ma et al., 2023; Shen et al., 2024). Pairwise methods prompt LLMs to compare two passages, determine their relative relevance, and aggregate these pairwise preferences into final rankings (Qin et al., 2024).

To obtain passage-level relevance scores for aggregation into item-level scores in NLRec tasks, we primarily focus on pointwise LLM-based relevance scoring, which we compare against GPR-LLM.

6 Conclusion

We introduced GPR-LLM, a method for NLRec that uses Gaussian Process Regression with a small set of LLM-judged anchor passages to estimate the underlying relevance function. GPR-LLM moves beyond DR's query-as-sole-signal assumption by combining multiple relevance signals through kernel-based GPR estimation, allowing it to better capture the multiple relevance modes that may arise in complex NLRec data while reducing reliance on exhaustive LLM labeling. Experiments across multiple datasets and LLM backbones show that GPR-LLM consistently outperforms strong baselines, including DR, Cross-Encoder, and pointwise LLM relevance scoring under the same labeling budgets, and often achieves comparable performance with substantially fewer LLM labels. These results establish GPR-LLM as an efficient and effective approach for passage-level relevance estimation and item ranking in NLRec.

Limitations

While GPR-LLM achieves consistent improvement over baselines, several limitations remain. First, the performance of GPR-LLM critically depends on the quality and consistency of the LLM relevance judgments. Variations in LLM prompting or labeling criteria can impact accuracy and reliability, potentially influencing the quality of the GPR.

Second, we use an ϵ-greedy sampling strategy for exploration that depends on uniform random sampling. However, alternative sampling methods

that explicitly leverage uncertainty-aware active learning or diversity-maximization exploration may align better with the exploration goal. Although such methods could further enhance performance, investigating them is beyond the scope of this paper and is thus left for future study.

Third, we aggregate item-level scores using a fixed aggregation function ϕ (mean or max) over the top-T passages. Alternative aggregation methods, such as harmonic mean or learned fusion networks, could be explored to potentially enhance performance. However, as our primary focus in this paper is on improving passage-level relevance scoring, we leave the investigation of more advanced aggregation methods for future work.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2024-00457882, National AI Research Lab Project).

Ethical Considerations

In deploying GPR-LLM, it is important to consider potential ethical implications. The reliance on LLMs introduces risks related to bias and fairness, as LLM relevance judgments may inherit or amplify biases present in training data. Thus, careful evaluation and monitoring of the generated labels and recommendation results are necessary to mitigate potential discriminatory impacts.

References

Jafar Afzali, Aleksander Mark Drzewiecki, and Krisztian Balog. 2021. POINTREC: A test collection for narrative-driven point-of-interest recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2478–2484.

Toine Bogers, Maria Gäde, Marijn Koolen, Vivien Petras, and Mette Skov. 2018. What was this movie about this chick?: A comparative study of relevance aspects in book and movie discovery. In *Transforming Digital Worlds*, volume 10766 of *Lecture Notes in Computer Science*, pages 323–334, Cham. Springer.

Toine Bogers, Maria Gäde, Marijn Koolen, Vivien Petras, and Mette Skov. 2019. Looking for an amazing game i can relax and sink hours into...: A study of relevance aspects in video game discovery. In *Information in Contemporary Society*, volume 11420 of *Lecture Notes in Computer Science*, pages 503–515, Cham. Springer.

Toine Bogers and Marijn Koolen. 2017. Defining and supporting narrative-driven recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 238–242, Como, Italy. Association for Computing Machinery.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122. Association for Computing Machinery.

Difan Jiao, Qianfeng Wen, Blair Yang, Zhenwei Tang, and Ashton Anderson. 2026. Thinktwice: Jointly optimizing large language models for reasoning and self-refinement. *arXiv preprint arXiv:2604.01591*.

Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren G. Terveen, and F. Maxwell Harper. 2017. Understanding how people use natural language to ask for recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 229–237, Como, Italy. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jiazhou Liang, Yifan Simon Liu, David Guo, Minqi Sun, Yilun Jiang, and Scott Sanner. 2026. Evaluating scene-based in-situ item labeling for immersive conversational recommendation. *Preprint*, arXiv:2604.09698.

