PersonalQuery-Benchmark: Evaluating Retrieval Robustness to Personalized User Expression Differences

WENYU LIANG*, Monash University, Australia
HONGZHI YIN, University of Queensland, Australia
TONGTONG WU, Monash University, Australia
FAN LIU, Southeast University, China
KEXIN DAI, Monash University, Australia
WEIQING WANG, Monash University, Australia

Existing personalized information retrieval methods mainly model differences in user needs through user preferences and query intent, while paying little attention to how different users express the same retrieval need. Real query data typically lack stable user identities, long-term expression histories, and intent-matched counterparts, which impedes systematic evaluation of the impact of personalized expression differences on retrieval outcomes. To this end, we propose PersonalQuery-Benchmark (PQB). PQB models personalized syntactic expression and personalized writing errors from users' historical texts, constructs personalized syntactic expression queries and personalized writing-error queries, and validates benchmark quality through LLM assessment and independent human evaluation. PQB covers three Amazon product domains with 6,516 query instances, and is evaluated on five retrieval models representing lexical, learned sparse, dense bi-encoder, late-interaction, and hybrid architectures, plus two LLM-based ranking models. Results show that even when the information need is held fixed, personalized expression differences induce measurable ranking variation; different retrieval models exhibit clearly different sensitivities to syntactic variation and writing-error variation, while higher overall retrieval effectiveness does not necessarily correspond to stronger robustness against expression variation. PQB provides a new benchmark and evaluation framework for systematically studying personalized expression robustness. Code and dataset are available at https://github.com/cpxjazz123/PersonalQuery-Benchmark.

CCS Concepts: • **Information systems** → **Retrieval models and ranking**; *Test collections*.

Additional Key Words and Phrases: personalized retrieval, syntax, robustness, benchmark

1 Introduction

Existing personalized information retrieval mainly models what different users want through user preferences, historical behavior, and query intent, and uses these signals to improve the personalized relevance of retrieval results [20, 29, 30, 36, 58, 62, 65, 67]. However, the differences between users are not only about the underlying information need. Even when the retrieval intent and core semantic content are kept aligned, different users can still express the same need in different linguistic forms [4, 27, 54, 55]. These expression differences can further affect lexical matching and query representation [9, 46], and downstream ranking [70], so retrieval robustness depends not only on whether the system understands what users want, but also on whether it can stably handle how users express that need [22].

We refer to these linguistic differences between users, which do not depend on changes in underlying intent, as personalized expression differences. In other words, personalization involves not only what users want, but also how

*Corresponding author.

Authors' Contact Information: WENYU LIANG, Monash University, Melbourne, Australia, wlia0047@student.monash.edu; Hongzhi Yin, University of Queensland, Brisbane, Australia, h.yin1@uq.edu.au; TONGTONG WU, Monash University, Melbourne, Australia, tongtong.wu@monash.edu; Fan Liu, Southeast University, Nanjing, China, liufancs@gmail.com; KEXIN DAI, Monash University, Melbourne, Australia, hannahdai04@gmail.com; WEIQING WANG, Monash University, Melbourne, Australia, teresa.wang@monash.edu.

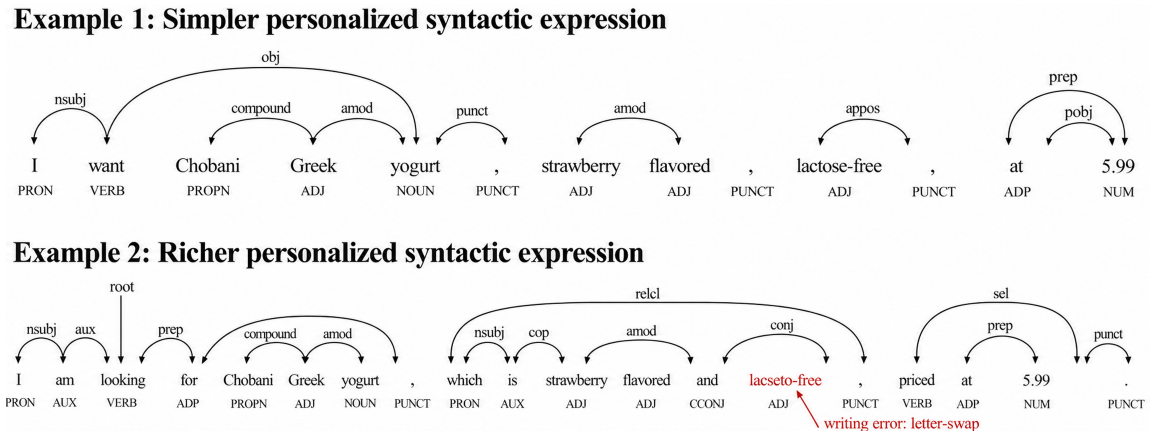


Fig. 1. Example of personalized expression differences under matched product intent and core product attributes, illustrating syntactic expression and writing-error variations.

users express the same need. In this paper, we aim to explore the impact of personalized expression differences on information retrieval methods. Existing benchmarks, however, do not directly support this question. The personalization signals in existing personalized retrieval datasets are dominated by preference, interaction behavior, or intent differences [8, 20, 58]. Another class of real-world query datasets without personalization contains natural user expressions but typically lacks stable user identities and sufficient historical expression evidence [1, 5, 7, 50, 69]. Motivated by this gap, this paper focuses on the following central question: **When users express the same retrieval need differently, how robust are retrieval systems to personalized expression differences?**

To answer this question, we first need to clarify what constitutes personalized expression. In real retrieval settings, user expression differences have already been established as a stable phenomenon: recurring writing habits and syntactic patterns can be observed in a user's historical texts, whereas different users differ identifiably in these respects [31, 32, 54, 55]. Such differences appear both at the level of word form and in how the same semantic content is organized into a sentence. We therefore focus on two conceptually distinct and complementary dimensions of personalized expression: personalized writing errors and personalized syntactic expression. The former captures a user's long-term writing-error habits [14, 31, 32], such as letter swaps, letter repetitions, or keyboard-adjacent substitutions, and different users can show stable differences in error occurrence frequency and in which error mechanisms they prefer. The latter captures the different ways in which users syntactically organize the same semantic content [34, 35, 53]: even when two queries share largely the same entities, attributes, and product requirements, they can still differ in phrase organization, modifier structure, and clause construction. Fig. 1 illustrates both dimensions together: the personalized syntactic expression query and its corresponding personalized writing-error query convey the same core product information (Chobani Greek yogurt, strawberry flavor, lactose-free, and a price of 5.99); the former uses a more direct attribute organization, whereas the latter opens with "I am looking for ...", uses a richer clause structure, and contains the letter-swap writing error "lacseto-free" for "lactose-free".

Based on this two-dimensional definition of personalized expression, this paper introduces **PersonalQuery-Benchmark (PQB)**, which evaluates how robust retrieval systems are to personalized expression differences. PQB

builds on the historical user texts in Amazon Reviews and models both dimensions for each user from their historical texts. The full construction procedure is described in Sections 3.2–3.4. We further validate the PQB along four dimensions— **Personalized Syntactic Attribution**, **Personalized Writing-Error Consistency**, **Query Quality**, and **Evaluation Agreement**—through LLM assessment and human evaluation.

On this basis, PQB covers three Amazon product domains—Musical Instruments, Video Games, and Baby Products—and evaluates personalized expression robustness on five retrieval models spanning lexical, learned sparse, dense bi-encoder, late-interaction, and hybrid architectures, together with two LLM-based ranking models. The results demonstrate that, even when the underlying product need is held fixed, personalized expression differences still cause clear ranking variation; different retrieval models show different sensitivities to syntactic variation versus writing-error variation, and overall retrieval effectiveness does not map in any simple way to expression robustness.

The main contributions of this paper are fivefold:

(1) **Personalized Expression Retrieval Setting.** We are the first to systematically investigate retrieval robustness to user-specific query expression differences derived from users' historical writing under a fixed information need, and we propose personalized syntactic expression and personalized writing errors as the two defining dimensions of personalized expression.

(2) **PersonalQuery-Benchmark.** We construct a personalized expression retrieval benchmark grounded in real users' historical expression evidence. Benchmark quality is validated along four dimensions—Personalized Syntactic Attribution, Personalized Writing-Error Consistency, Query Quality, and Evaluation Agreement—through LLM assessment and human evaluation. The benchmark covers three Amazon product domains and 6,516 query instances.