Yifan Liu, Gelila Tilahun, Xinxiang Gao, Qianfeng Wen, and Michael Gervers. 2025a. A comparative study of static and contextual embeddings for analyzing semantic changes in medieval Latin charters. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 182–192, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yifan Liu, Qianfeng Wen, Mark Zhao, Jiazhou Liang, and Scott Sanner. 2025b. MA-DPR: Manifold-aware distance metrics for dense passage retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31085–31103, Suzhou, China. Association for Computational Linguistics.

Yifan Simon Liu, Ruifan Wu, Liam Gallagher, Jiazhou Liang, Armin Toroghi, and Scott Sanner. 2026. Semantic xpath: Structured agentic memory access for conversational ai. *arXiv preprint arXiv:2603.01160*.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document

reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424*.

OpenAI. 2024. GPT-4o system card. Technical report, OpenAI.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Carl E. Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference (TREC-3)*, pages 109–126. National Institute of Standards and Technology.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15933–15946, Bangkok, Thailand. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. *arXiv preprint arXiv:2406.06519*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Qianfeng Wen, Yifan Liu, Justin Cui, Joshua Zhang, Anton Korikov, George-Kirollos Saad, and Scott Sanner. 2025a. A simple but effective elaborative query reformulation approach for natural language recommendation. *arXiv preprint arXiv:2510.02656*.

Qianfeng Wen, Yifan Liu, Joshua Zhang, George Saad, Anton Korikov, Yury Sambale, and Scott Sanner. 2024. Elaborative subtopic query reformulation for broad and indirect queries in travel destination recommendation. In *Proceedings of the 1st Workshop on Risks, Opportunities, and Evaluation of Generative Models in Recommender Systems (ROEGEN@RecSys 2024)*. Also available as arXiv:2410.01598.

Qianfeng Wen, Zhenwei Tang, and Ashton Anderson. 2025b. Chessqa: Evaluating large language models for chess understanding. *arXiv preprint arXiv:2510.23948*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Haochen Zhang, Anton Korikov, Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Manasa Bharadwaj, Ali Pesaranghader, Xi Yu Huang, Yi Xin Lok, Zhaoqi Wang, Nathan Jones, and 1 others. 2023. Recipe-MPR: A test collection for evaluating multi-aspect preference-based natural language retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2744–2753.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-bfgs-b, fortran routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024. Beyond yes and no: Improving zero-shot LLM rankers

via scoring fine-grained relevance labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 358–370, Mexico City, Mexico. Association for Computational Linguistics.

A UMBRELA Prompt for LLM Relevance Judgment

We show the `UMBRELA` prompt used to judge the relevance between a query and a single passage (pointwise setting):

UMBRELA Prompt (Pointwise Final Score)

Task: Given a query and a passage, assign a single integer relevance score from 0 to 3.
Relevance scale: 0 = The passage is unrelated to the query. 1 = The passage is somewhat related but does not answer the query. 2 = The passage contains relevant information, but the answer is incomplete, unclear, or mixed with extraneous content. 3 = The passage directly and fully answers the query.
Instructions:

- Assign 1 if the passage is topically related but does not answer the query.
- Assign 2 if the passage provides useful but partial or unclear information.
- Assign 3 if the passage fully and directly answers the query.
- Assign 0 if none of the above applies.

Input: Query: `{query}` Passage: `{passage}`
Evaluation steps:

1. Identify the underlying intent of the query.
2. Measure how well the passage matches the query intent (M).
3. Assess the trustworthiness of the passage (T).
4. Decide on the final score (O).

Output format: Return only a single integer in {0, 1, 2, 3}. Do not include any explanation or additional text.

`##final score:`

B Implementation Details

We implement the following baseline methods for relevance scoring in Natural Language Recommendation (NLRec).

BM25. We employ the Okapi BM25 algorithm (Robertson et al., 1994) as implemented in the `Pyserini` toolkit, using default hyperparameters ($k_1 = 0.9$, $b = 0.4$).

Dense Retrieval (DR). We use the `all-MiniLM-L6-v2` embedding model (Reimers and Gurevych, 2019) with $D = 384$ and the `msmarco-distilbert-base-tas-b` model (Hofstätter et al., 2021) with $D = 768$ from Hugging Face. Candidate relevance scores are computed as the inner product between query and passage embeddings.

Cross-Encoder (CE). We use the `cross-encoder/ms-marco-MiniLM-L-6-v2` model (Reimers and Gurevych, 2019; Wolf et al., 2020) from Hugging Face, which jointly encodes query–passage pairs and outputs a fine-grained scalar relevance score. The CE is applied to the top-R passages retrieved by DR to maintain comparability with GPR-LLM and pointwise LLM-based scoring.

Pointwise LLM-based Relevance Scoring (PW). We use the commonly adopted UMBRELA `prompt` (Upadhyay et al., 2024) and follow the procedure of Zhuang et al. (2024) to obtain LLM-based passage relevance scores. For each query q and passage p_j, the LLM outputs a vector of logits:

$$\mathbf{z} = \mathrm{LLM}(q, p_j, \texttt{prompt}) = [z_0, z_1, \ldots, z_{K-1}], \tag{10}$$

where each z_k corresponds to the logit of a predefined discrete relevance label $r_k \in \{0, 1, \ldots, K-1\}$. We compute the scalar LLM-based relevance score using the *expected relevance* (ER) formulation:

$$\mathcal{S}_{i,j}^{\mathrm{LLM}} = \sum_{k=0}^{K-1} \left(\frac{e^{z_k}}{\sum_{j=0}^{K-1} e^{z_j}} \right) \cdot r_k. \tag{11}$$

We use one closed-source LLM (`GPT-4o`; OpenAI, 2024) and two open-source LLMs (`Qwen3-80B` and `Qwen3-8B`; Yang et al., 2025) for generating relevance judgments. Pointwise LLM-based scoring is applied to the top-R passages retrieved by DR to ensure fair comparability with GPR-LLM.

C Asymptotic Behavior of PW Scoring

This analysis aims to determine whether the advantage of GPR-LLM at low labeling budgets would disappear if PW were allowed substantially more

LLM calls. To this end, we compare PW with budgets ranging from 200 to 600 against GPR-LLM with a budget of 100 using the `Qwen3-80B`.

As shown in Table 4, increasing the PW budget yields diminishing returns at larger budgets. Despite using only 100 labels, GPR-LLM remains competitive with much higher-budget PW on most datasets. These results suggest that the benefit of additional PW labels eventually weakens, while GPR-LLM can achieve comparable performance with substantially fewer labels. Therefore, the advantage of GPR-LLM is not merely that it operates under a smaller budget, but that it uses labeled passages more efficiently than brute-force PW scoring in several settings.

D Effect of Sampling Quality on GPR-LLM

To examine how the quality of the labeled passage subset affects GPR-LLM, we conduct a controlled study in which the initial sampled subset is augmented with additional passages of different relevance levels. This experiment is intended to assess whether improved sampling can further enhance GPR-LLM, and whether the resulting gains depend on the choice of kernel.

We start from a baseline subset of 100 passages selected greedily according to dense-retrieval scores. We then incrementally augment this subset with 1–20 additional passages drawn from two supervision pools:

1. **High-relevance augmentation:** passages assigned high LLM relevance scores (2–3);
2. **Lower-relevance augmentation:** passages assigned low LLM relevance scores (0–1).

Results are shown in Figure 6. Adding lower-relevance passages leads to limited and inconsistent changes across all kernels, suggesting that weakly informative samples provide little benefit. By contrast, augmenting the subset with high-relevance passages yields consistent improvements only for the RBF kernel. The dot product and cosine kernels show little sensitivity to either augmentation type.

These findings suggest that the benefit of improved sampling quality depends on kernel expressiveness. When more informative anchor passages are available, the RBF kernel better captures multimodal relevance patterns due to its locality, which allows relevance to propagate within local neighborhoods. This observation supports moving beyond purely greedy DR-based sampling toward the proposed ϵ-greedy strategy, which introduces exploration to recover additional relevant passages.

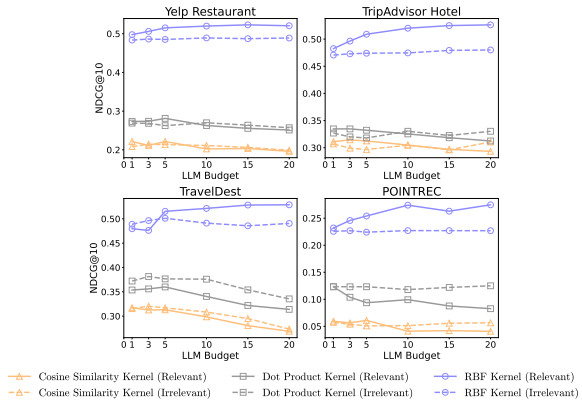

Figure 6: Effect of augmenting the sampled subset with high-relevance and lower-relevance passages on `TravelDest`. We report NDCG@10 as 1–20 additional passages are added. Only the RBF kernel exhibits consistent gains under high-relevance augmentation, whereas the dot product and cosine kernels remain relatively insensitive to both augmentation types.