(3) **Personalized Expression Modeling and Query Construction.** The benchmark construction is grounded in users' real historical review texts. For personalized syntactic expression, PQB extracts structural rules such as dependency tuples and POS trigrams from historical expressions, learns representations that distinguish users' syntactic patterns through same-product supervised contrastive learning under a fixed product context, and fits a per-user low-rank-plus-diagonal Gaussian to capture the user's characteristic syntactic position and within-user variation. For personalized writing errors, PQB automatically extracts and aligns error annotations from historical reviews with GECToR and ERRANT, retains only errors that primarily alter surface form, and derives each user's error frequency and mechanism preference.

(4) **Retrieval Robustness Evaluation.** We systematically evaluate personalized expression robustness on five representative retrievers spanning lexical, learned sparse, dense bi-encoder, late-interaction, and hybrid architectures, together with two LLM-based ranking models, analyzing how syntactic variation and writing-error variation each affect ranking, and show that overall retrieval effectiveness does not correspond in any simple way to robustness to expression variation.

(5) **Personalized Expression Modeling Validation.** We further validate the effectiveness of personalized expression modeling through two experiments. For personalized syntactic expression, we compare the rank-one-plus-diagonal Gaussian with single-embedding representations, alternative Gaussian covariance structures, probabilistic distribution models, and nearest-neighbor methods. The results demonstrate our proposed rank-one-plus-diagonal Gaussian's stronger user attribution and discrimination, indicating its effectiveness in capturing user-specific syntactic patterns. For personalized writing errors, we compare user-specific error-mechanism sampling with global, uniform, character-random, and wrong-user profile baselines. Personalized

sampling achieves better user attribution, demonstrating that historical writing-error profiles capture individual error-mechanism preferences and support personalized query construction in PQB.

2 Related Work

The work most closely related to PQB falls into three strands: query variation and controlled generation, the effects of writing errors on retrieval, and personalized retrieval. Prior work has shown that, even when the underlying information need stays essentially unchanged, query formulation and surface-form variation can still influence retrieval behavior; at the same time, personalized retrieval and synthetic query generation have provided the foundations for exploiting user histories and constructing controlled query variations. These strands, however, are typically studied separately—one focuses on query reformulation, another on retrieval noise, another on user preference, and another on synthetic variation—and relatively little work constructs controlled expression differences from individual users' historical language behavior while fixing the target product and the retrieval intent. PQB lies at the intersection of these strands and studies personalized expression differences as an independent evaluation variable for retrieval ranking.

2.1 Query Variation and Controlled Query Generation

When the underlying information need stays essentially unchanged, query formulation can still affect retrieval effectiveness. Prior work has examined verbose queries, query variability, syntactic complexity, query clarity, and query-performance prediction [6, 11, 15, 26, 42], and has found that structural differences between human-written and LLM-generated queries can also change retrieval behavior [25, 66]. IR research also uses POS patterns, dependency structures, and syntactic organization to describe or model query structure [34, 35, 53]; meanwhile, authorship and writing-style research shows that structural linguistic patterns can reflect individual writing characteristics [54, 55]. Together, these studies indicate that syntactic form can influence retrieval behavior, yet existing retrieval studies usually examine syntactic variation at the level of query reformulation and rarely connect it to the long-term expression patterns of individual users.

Synthetic query generation provides a complementary route to controlled query variation. Related methods have been widely used for training-data augmentation, semantic annotation, and retrieval benchmark construction [5, 7, 18, 50], and other studies have increased expression diversity through demographic attributes, personality traits, query rewriting, or term-level controls [1, 2, 4, 27, 41, 69]. These methods can generate large numbers of query variants efficiently under semantic constraints, but their variation controls usually come from prompts, generic rewriting mechanisms, or predefined attributes, and therefore need not correspond to the expression behavior that individual users develop over time. PQB likewise uses controlled generation to fix product semantics, but its personalization signals and query-selection constraints are derived from historical user texts: historical expressions are used to estimate personalized syntactic distributions and further guide candidate-query selection.

2.2 Writing Errors and Retrieval Performance

Writing errors and orthographic variation can also change retrieval matching behavior. Prior work has analyzed natural spelling variation, spelling correction, and regional conventions [12, 52, 68], and has tested the robustness of neural retrieval models to misspellings through character-level, token-level, or adversarial perturbations [21, 37, 38, 47, 61]. These studies show that, even when the underlying information need is preserved, local surface-form changes can still cause retrieval degradation.

Existing robustness evaluations, however, usually model writing errors as population-level or task-level perturbations, such as uniform corruption probability, fixed error mechanisms, or adversarial modifications, and rarely model long-term error frequency, mechanism preference, and context-conditioned patterns from the historical writing behavior of individual users. PQB therefore automatically identifies writing-error evidence from historical user texts, builds user-specific writing-error profiles, and applies constrained character-level editing with these profiles to the already determined personalized syntactic expression queries, thereby forming same-user personalized syntactic expression and personalized writing-error query pairs. This setting allows the effects of writing errors on retrieval outcomes to be evaluated while keeping the target product, the retrieval intent, and the original syntactic expression essentially unchanged.

2.3 Personalized Retrieval and User Modeling

Personalized retrieval typically relies on user embeddings, interaction histories, session context, long-term behavior, and learning-to-rank signals to improve personalized relevance matching [3, 20, 29, 30, 58, 65, 67]; datasets such as JDsearch further provide real queries alongside user-interaction histories [36]. These methods use personalization signals to model the preferences, behaviors, or intents of different users, that is, they focus on what different users want and change relevance estimation or ranking accordingly.

PQB addresses a different personalization dimension. Instead of using user expression history to change the relevance target, it learns personalized expression signals from users' historical texts and asks whether ranking still changes when users express the same need differently. Specifically, PQB constructs same-product cross-user personalized syntactic expression query cohorts to compare the personalized syntactic expressions of different users on the same product, and constructs same-user personalized syntactic expression and personalized writing-error query pairs to evaluate personalized writing-error perturbations. Personalization in PQB therefore serves mainly as an expression variable under study rather than as an additional user signal for improving relevance matching.

Overall, prior work has separately shown that query formulation, writing errors, user histories, and controlled query generation all affect retrieval modeling, but these directions are usually independent of one another. PQB further organizes them into a unified analytical framework: it learns personalization signals from the historical language behavior of individual users and constructs cross-user syntactic variation and within-user writing-error variation separately, so as to systematically analyze how personalized expression differences affect retrieval outcomes.

3 Benchmark Construction

PQB's construction goal is to introduce personalized variation built from real users' historical expressions while holding retrieval intent and core product semantics fixed. The construction pipeline therefore has four stages. Section 3.1 fixes the benchmark's data sources and sample structure and unifies product metadata with user histories as the basis for the rest of the pipeline. Section 3.2 learns personalized syntactic distributions from each user's historical texts. Section 3.3 generates candidate queries under shared semantic constraints and selects one personalized syntactic expression query per user that matches the target user's syntactic expression profile. Section 3.4 constructs the corresponding personalized writing-error query based on each user's historical writing-error profile. The full benchmark construction pipeline is shown in Fig. 2.

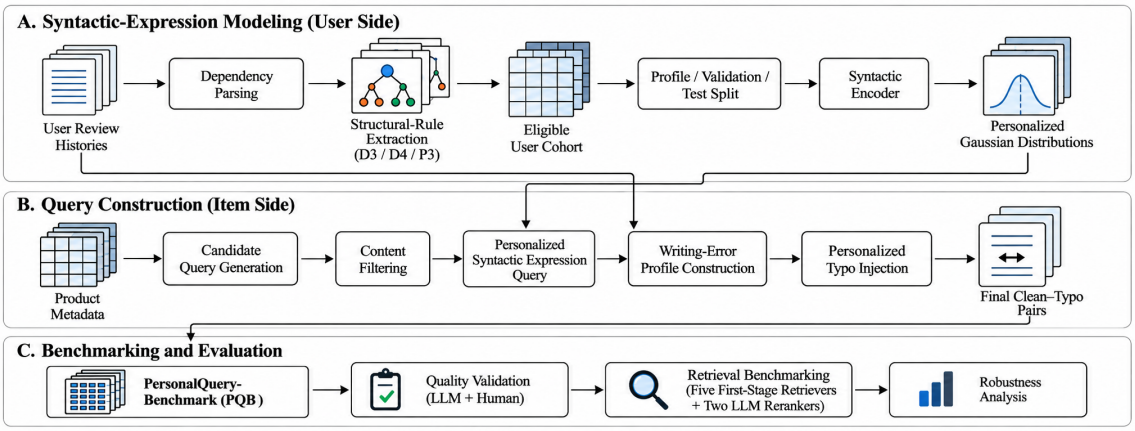


Fig. 2. PQB benchmark construction pipeline from product metadata and user review histories to personalized syntactic expression queries and personalized writing-error queries.