E Impact of Different Embeddings

Table 5 presents the results obtained using the `msmarco-distilbert-base-tas-b` encoder with GPT-4o as the LLM backbone. The results follow the same overall trend observed with the `MiniLM-L6-v2` encoder, where GPR-LLM consistently outperforms both pointwise LLM-based scoring and other baseline methods across all datasets and LLM labeling budgets. The performance gains are most pronounced at smaller labeling budgets, indicating that GPR-LLM effectively leverages limited high-quality supervision to model the underlying multimodal relevance function. This consistent pattern across distinct embedding models demonstrates that the improvements of GPR-LLM stem from its core modeling design rather than encoder-specific characteristics.

F Hyperparameter Settings for GPR

F.1 Observation Noise Variance α.

In a noisy observation setting (i.e., the ground truth relevance is unknown), the performance of GPR can be affected by the variance of the Gaussian observation noise α. Thus, we examine the performance of GPR-LLM under nDCG@10 using different values of α in Figure 7.

Overall, we observe a non-monotonic relation-

Method	Budget	TravelDest				POINTREC				Yelp Restaurant				TripAdvisor Hotel			
		P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30
PW	200	0.435	0.448	0.345	0.376	0.199	0.221	0.123	0.216	0.420	0.460	0.338	0.409	0.336	0.452	0.204	0.495
	300	0.456	0.470	0.365	0.398	0.197	0.221	0.128	0.221	0.452	0.484	0.356	0.427	0.366	0.478	0.219	0.524
	400	0.469	0.487	0.381	0.416	0.210	0.222	0.122	0.211	0.479	0.514	0.372	0.446	0.381	0.492	0.225	0.537
	500	0.480	0.506	0.390	0.427	0.205	0.220	0.114	0.203	0.489	0.520	0.375	0.453	0.388	0.499	0.231	0.548
	600	0.482	0.509	0.391	0.429	0.218	0.230	0.117	0.207	0.493	0.524	0.376	0.453	0.391	0.501	0.234	0.552
GPR-LLM	100	**0.472**	**0.486**	**0.362**	**0.399**	**0.225**	**0.258**	**0.123**	**0.226**	**0.491**	**0.534**	**0.355**	**0.443**	**0.345**	**0.456**	**0.220**	**0.520**

Table 4: Comparison between higher-budget pointwise LLM scoring (PW) and GPR-LLM (RBF kernel, $\epsilon = 0.3$, τ includes all passages) using the `Qwen3-80B` backbone. PW is evaluated at budgets 200–600, while GPR-LLM uses 100 labels.

Budget	Method	TravelDest				POINTREC				Yelp Restaurant				TripAdvisor Hotel			
		P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30
N/A	DR	0.358	0.365	0.318	0.333	0.143	0.161	0.094	0.165	0.363	0.385	0.294	0.351	0.224	0.275	0.165	0.349
	BM25	0.234	0.238	0.237	0.239	0.025	0.032	0.025	0.038	0.309	0.327	0.236	0.283	0.205	0.257	0.153	0.325
25	PW	0.344	0.379	0.313	0.339	0.154	0.188	0.094	0.178	0.340	0.382	**0.289**	**0.355**	0.227	0.302	0.165	0.369
	GPR-LLM	**0.370**	**0.402**	**0.325**	**0.356**	**0.159**	**0.192**	**0.098**	**0.183**	**0.359**	**0.397**	0.272	0.343	**0.286**	**0.378**	**0.177**	**0.419**
50	PW	0.380	0.419	0.298	0.339	0.171	0.215	0.098	0.193	0.368	0.410	0.282	0.355	0.251	0.336	0.166	0.389
	GPR-LLM	**0.416**	**0.453**	**0.348**	**0.386**	**0.176**	**0.217**	**0.103**	**0.197**	**0.376**	**0.414**	**0.286**	**0.361**	**0.325**	**0.402**	**0.199**	**0.470**
100	PW	0.428	0.467	0.323	0.371	0.214	0.245	0.102	0.200	0.391	0.432	0.285	0.366	0.273	0.366	0.170	0.408
	GPR-LLM	**0.444**	**0.479**	**0.344**	**0.389**	**0.215**	**0.246**	**0.121**	**0.227**	**0.430**	**0.472**	**0.311**	**0.398**	**0.360**	**0.476**	**0.225**	**0.535**