3.1 Dataset Construction Framework

As the first stage of the construction pipeline, PQB fixes the benchmark's data sources, retrieval targets, and user histories used for personalized expression modeling. PQB is built on the Amazon Reviews 2023 dataset, which provides user reviews, user–item interactions, and product metadata [23]. We select three product domains—Musical Instruments, Video Games, and Baby Products—referred to as Musical, Video, and Baby. PQB organizes benchmark samples by product ASIN. For each user, the products they have previously reviewed are treated as the target products the user has purchased or interacted with, and the corresponding ASINs serve as retrieval targets. PQB then links each ASIN's product metadata with the historical reviews of its associated users, so that product-side semantic information and user-side expression evidence become available together as a unified basis for personalized syntactic expression modeling and writing-error modeling.

3.1.1 Product Constraints and User History. After fixing the target products and their associated users, the next step is to separate the information that controls product semantics from the information that provides personalized expression evidence. The product side fixes the core product content expressed by each query. For each ASIN, PQB merges the title, description, and features into a retrieval document and selects the main product attributes from the cleaned metadata as query-generation constraints. Attribute-conditioned generation helps limit the semantic variation across different queries [24]. Corresponding to the product-side semantic control, the user side provides personalized expression evidence. PQB aggregates the historical reviews of the users linked to each target ASIN and uses these historical texts for two modeling processes: learning each user's syntactic expression patterns and estimating each user's writing-error habits [54, 57].

3.1.2 Benchmark Sample Definition. After the product-side semantic constraints and the user-side expression histories are determined, PQB organizes them into a unified benchmark instance. Each benchmark row contains a target product a, a selected user u, and the personalized syntactic expression query $q_{u,a}$. For records where a valid writing-error variant is successfully generated, the corresponding personalized writing-error query $\tilde{q}_{u,a}$ is additionally provided. Table 1 defines the fields.

Table 1. Structure of a PQB benchmark sample.

Symbol	Meaning
a	Target product
u	Target user
$q_{u,a}$	Personalized syntactic expression query
$\tilde{q}_{u,a}$	Personalized writing-error query

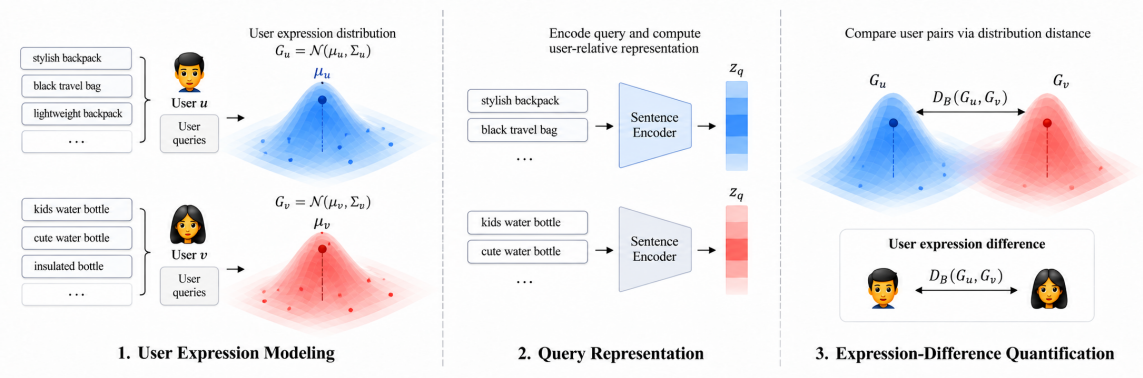


Fig. 3. Personalized syntactic expression quantification: user-level expression modeling, query representation, and expression-difference measurement between user pairs.

3.2 Expression-Difference Quantification

After Section 3.1 fixes the target products, semantic constraints, and user histories, PQB further converts the syntactic expression patterns in each user's historical texts into comparable quantitative representations and constructs personalized syntactic distributions. The complete quantification pipeline is shown in Fig. 3. Concretely, Section 3.2.1 extracts structural rules from historical expressions and builds a unified syntactic feature representation; Section 3.2.2 filters eligible user–product pairs by historical sentence count; Section 3.2.3 applies same-product supervised contrastive learning, pulling the historical expressions of the same user together under a fixed product context and separating the syntactic patterns of different users, so as to learn representations that characterize personalized syntactic differences; Section 3.2.4 partitions historical expressions into profile and validation subsets; Section 3.2.5 fits a personalized low-rank-plus-diagonal Gaussian to the profile representations, capturing each user's typical syntactic position and within-user variation; Section 3.2.6 examines the validity and stability of each personalized Gaussian on held-out validation data and through repeated subsampling. Together these steps yield validated personalized syntactic distributions, which Section 3.3 then uses for personalized syntactic expression query generation.

3.2.1 Structural Rule Extraction and Representation. To convert the historical user texts obtained in Section 3.1 into a representation that mainly describes syntactic organization while minimizing the influence of specific lexical content, PQB first extracts structural rules from each historical sentence. Prior authorship and writing-style research has shown that dependency structures and POS-based patterns provide useful information for characterizing individual language expression [40, 44, 55]. We use three kinds of structural rules: dependency 3-tuples (D3), dependency 4-tuples with

grandparent context (D4), and part-of-speech trigrams (P3). A D3 rule is written as

$$(\text{parent POS}, \text{dependency relation}, \text{child POS}),$$

for example (NOUN, amod, ADJ). A D4 rule adds the grandparent context to a D3 rule:

$$(\text{grandparent POS}, \text{parent POS}, \text{dependency relation}, \text{child POS}),$$

for example (VERB, NOUN, amod, ADJ). A P3 rule is the POS sequence of three consecutive tokens, for example (DET, ADJ, NOUN).

After obtaining sentence-level structural rules, PQB builds a unified global rule vocabulary so that historical expressions from different users can be mapped into the same feature space. We aggregate the unique D3, D4, and P3 rule types observed across all historical sentences and record the sentence frequency (SF) of each rule. Only rules with $SF \geq 2$ are retained to reduce the sparse noise introduced by very low-frequency structural patterns. Based on this vocabulary, each historical sentence is represented as a 21,737-dimensional sparse rule vector, whose nonzero dimensions correspond to the retained structural rules actually present in that sentence. This representation then serves as the input to the Section 3.2.3 syntax encoder.

3.2.2 Pre-Encoder User–Product Pair Eligibility. After Section 3.2.1 produces a unified structural representation, not every user–product pair has enough historical expressions to support reliable personalized modeling. Before training the syntax encoder, PQB therefore filters eligible user–product pairs by the historical sentence count available for each user under each product. Letting $h_{u,a}$ denote the number of historical sentences of user u under product a, the eligibility requirement is defined as

$$h_{u,a} \geq h_{\min}. \tag{1}$$

This criterion depends only on the historical expression evidence observable before encoder training. It does not use learned representations, Gaussian quality, or downstream retrieval results, and therefore does not condition sample retention on later model performance.

To choose $h_{\min}$, we compare, across the Musical, Video, and Baby domains, the number of eligible user–product pairs at different minimum-history thresholds and the number of products that still support same-product cross-user comparison, i.e., ASINs with at least two eligible users. The results are reported in Appendix Table 8. We therefore select $h_{\min} = 30$ as the empirical operating point, which retains enough historical expression evidence per user–product pair while preserving as much cohort coverage as possible for same-product personalized comparison.

User–product pairs satisfying $h_{u,a} \geq 30$ are then grouped into same-product user cohorts by parent ASIN. For each product a, only cohorts containing at least two eligible users are retained. This constraint directly matches the learning objective of Section 3.2.3: the syntax encoder must distinguish the syntactic expression patterns of different users under a fixed product context, where historical sentences from the same user serve as positive samples and sentences from other users under the same product provide cross-user negative samples. A product with only one eligible user cannot form such a same-product personalized comparison and is excluded from subsequent syntactic representation learning.