Table 5: Comparison of BM25, DPR, Pointwise LLM-based Scoring (PW), and GPR-LLM (RBF kernel, $\epsilon = 0.3$, τ includes all passages) using the `msmarco-distilbert-base-tas-b` encoder with GPT-4o as the LLM backbone across four datasets. Bold values indicate the best performance in each column.

ship between α and performance. Extremely low noise levels (e.g., $\alpha = 10^{-4}$, effectively assuming near-perfect LLM relevance judgments) tend to *underperform*, likely due to overfitting to the few LLM judgments. Conversely, very high noise settings (e.g., $\alpha = 10$, treating LLM relevance judgments as extremely noisy) also degrade performance by underfitting the relevance signal.

In all cases, moderate noise variance yields the best results: performance generally peaks at intermediate α values (around 10^{-2} to 10^{-1} in our experiments) and remains fairly stable across a broad mid-range. This trend is consistent across datasets, though the optimal α can vary slightly by dataset (each dataset's curve achieves its maximum nDCG at a slightly different α). Crucially, adding a reasonable amount of observation noise improves generalization, but too much noise or none at all is detrimental.

F.2 Length Scale of the RBF Kernel.

The RBF kernel in GPR is defined as

$$k(\mathbf{v}, \mathbf{v}') = \exp\left(-\frac{\|\mathbf{v} - \mathbf{v}'\|^2}{2\ell^2}\right) \qquad (12)$$

where ℓ is the length scale hyperparameter that controls the smoothness of the learned function. Smaller values of ℓ yield more flexible, highly localized functions, while larger values impose smoother, more global behavior.

While ℓ is optimized using the L-BFGS-B algorithm, we empirically evaluate the impact of varying the initial value of ℓ over the range $[0.001, 0.1, 1, 10, 100, 1000]$. As shown in Figure 8, GPR-LLM is robust to different initializations of the length scale except under very small LLM labeling budgets, where label noise has a larger impact.

We also evaluate performance across different fixed values of ℓ without applying the L-BFGS-B algorithm. As shown in Figure 9, under a fixed α, the best performance achieved without L-BFGS-B is comparable to that with L-BFGS-B. This suggests that L-BFGS-B provides an efficient and effective method for kernel parameter selection in GPR-LLM.

F.3 Scaling of Relevance Scores.

Figure 10 illustrates the effect of varying the rating scale used by the LLM to provide relevance judgments, with results shown for all four datasets (nDCG@10 across different LLM budgets of labels). Across all four datasets, applying extreme scaling to the LLM relevance scores consistently