3.2.3 Syntactic Representation Learning. After Section 3.2.2 retains eligible user–product pairs with sufficient historical evidence, PQB follows the splitting strategy defined in Section 3.2.4 to partition each eligible user's historical sentences into profile and validation subsets, and uses only the profile subset for syntax encoder representation learning. After obtaining the structural-rule representations, PQB adopts same-product supervised contrastive learning: historical expressions from the same user under the same product serve as positive samples, whereas expressions from

other users under that product serve as negative samples. This design encourages the representation space to reflect cross-user syntactic differences under a matched product context. For the i-th profile sentence of user u under product a, the latent representation is defined as (detailed hyperparameter settings are provided in Appendix A.2):

$$z_{u,a,i} = f_\theta(c_{u,a,i}) \in \mathbb{R}^d. \tag{2}$$

To preserve personalized syntactic differences within a fixed product context, training uses a supervised contrastive objective. For an anchor sentence i, positive samples come from other historical sentences of the same user under the same ASIN within the profile subset, while negative samples come from profile sentences of other eligible users under the same ASIN. Each eligible user is used as the anchor in turn, which lets the model learn user-specific syntactic representations under a fixed product context. All latent representations are L2-normalized, so the inner product between representations corresponds to cosine similarity.

Concretely, let $P(i)$ denote the positive samples belonging to the same user–product pair as anchor i, and let $A(i)$ denote the samples in the same ASIN cohort, other than the anchor, that participate in the contrastive comparison. Training uses

$$\mathcal{L} = \sum_{i=1}^{N} -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i^\top z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i^\top z_a/\tau)}. \tag{3}$$

Through this cohort-based sampling, the encoder pulls historical expressions from the same user's profile subset closer together within a product and pushes away syntactic representations from other users. Section 3.2.4 describes how the profile and validation subsets are formed and how they are used in Gaussian fitting and validation.

3.2.4 Profile–Validation Split. For each eligible user–product pair, PQB uses a deterministic split based on the sentence identifiers to divide historical sentences 80% / 20% into profile and validation subsets. The profile subset is used for syntax encoder learning and personalized Gaussian fitting; the validation subset does not participate in parameter estimation and is reserved for the held-out validity checks of Section 3.2.6. This fixed split guarantees that the same sentence never enters both fitting and validation and lets all experiments reproduce the same held-out samples. Because the eligibility threshold is $h_{u,a} \geq 30$, every pair's validation subset contains at least five held-out representations.

3.2.5 Per-User Low-Rank-plus-Diagonal Gaussian Modeling. After Section 3.2.4 yields the profile representations, PQB aggregates the profile representations of each user across all of that user's eligible product contexts and fits a personalized syntactic distribution at the user level.

A single embedding vector can represent a user's typical syntactic style but cannot describe the variation among the different syntactic expressions that the same user uses over time. Prior studies have adopted Gaussian embeddings to represent variability beyond fixed vector representations [28, 60], while distribution-based personalized recommendation has used Gaussian means and covariances to characterize user preferences and relationships between latent dimensions [17]. Motivated by these approaches, PQB adopts a Gaussian distribution as a tractable approximation of each user's historical syntactic representations, rather than assuming that these representations strictly follow a Gaussian distribution. The mean characterizes the user's typical syntactic style, while the covariance captures within-user variation. This representation also provides a unified basis for Gaussian likelihood scoring and distance-based query selection in Section 3.3. The structure of the covariance determines how much of the within-user variation this distribution can retain: a full covariance matrix is the most general choice, but each user provides only a limited

number of historical expressions, so estimating it directly introduces substantial estimation noise, whereas keeping only the dimension-wise variances is more stable yet discards the correlated variation across latent dimensions. The rank-one-plus-diagonal structure offers a compromise between these two alternatives: a low-rank component along a principal direction captures the correlated variation among dimensions, and diagonal residuals model the remaining dimension-wise variation, so the within-user variation structure is retained at a tractable estimation complexity.

For user u, let the profile subset contain n_u latent syntactic representations:

$$Z_u^{\text{profile}} = \{z_{u,1}, \dots, z_{u,n_u}\}, \qquad z_{u,i} \in \mathbb{R}^{16}. \tag{4}$$

PQB uses the empirical mean to describe the central position of the user's historical syntactic expressions in the latent space:

$$\mu_u = \frac{1}{n_u} \sum_{i=1}^{n_u} z_{u,i}. \tag{5}$$

Motivated by low-rank probabilistic covariance modeling [56], PQB adopts the rank-one-plus-diagonal covariance structure:

$$\Sigma_u = \lambda_u v_u v_u^\top + \operatorname{diag}(\sigma_u^2), \tag{6}$$

where v_u denotes the main variation direction of the user's syntactic representations, λ_u is the variance contribution along that direction, and σ_u^2 is the residual variance along each latent dimension. These parameters are estimated from the empirical covariance C_u of each user's profile representations. Specifically, v_u is the eigenvector associated with the largest eigenvalue λ_u of C_u, and the dimension-wise residual variance is obtained from the diagonal of the covariance remaining after subtracting the rank-1 component,

$$\sigma_u^2 = \operatorname{diag}\left(C_u - \lambda_u v_u v_u^\top\right), \tag{7}$$

with each residual variance lower-bounded by 10^{-4} for numerical stability.

Based on these parameters, the personalized syntactic distribution of user u is represented as

$$z \mid u \approx \mathcal{N}(\mu_u, \Sigma_u). \tag{8}$$

Here, μ_u captures the user's characteristic syntactic position, while Σ_u describes the principal variation direction and dimension-wise residual variation. The fitted distribution is subsequently evaluated for numerical validity, held-out calibration, and estimation stability in Section 3.2.6.

3.2.6 Gaussian Validity Filter. After Section 3.2.5 produces per-user personalized Gaussians, PQB further applies held-out validation data and runs three sequential checks on each Gaussian profile: (1) **numerical validity**, (2) **held-out calibration**, and (3) **repeated-sampling stability**. Only profiles that pass all three checks are retained and used for the personalized syntactic expression query generation in Section 3.3.

(1) **Numerical validity.** Checks whether the Gaussian parameters and the validation Mahalanobis distances can be stably computed: the rank-1 variance and all diagonal residual variances must be positive, the Gaussian log-determinant must be finite, and the squared Mahalanobis distances for the validation representations must

be computable. This step primarily excludes profiles with degenerate covariance matrices or other numerical failures.

(2) **Held-out calibration.** To evaluate how well a user's Gaussian fits unseen syntactic expressions, we compute the Mahalanobis distances of held-out validation representations and require that at least 80% of each user's validation samples satisfy $D^2 \leq \chi^2_{16,0.95} = 26.296$.

(3) **Repeated-sampling stability.** To evaluate the stability of user Gaussian distributions against variations in historical samples, PQB repeatedly performs random subsampling and refits the Gaussians to examine their numerical validity, held-out distance ranking, and acceptance-region consistency. Specific filtering thresholds and stage-wise filtering results are provided in Appendix Table 10 and Table 11. Empirical results show that the vast majority of Gaussian profiles satisfy the numerical validity requirement, with held-out calibration serving as the primary filtering stage, while repeated-sampling stability further removes distributions that are sensitive to historical sample variations. All three domains exhibit a consistent filtering trend, indicating that this filtering procedure improves the stability of the final Gaussian profiles while retaining high user coverage.

3.3 Personalized Syntactic Expression Query Generation

After Section 3.2 yields fitted syntactic distributions, PQB constructs history-informed queries under matched product attributes. Section 3.3.1 describes the frozen candidate pool and its shared product constraints. Section 3.3.2 encodes candidates with the fixed syntax encoder and applies exact Gaussian scoring and feasibility-aware pair-conditional selection.

3.3.1 Content-Controlled Candidate Query Generation. Section 3.3.1 first constructs a candidate query pool under strict lexical and semantic constraints. For each target product, PQB cleans the product metadata by removing missing, duplicate, and unsuitable attributes, and normalizes attribute values with format anomalies. It then fixes five core retrieval terms for each product, which serve as shared lexical constraints for all candidate queries of that product. Each query must fully retain these five core retrieval terms, with each term appearing exactly once; deletion, repetition, replacement, or synonymous rewriting is not allowed. Queries may not introduce additional product attributes or new product semantic content. Variation between queries is allowed only in function words, connective expressions, modification structures, and clause organization, i.e., in syntactic realization.