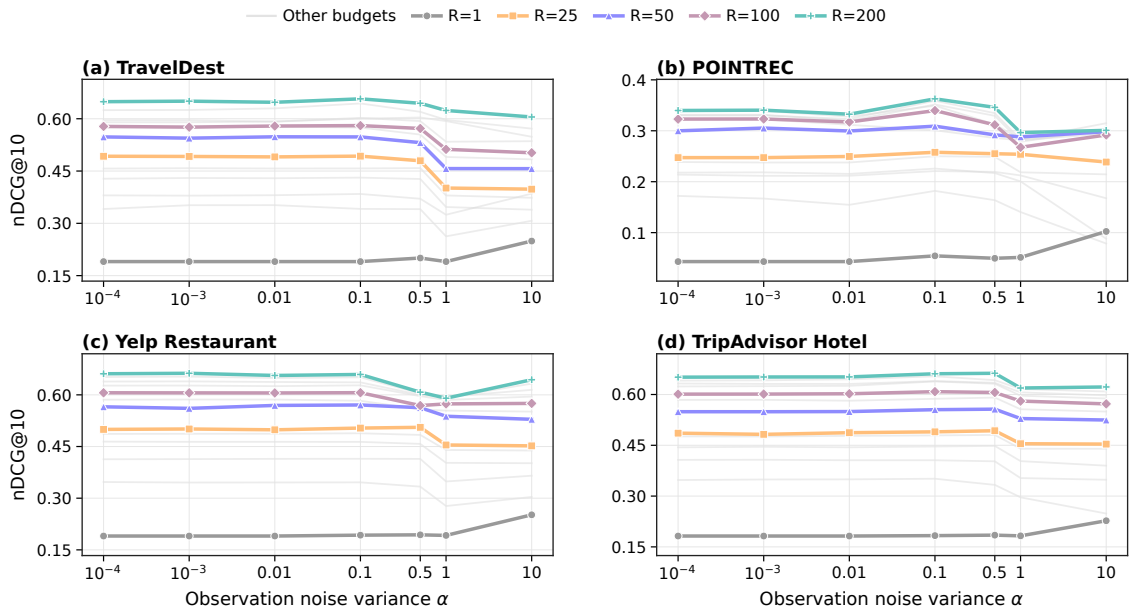

Figure 7: Sensitivity of GPR-LLM to the observation noise variance α across datasets and LLM labeling budgets. Moderate noise values generally provide the best trade-off between overfitting noisy LLM judgments and underfitting the relevance signal.

degrades GPR-LLM's nDCG@10 performance, whereas keeping the labels at or near their original scale yields the best results.

In particular, using the original rating range (scale factor = 1) produces the highest nDCG@10 on TravelDest, POINTREC, Yelp Restaurant, and TripAdvisor Hotel, for both low and high LLM budgets. Any substantial compression of the label range (e.g., a 0.1× factor) or expansion (e.g., a 10× factor) leads to a notable drop in ranking effectiveness across all LLM budgets.

This suggests that over-compressing the relevance scores blurs meaningful differences between items, while over-expanding them amplifies noise and overemphasizes minor relevance distinctions, in both cases hurting the GPR-LLM's ability to accurately rank items. Consequently, maintaining the original scale preserves the proper balance of signal to noise in the LLM labels and achieves the strongest overall nDCG@10 performance in the GPR-LLM framework.

G Hyperparameter Settings for Item-Level Relevance Aggregation

The final item-level score $\mathcal{S}_i$ for item $i \in \mathcal{I}$ is computed by aggregating the top-T passage-level scores:

$$\mathcal{S}_i = \phi\left(\left\{\, \hat{f}_q(\mathbf{v}_{p_j}) \mid p_j \in \mathrm{top}_T(\mathcal{P}_i, \hat{f}_q) \,\right\}\right), \tag{13}$$

where ϕ is the aggregation function (e.g., mean, max) and $\mathrm{top}_T(\mathcal{P}_i, \hat{f}_q)$ returns the top-T passages for item i ranked by the GPR relevance scoring function $\hat{f}_q$.

We vary the number T of top-ranked passages used in aggregation and evaluate its effect on item-level ranking performance. A small T may ignore informative passages, while a large T may include irrelevant noise. We vary T over $\{1, 3, 5, 10, 25, 50\}$ and set the LLM budget and ϵ to 100 and 0, respectively.

As shown in Figure 11, mean aggregation generally improves as more than one passage is used and performs best with a moderate number of top passages. Performance then decreases when many lower-ranked passages are included, suggesting that T around 3–10 best balances coverage and noise. Mean aggregation also consistently outperforms max aggregation once multiple passages are considered.

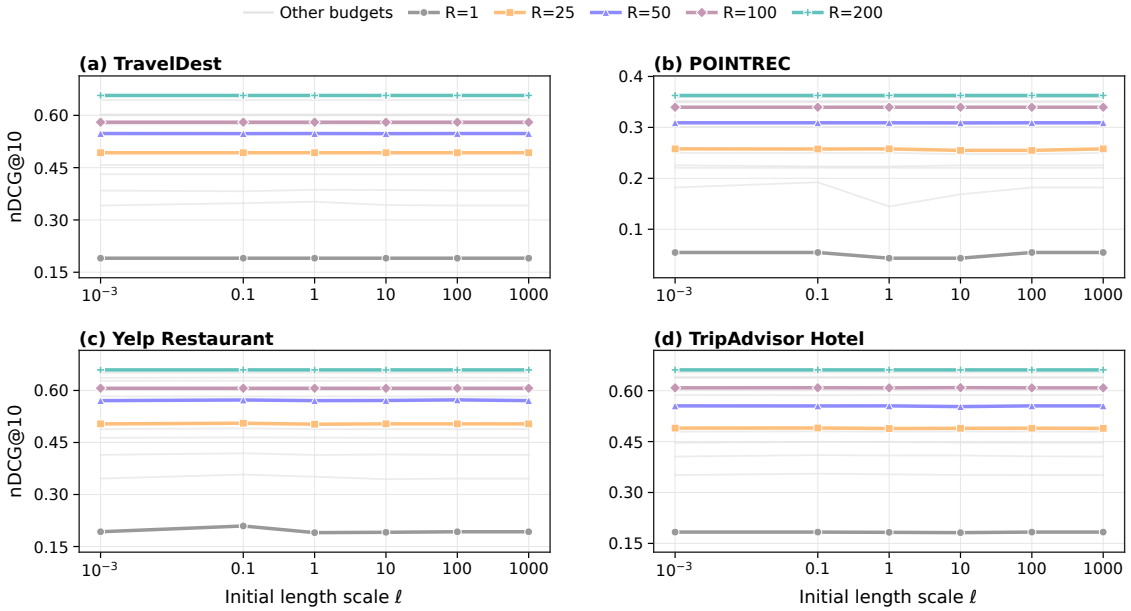

Figure 8: Sensitivity to the initial RBF length scale ℓ when L-BFGS-B is used to optimize the kernel hyperparameter. Results are shown across datasets and LLM label budgets.

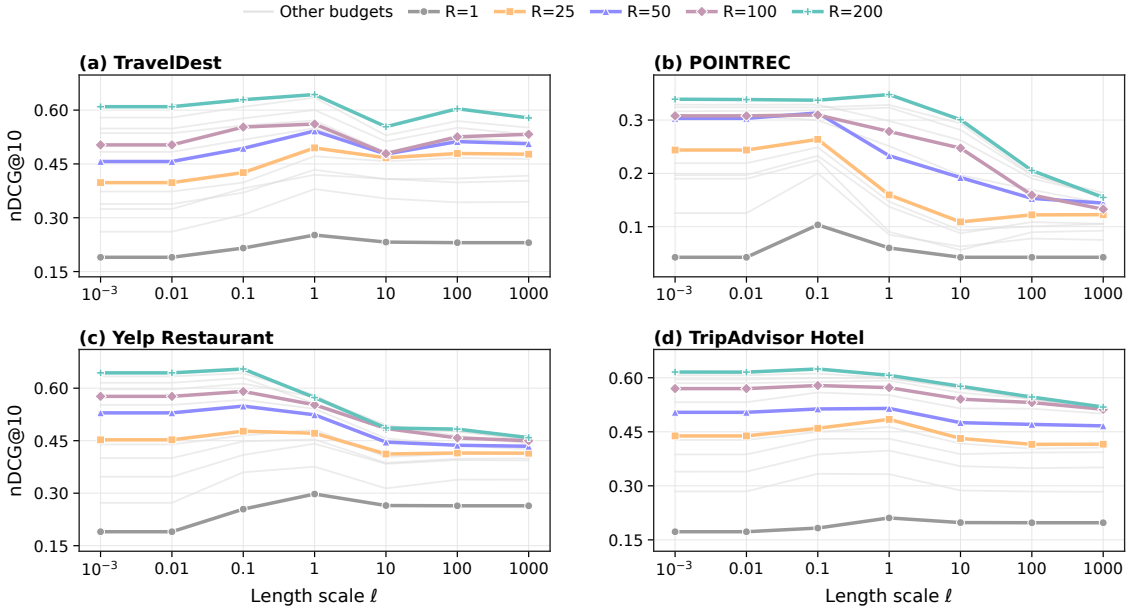

Figure 9: Sensitivity to fixed RBF length scale values ℓ without L-BFGS-B optimization. The panels show that carefully selected fixed length scales can approach optimized performance, but optimization avoids manual tuning across datasets and budgets.