PQB uses Qwen2.5-0.5B-Instruct [64] to generate first-person shopper queries under a unified generation configuration across the three product domains. The above lexical and semantic constraints guide generation and are enforced through post-generation content validation. PQB retains only queries in which all five core retrieval terms appear exactly once and no additional product semantic content is introduced. For products with insufficient valid candidates, PQB continues iterative candidate generation and adds validated queries to the candidate pool. Subsequently, the syntax encoder from Section 3.2 encodes the candidate queries, and personalized syntactic expression query selection is completed in Section 3.3.2. Detailed generation parameters are provided in Appendix Table 12.

To keep the comparison within one product about expression form rather than meaning, PQB further constrains the semantic similarity of the retained query pairs. For two queries q_1 and q_2 of the same target product, $\mathrm{SemanticSim}(q_1, q_2)$ denotes the cosine similarity between their sentence embeddings computed with the all-mpnet-base-v2 encoder [48], and PQB retains only query pairs that satisfy $\mathrm{SemanticSim}(q_1, q_2) \geq 0.9$. The constraint is enforced during construction, so the query pairs evaluated in Section 4 differ in syntactic realization rather than in semantic content, with the five core retrieval terms remaining the shared lexical basis of both queries.

3.3.2 Gaussian-Based Personalized Syntactic Expression Query Selection. PQB uses the personalized syntactic distributions learned in Section 3.2 to select representative queries for different users from the candidate query pool of the same product. Unlike random sampling from the user population, PQB focuses on user pairs with clear syntactic expression differences, in order to construct challenging query samples that can evaluate retrieval models' sensitivity to cross-user expression variation. To this end, the query selection process jointly considers how well candidate queries fit each user's syntactic distribution, as well as the differences between different users' syntactic distributions.

For a product a, let $\mathcal{U}_a$ be the set of users with valid Gaussian profiles and let $\mathcal{Q}_a$ be the candidate query set constructed in Section 3.3.1. All candidate queries are mapped by the trained syntax encoder into a 16-dimensional representation space, giving the syntactic representations z_q. For a user u, the personalized syntactic distribution is represented as $G_u = \mathcal{N}(\mu_u, \Sigma_u)$. The log-probability density of a candidate query q under this distribution is defined as

$$\log p_u(z_q) = -\frac{1}{2}\left[(z_q-\mu_u)^\top \Sigma_u^{-1}(z_q-\mu_u) + \log|\Sigma_u| + 16\log(2\pi)\right]. \tag{9}$$

To select queries from the candidate pool of the same product that match the syntactic expression characteristics of different users, PQB filters candidates by how well they fit the Gaussian distributions of two users. For a user pair $\{u, v\}$, the candidate queries of user u must lie inside the acceptance region of its own Gaussian, outside the acceptance region of user v, and satisfy a cross-user log-density margin. Their eligible query set is defined as

$$\begin{aligned}\mathcal{A}_u(v) = \{q \in \mathcal{Q}_a \,:\, & D_u^2(z_q) \le \tau, \\ & D_v^2(z_q) > \tau, \\ & \log p_u(z_q) - \log p_v(z_q) \ge 1\},\end{aligned} \tag{10}$$

where

$$D_u^2(z_q) = (z_q-\mu_u)^\top \Sigma_u^{-1}(z_q-\mu_u) \tag{11}$$

denotes the squared Mahalanobis distance of a candidate query under user u's Gaussian distribution, and $\tau = \chi^2_{16,0.95} = 26.296$ is the Gaussian reference acceptance threshold, which implements a consistent candidate-screening rule across users. The same filtering rule applies symmetrically to user v, yielding its eligible query set $\mathcal{A}_v(u)$.

For each user, PQB selects the query with the highest Gaussian log-probability density from its eligible query set:

$$\hat{q}_{u,a}^{(u,v)} = \arg\max_{q \in \mathcal{A}_u(v)} \Big(\log p_u(z_q),\ \log p_u(z_q) - \log p_v(z_q),\ -i(q)\Big), \tag{12}$$

where $i(q)$ denotes the index of the candidate query in the original candidate pool, and the triple is maximized lexicographically. The rule first prefers the query that best matches the target user's syntactic distribution, then uses the cross-user density margin and the candidate index to resolve ties.

PQB then enumerates the user pairs under each product. A pair is feasible when both users have at least one eligible query and the finally selected queries have different texts and different source indices. The feasible user pair set for product a is defined as

$$\begin{aligned}\mathcal{F}_a = \{\{u, v\} \subseteq \mathcal{U}_a \,:\, & \mathcal{A}_u(v) \neq \emptyset, \\ & \mathcal{A}_v(u) \neq \emptyset, \\ & \hat{q}_{u,a}^{(u,v)} \neq \hat{q}_{v,a}^{(u,v)}\},\end{aligned} \tag{13}$$

where the final condition states that the selected query texts differ, and their source indices are also guaranteed to differ during selection.

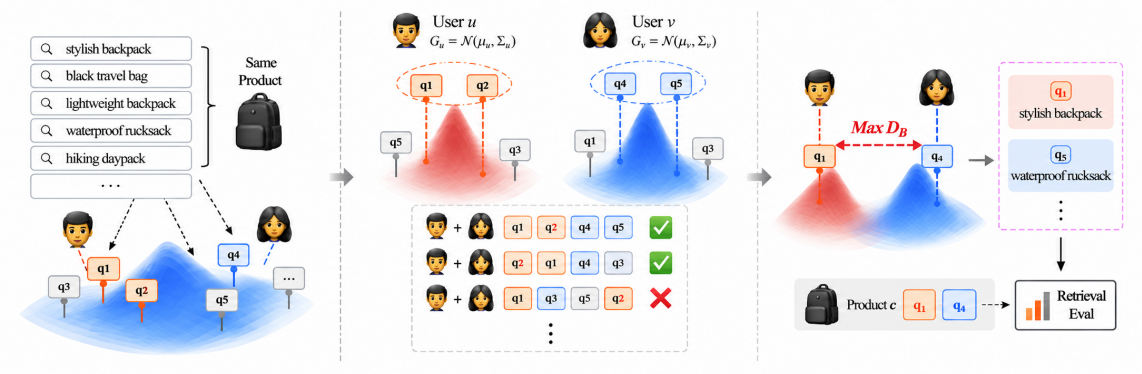


Fig. 4. Gaussian-based personalized syntactic expression query selection from a shared candidate pool: pair-conditioned eligibility filtering, feasible user-pair enumeration, and Bhattacharyya-maximizing pair selection.

To construct evaluation samples with large inter-user syntactic expression differences, PQB selects the user pair with the largest distance between Gaussian distributions among the feasible pairs:

$$\{u_a^*, v_a^*\} = \underset{\{u,v\}\in\mathcal{F}_a}{\arg\max} D_{\mathrm{B}}(G_u, G_v), \tag{14}$$

where D_{B} denotes the Bhattacharyya distance:

$$\begin{aligned} D_{\mathrm{B}}(G_u, G_v) = &\frac{1}{8}(\mu_u - \mu_v)^{\top}\bar{\Sigma}^{-1}(\mu_u - \mu_v) \\ &+ \frac{1}{2}\log\frac{|\bar{\Sigma}|}{\sqrt{|\Sigma_u||\Sigma_v|}}, \end{aligned} \tag{15}$$

and $\bar{\Sigma} = (\Sigma_u + \Sigma_v)/2$. When several feasible user pairs share the same distributional distance, selection is deterministic according to the sorted user identifiers.

The Bhattacharyya distance simultaneously considers the centers and covariance differences of the users' syntactic distributions, and can therefore measure the distinguishability between different users' expression distributions. By maximizing this distance, PQB preferentially retains feasible user pairs with larger distributional differences, enhancing the distinguishability of evaluation samples at the syntactic expression level.

Through the above procedure, PQB first screens user pairs satisfying the personalized query matching constraints, then selects the two users with the largest Gaussian distributional distance, and respectively selects from the candidate query pool the query that best matches each user's personal syntactic distribution. These queries reflect different users' syntactic expression features while expressing the same product-related retrieval content, providing a unified product context for subsequent analysis of cross-user retrieval variability.