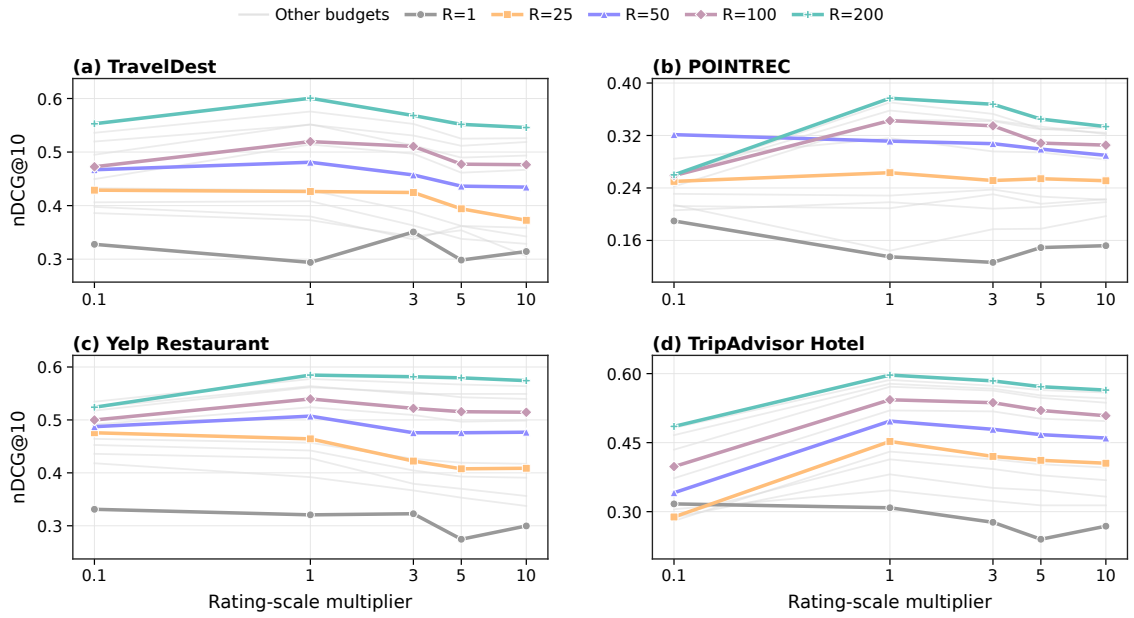

Figure 10: Effect of scaling the LLM relevance labels before fitting GPR-LLM. Across datasets, the original label scale generally provides the most stable performance, while strong compression or expansion degrades ranking quality.

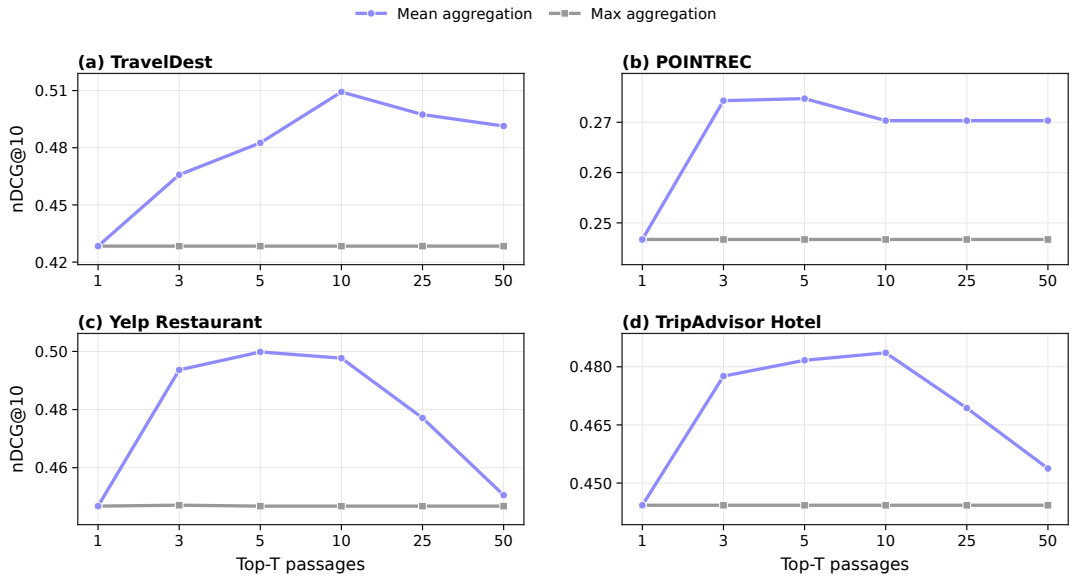

Figure 11: Performance of GPR-LLM under varying numbers of top-T passages and different aggregation functions for item-level relevance scoring. Moderate values of T retain informative evidence while avoiding excessive noisy passages.