As shown in Fig. 4, PQB first filters candidate queries according to the users' Gaussian syntactic distributions, then maximizes the Bhattacharyya distance among the user pairs satisfying the query feasibility constraints, and finally selects a representative personalized syntactic expression query for each of the two users of the same product.

3.4 Personalized Writing-Error Generation

After Section 3.3 constructs the personalized syntactic expression queries, PQB further uses the writing-error characteristics in the users' historical reviews to generate personalized character-level perturbations for the queries. The process first extracts writing errors and their syntactic contexts from the users' historical reviews to build personalized error profiles, then applies local character modifications to these queries according to the users' historical transformation records and error-mechanism preferences, forming personalized writing-error queries for evaluating changes in retrieval performance.

3.4.1 Personalized Writing-Error Profile Construction. PQB extracts writing-error characteristics from the users' historical reviews to construct personalized error profiles. Specifically, GECToR [45] first produces the corrected forms of the texts, and ERRANT [10] then aligns the original and corrected texts to identify and annotate the edit operations. For the extracted historical edits, PQB records the error type, the transformation form and the associated syntactic context, and counts the historical transformations that are available for character-level perturbation.

Unlike continuous syntactic representations, writing errors consist of discrete character-level editing operations. Their personalized characteristics involve not only how frequently users make errors, but also which error mechanisms they prefer and how these mechanisms relate to syntactic contexts. A shared global error distribution overlooks individual differences in mechanism preferences, whereas a context-independent user distribution cannot capture the relationship between writing errors and local syntactic structures. Meanwhile, fine-grained contexts may contain only limited historical error evidence for individual users. PQB therefore adopts a user-specific, context-conditioned error-mechanism distribution estimated from historical frequencies, while retaining each user's overall mechanism preferences and observed character transformations. To characterize a user's writing-error preferences under different syntactic structures, PQB further tabulates the occurrences of each error mechanism in each syntactic context. For user u, let $n_u(e, r)$ denote the number of historical occurrences of error mechanism e in syntactic context r; the context-conditioned error-mechanism distribution is then defined as

$$P_u(e \mid r) = \frac{n_u(e, r)}{\sum_{e' \in \mathcal{E}} n_u(e', r)}. \tag{16}$$

where $\mathcal{E}$ denotes the set of error mechanisms. This distribution describes the relative frequency of different error mechanisms for the user in a given syntactic context. In parallel, PQB aggregates the user's historical edits to compute the overall character-level error-mechanism preferences and historical transformation frequencies. A user's writing-error profile therefore consists of the historical edits, the global error-mechanism preferences and the context-conditioned error distribution, and is used for the personalized writing-error query generation in Section 3.4.2. The per-domain user profile statistics are reported in Appendix Table 13.

3.4.2 Personalized Writing-Error Query Generation. After the personalized writing-error profiles are obtained, PQB perturbs the personalized syntactic expression queries selected in Section 3.3 at the character level. For user u and the corresponding product a, let the personalized syntactic expression query be $q_{u,a}$ and the generated personalized writing-error query be $\tilde{q}_{u,a}$. The process uses the user's historical error mechanisms and their context information to modify local characters of the query, constructing paired queries for evaluating the impact of writing errors on retrieval outcomes. For each personalized syntactic expression query, PQB ranks the alphabetic tokens of at least three characters by the user's historical character-level error rate and determines the candidate edit positions. At each selected

position, PQB first attempts to reuse a character transformation observed in the user's historical reviews. Transformation matching proceeds from the most specific context to the least specific, backing off from the full syntactic context through dependency and POS information, coarse-grained context, and token-level context. When no applicable historical transformation exists, PQB samples an error type from the frequency distribution of the user's overall character-level error mechanisms and generates the corresponding character modification. To control the perturbation scope, PQB attempts to modify at most three distinct token positions per query and constrains the edit distance and the query length change. To monitor the semantic effect of these edits, PQB constrains the semantic similarity between each personalized syntactic expression query and its writing-error counterpart: a generated writing-error variant is rejected when its semantic similarity to the original query falls below 0.9, where $\mathrm{SemanticSim}(q_{u,a}, \tilde{q}_{u,a})$ is computed with the same all-mpnet-base-v2 encoder [48]. The detailed per-domain construction statistics are reported in Appendix Table 14.

4 Retrieval Benchmark Evaluation

This section evaluates, through five research questions, how personalized syntactic expression differences and writing errors affect retrieval results, and validates the constructed components of PQB. RQ1 first examines whether personalized syntactic expressions from different users correspond to different retrieval outcomes under a fixed retrieval target. RQ2 evaluates the effect of personalized writing errors on retrieval hits and ranking performance, and RQ3 validates benchmark quality in terms of personalized characteristics, query quality, and evaluation agreement. RQ4 compares the adopted user-specific syntactic representation with alternative embedding- and distribution-based models on held-out user discrimination and expression coverage, and RQ5 examines whether the constructed writing-error profiles identify users better than non-personalized sampling baselines. By applying a unified paired evaluation protocol across different retrieval systems, this section examines whether the above retrieval-result changes are consistent across systems.

4.1 Experimental Setup

This study revolves around the following five research questions:

(1) **RQ1. Personalized Syntactic Expression Differences and Retrieval Robustness.** Under a fixed retrieval target, do different personalized syntactic expressions produce retrieval-result differences? Is this phenomenon stable across different retrieval systems?

(2) **RQ2. Personalized Writing Errors and Retrieval Robustness.** How do personalized writing errors affect retrieval hits and ranking performance? Is this effect stable across different retrieval systems?

(3) **RQ3. Benchmark Quality Validation.** Do the queries generated by PQB exhibit good linguistic quality and identifiable personalized syntactic characteristics? Are the generated writing errors consistent with users' historical error behavior, and is the agreement between automatic and human evaluation reliable?

(4) **RQ4. User-Specific Syntactic Modeling Comparison.** Compared with other user-specific syntactic representations, does the adopted rank-one-plus-diagonal Gaussian model users' syntactic expressions more effectively and achieve better results in user attribution and user discrimination?

(5) **RQ5. Personalized Writing-Error Profile Evaluation.** Compared with global, uniform, character-random, and wrong-user sampling, does the proposed personalized writing-error mechanism sampling capture users' specific writing-error patterns more effectively and achieve better user attribution performance?

4.1.1 Retrieval Systems and Evaluation Protocol. To evaluate the retrieval-result changes corresponding to different user query expressions, we adopt five first-stage retrieval models with different representation and matching mechanisms: BM25 [49], SPLADE [19], GTE-base [33], ColBERTv2 [51], and BGE-M3 Hybrid [13], together with two LLM-based reranking models: Llama-3.1-8B [39] and Qwen3-Reranker-8B [63]. These models cover lexical retrieval, learned sparse retrieval, dense bi-encoder, late interaction, hybrid retrieval, and LLM-based reranking. In particular, the two LLM-based ranking models allow us to examine whether personalized syntactic expression variations and writing-error variations affect retrieval outcomes consistently across general-purpose and specialized LLM ranking models. Detailed model and parameter configurations are provided in Appendix Table 15.

4.1.2 Evaluation Metrics. We measure retrieval-result differences corresponding to query changes from two aspects: the hit status and the reciprocal rank of the target product. Both types of metrics are computed for each retrieval model, and smaller outcome variation indicates greater robustness under the corresponding metric; the reported differences are therefore read as the robustness behavior of individual models. For retrieval system m, let $r_m(q)$ be the retrieval rank of the target product for query q, with $r_m(q) = \infty$ when the target product is not retrieved. We first define the top-K hit indicator:

$$H_{m,K}(q) = \mathbb{I}[1 \le r_m(q) \le K]. \tag{17}$$

Here, $H_{m,K}(q) = 1$ indicates that the target product appears in the top-K retrieval results, and 0 otherwise.

(1) **Flip@K.** This metric measures whether two different queries produce different top-K hit statuses under the same retrieval target, considering only query pairs in which at least one query retrieves the target product. For the N query pairs in the evaluation set, it is defined as:

$$\mathrm{Flip}_m@K = \frac{\sum_{i=1}^{N} \mathbb{I}\left[H_{m,K}(q_{i,1}) \neq H_{m,K}(q_{i,2})\right]}{\sum_{i=1}^{N} \mathbb{I}\left[H_{m,K}(q_{i,1}) \vee H_{m,K}(q_{i,2})\right]}. \tag{18}$$

Here, $q_{i,1}$ and $q_{i,2}$ denote the two queries corresponding to the same target product. Query pairs in which neither query retrieves the target product within the top-K are excluded from the denominator. A higher Flip@K indicates a higher proportion of different target-product hit statuses among query pairs with at least one successful retrieval. When the denominator is zero, Flip@K is reported as N/A. We use $K \in \{1, 5, 10, 20\}$ to evaluate retrieval-result differences at different retrieval depths.

(2) **Mean RR_{std}.** To measure the difference in target-product rank between two queries, we first define the reciprocal rank within the top 200 retrieval results:

$$RR_{m,200}(q) = \begin{cases} 1/r_m(q), & 1 \le r_m(q) \le 200, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

We then compute the sample standard deviation $\mathrm{Std}_{sample}(RR_1, RR_2) = \frac{|RR_1 - R_2|}{\sqrt{2}}$ of the reciprocal ranks of the two queries for each query pair and average it over all query pairs:

$$\text{Mean RR}_{\mathrm{std},m} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{Std}_{sample}\left(RR_{m,200}(q_{i,1}), RR_{m,200}(q_{i,2})\right). \tag{20}$$

A higher value indicates a larger reciprocal-rank difference for the same target product under the two queries. Unlike Flip@K, which focuses on hit status at a given retrieval depth, Mean RR_{std} further measures the rank change of the target product.

(3) **Deg@K.** To measure the effect of writing errors on retrieval performance, let q_i be the personalized syntactic expression query and $\tilde{q}_i$ its corresponding personalized writing-error query. Deg@K is defined as the net hit-rate degradation between the two query conditions:

$$\mathrm{Deg}_m@K = \frac{1}{N}\sum_{i=1}^{N}\left[H_{m,K}(q_i) - H_{m,K}(\tilde{q}_i)\right]. \tag{21}$$

When Deg@K is positive, writing errors reduce the overall top-K hit rate of the target product; when it is negative, the overall hit rate improves. Because this metric simultaneously contains hit-to-miss and miss-to-hit changes, it reflects the net retrieval-performance change rather than a one-directional hit loss.

(4) **MRR Degradation.** In addition to hit-rate change, we further use the mean reciprocal rank to measure the rank degradation corresponding to writing errors. Let $RR_m(q)$ be the reciprocal rank of the target product within the effective ranking range of the corresponding retrieval configuration, taking zero on a miss. The MRR change between the personalized syntactic expression query and the personalized writing-error query is defined as:

$$\Delta MRR_m = \frac{1}{N}\sum_{i=1}^{N}\left[RR_m(\tilde{q}_i) - RR_m(q_i)\right]. \tag{22}$$

To make a positive value correspond to retrieval-performance degradation, we further define:

$$\mathrm{MRR}_{\mathrm{deg},m} = -\Delta MRR_m. \tag{23}$$

Thus, $\mathrm{MRR}_{\mathrm{deg},m} > 0$ indicates that writing errors lower the mean reciprocal rank of the target product, and a larger value indicates a more pronounced net rank degradation. For retrieval configurations that use a limited candidate set, the reciprocal rank is computed over the corresponding effective ranking range.

4.2 Personalized Syntactic Expression Differences and Retrieval Robustness

This section answers RQ1: under a fixed retrieval target, do different personalized syntactic expressions correspond to observable differences in retrieval robustness, and is this phenomenon persistent across different retrieval systems? To this end, we use personalized syntactic expression query pairs under the same target product to evaluate changes in target-product hit status and reciprocal rank across different retrieval configurations, examining the retrieval-result differences corresponding to personalized syntactic expression differences.

4.2.1 Personalized Syntactic Expression Evaluation Setup. This experiment uses the personalized syntactic expression queries constructed in Section 3.3. Each target product corresponds to the representative queries of two distinct users. Following the Gaussian query selection procedure of Section 3.3, the two queries come respectively from the exclusive parts of the corresponding users' Gaussian acceptance regions and are selected according to the user-specific syntactic distributions. The two queries share the same target product and product-attribute constraints, but reflect the personalized syntactic expression characteristics of different users. Both queries come from the candidate pool that has passed the strict lexical and semantic validation of Section 3.3.1: they share the same target product and five identical core retrieval terms, with each core term appearing exactly once. The differences between the queries mainly lie in function words, modification relations, and syntactic organization, and are personalized through different users' Gaussian syntactic distributions.

4.2.2 Results and Analysis. Table 2 presents the retrieval-result differences of the seven retrieval configurations under personalized syntactic expression variation. We analyze them from three aspects: the target-product hit disagreement associated with controlled syntactic variation, the sensitivity of different retrieval architectures to expression variation, and the behavior across retrieval depths and ranking variation.

Table 2. Retrieval differences under personalized syntactic expression variation across three product domains and their all-domain average.

	Baby Products					Musical Instruments				
Model	Flip@1	Flip@5	Flip@10	Flip@20	Mean RR_{std}	Flip@1	Flip@5	Flip@10	Flip@20	Mean RR_{std}
BM25	44.44%	27.03%	31.68%	23.28%	6.75%	39.34%	29.27%	27.81%	22.91%	5.48%
SPLADE	32.35%	13.51%	12.09%	14.95%	4.03%	24.59%	19.67%	13.99%	8.13%	3.27%
GTE-base	35.00%	35.90%	25.58%	38.60%	3.78%	44.44%	31.82%	19.61%	19.84%	4.58%
BGE-M3 Hybrid	41.94%	23.94%	18.89%	21.82%	5.42%	35.62%	17.54%	19.86%	19.05%	4.48%
ColBERTv2	57.14%	42.22%	36.67%	32.86%	5.46%	52.50%	43.84%	39.13%	37.93%	5.21%
Llama-3.1-8B	45.24%	30.00%	26.92%	20.69%	7.78%	36.84%	31.85%	28.92%	20.11%	6.54%
Qwen3-Reranker-8B	42.50%	30.00%	31.48%	22.69%	8.01%	41.86%	24.82%	20.48%	19.78%	7.08%
	Video Games					**All-domain-average**				
Model	Flip@1	Flip@5	Flip@10	Flip@20	Mean RR_{std}	Flip@1	Flip@5	Flip@10	Flip@20	Mean RR_{std}
BM25	41.22%	28.37%	23.77%	21.45%	3.92%	41.67%	28.22%	27.76%	22.54%	5.38%
SPLADE	22.76%	11.66%	11.28%	10.29%	2.00%	26.57%	14.95%	12.45%	11.12%	3.10%
GTE-base	41.67%	38.89%	27.19%	24.44%	2.56%	40.37%	35.53%	24.13%	27.62%	3.64%
BGE-M3 Hybrid	31.41%	22.48%	18.79%	16.05%	3.03%	36.32%	21.32%	19.18%	18.97%	4.31%
ColBERTv2	46.94%	35.36%	33.48%	32.59%	3.25%	52.19%	40.47%	36.43%	34.46%	4.64%
Llama-3.1-8B	44.05%	33.02%	27.44%	23.02%	4.68%	42.04%	31.62%	27.76%	21.27%	6.33%
Qwen3-Reranker-8B	45.40%	31.40%	22.63%	22.33%	4.81%	43.25%	28.74%	24.86%	21.60%	6.63%

In each column, the lowest and highest values are highlighted with red and blue backgrounds, respectively. All-domain-average is the equal-weight average of the three domain-specific values.

(1) **Syntactic variation under identical retrieval content still corresponds to clear hit-status disagreement.** When the target product and the five core retrieval terms are kept identical, all seven retrieval configurations exhibit nonzero Flip@K, showing that variations in a query's syntactic organization alone may still be associated with different target-product hit outcomes. The differences between these queries lie mainly in function words, modification relations, and syntactic structure rather than in the core retrieval content. The phenomenon indicates that how a retrieval system handles user expression depends not only on which content words the query contains, but also on how those words are organized into a complete query. It appears across architectures, which suggests that personalized syntactic expression differences are a problem faced by many retrieval mechanisms alike, and it also illustrates the value of PQB for evaluating expression sensitivity under a fixed retrieval need.

(2) **Different retrieval models exhibit differing robustness to personalized syntactic expression differences.** Direct lexical matching shows a higher rate of hit-status disagreement than learned sparse retrieval, late-interaction scoring shows the highest rate of hit-status disagreement among the first-stage configurations, and dense bi-encoder and hybrid representations fall between these extremes. Both LLM-based ranking models reach a disagreement level comparable to the first-stage systems, and the specialized reranker does not consistently exhibit less hit-status disagreement than the general-purpose instruction-tuned model. This ordering does not follow overall retrieval effectiveness: the architectures with the strongest semantic capacity are not the ones with the smallest expression-induced disagreement. The pattern therefore points to how each

architecture converts surface form into a matching decision, rather than to how well it retrieves in absolute terms. Since Flip@K is computed over the query pairs with at least one successful retrieval for each system, these differences describe system-specific conditional variation rather than a direct ranking of overall retrieval effectiveness.

(3) **Rank variations corresponding to personalized syntactic expressions manifest as non-monotonic hit differences across retrieval depths.** Flip@K does not decrease monotonically as the cutoff grows: for some configurations the disagreement rises from $K = 1$ to $K = 5$ before falling again, while for others it falls first and rises again at larger cutoffs. This non-monotonicity follows from the composition of the eligible pair set at each depth, because a wider cutoff admits pairs in which one query succeeds only at that depth and removes pairs that disagree at one depth but agree at another, so both the number of disagreements and the conditional denominator change together. Mean RR_{std} is nonzero for every configuration, which shows that the expression-induced change is not confined to the membership of a particular top-K list but also shifts how far down the target product is ranked. A single cutoff therefore cannot characterize the robustness of a model to expression variation, and depth-dependent behavior should be read as a change in the composition of disagreement rather than as a monotone gain or loss.

4.3 Personalized Writing Errors and Retrieval Robustness

This section answers RQ2: how do personalized writing errors affect retrieval hits and ranking performance, and how robust are retrieval systems to these errors? RQ1 mainly examined retrieval-result differences corresponding to different syntactic expressions. This section further focuses on writing errors in user queries. By comparing personalized syntactic expression queries with their corresponding personalized writing-error queries, we evaluate the retrieval-performance changes corresponding to writing errors and analyze the manifestation of this phenomenon across different product domains, retrieval systems, and retrieval depths.

4.3.1 Personalized Writing-Error Evaluation Setup. This experiment uses paired queries over the same target products and retrieval configurations as RQ1. For each personalized syntactic expression query, a corresponding personalized writing-error query is generated based on the user writing-error profile constructed in Section 3.4. The evaluation retains only valid query pairs where the emitted text is nonempty and distinct from the personalized syntactic expression query. Queries that do not generate a valid personalized writing-error query are excluded from the paired statistics of this experiment. Each query pair formed by a personalized syntactic expression query and its corresponding personalized writing-error query shares the same target product and executes retrieval separately under identical retrieval systems. For configurations involving candidate reranking, the personalized syntactic expression query and the personalized writing-error query of each pair separately perform candidate retrieval and reranking to evaluate the behavior of the full retrieval pipeline under writing-error conditions.

4.3.2 Results and Analysis. Table 3 presents the retrieval-performance changes of the seven retrieval configurations under personalized writing errors. We analyze them from three aspects: the directional performance degradation brought by writing errors, the error sensitivity of different retrieval architectures, and the behavior across retrieval depths and target-product ranking losses.

Table 3. Retrieval performance degradation under personalized writing errors across three product domains and their all-domain average.

	Baby Products					Musical Instruments				
Model	Deg@1	Deg@5	Deg@10	Deg@20	MRR_{deg}	Deg@1	Deg@5	Deg@10	Deg@20	MRR_{deg}
BM25	6.38%	10.74%	15.44%	18.46%	8.92%	4.01%	7.16%	8.73%	10.99%	5.64%
SPLADE	4.70%	5.70%	7.38%	6.38%	5.10%	3.32%	5.58%	5.93%	5.76%	4.88%
GTE-base	5.03%	6.38%	4.36%	3.36%	5.20%	4.71%	6.11%	7.68%	9.25%	5.92%
ColBERTv2	2.68%	6.38%	9.06%	7.38%	4.25%	2.97%	4.71%	4.89%	8.73%	3.67%
BGE-M3 Hybrid	3.36%	6.71%	6.04%	6.38%	4.17%	2.27%	4.71%	4.36%	4.36%	3.03%
Llama-3.1-8B	5.37%	12.08%	13.09%	16.44%	8.17%	3.49%	6.81%	9.95%	12.22%	5.56%
Qwen3-Reranker-8B	7.38%	12.08%	11.74%	16.11%	9.89%	3.66%	7.33%	8.90%	10.82%	5.54%
	Video Games					**All-domain-average**				
Model	Deg@1	Deg@5	Deg@10	Deg@20	MRR_{deg}	Deg@1	Deg@5	Deg@10	Deg@20	MRR_{deg}
BM25	2.16%	4.82%	6.03%	7.53%	3.55%	4.18%	7.57%	10.07%	12.33%	6.04%
SPLADE	2.46%	4.82%	4.82%	5.68%	3.50%	3.49%	5.37%	6.04%	5.94%	4.49%
GTE-base	1.66%	4.62%	5.47%	6.03%	2.87%	3.80%	5.70%	5.84%	6.21%	4.66%
ColBERTv2	2.76%	4.52%	4.97%	5.22%	3.45%	2.80%	5.20%	6.31%	7.11%	3.79%
BGE-M3 Hybrid	1.91%	3.21%	3.37%	4.02%	2.53%	2.51%	4.88%	4.59%	4.92%	3.24%
Llama-3.1-8B	3.16%	4.92%	5.98%	7.18%	4.19%	4.01%	7.94%	9.67%	11.95%	5.97%
Qwen3-Reranker-8B	3.47%	5.42%	6.53%	7.23%	4.48%	4.84%	8.28%	9.06%	11.39%	6.64%

In each column, the lowest and highest values are highlighted with red and blue backgrounds, respectively. All-domain-average is the equal-weight average of the three domains.

(1) **Personalized writing errors are associated with a consistent net degradation in retrieval performance.** Unlike the hit-status differences between personalized syntactic expressions, the writing-error experiment directly compares the retrieval performance of an original query with that of its error variant. The overall results of all seven retrieval configurations show positive Deg@K and MRR Degradation, indicating that character-level expression changes affect not only whether the target product can be retrieved but also correspond to a decline in overall retrieval effectiveness. Although the original query and its error variant share the same retrieval target and retain high semantic similarity, local word-form changes still weaken the matching signal between the query and the product documents. This shows that the effect of personalized writing errors is not only that retrieval outcomes change, but that it further manifests as a directional loss in hit-rate and ranking performance.

(2) **Different retrieval models exhibit differing robustness to personalized writing errors.** BM25 shows relatively pronounced overall hit-rate degradation, reflecting the sensitivity of direct term matching to character-level errors. In contrast, SPLADE's learned sparse representations and BGE-M3 Hybrid's multiple matching signals exhibit smaller performance losses, with BGE-M3 Hybrid achieving the lowest degradation in the cross-domain average results. GTE-base and ColBERTv2 are also affected, indicating that overall semantic encoding and token-level interactions cannot fully eliminate the shifts in representations and matching caused by writing errors. For LLMs, both the general instruction model Llama-3.1-8B and the reranking-specialized Qwen3-Reranker-8B exhibit clear performance degradation. The latter has a more explicit relevance-ranking training objective, yet still shows substantial losses on reciprocal-rank metrics, indicating that specialized training for ranking tasks does not necessarily reduce performance degradation induced by personalized writing errors.

(3) **The performance losses caused by personalized writing errors are observed at different retrieval depths and in target-product rankings.** Across the three product domains, the average Deg@K generally increases as retrieval depth expands, indicating that net hit-rate losses from writing errors are not confined

to top-ranked positions. As the retrieval scope expands, target products affected by errors are not sufficiently recovered. Meanwhile, all retrieval configurations exhibit positive MRR Degradation, indicating that writing errors not only reduce the overall probability of target products entering the top-K, but also weaken their reciprocal-rank performance. Together, both metrics demonstrate that retrieval performance degradation corresponding to personalized writing errors involves both retrieval coverage and ranking positions of target products.

4.4 Benchmark Quality Validation

This section answers RQ3 by assessing the benchmark quality of PQB along four dimensions: Personalized Syntactic Attribution, Personalized Writing-Error Consistency, Query Quality, and Evaluation Agreement.

4.4.1 Benchmark Quality Evaluation Setup. For benchmark quality evaluation, we used DeepSeek-V4-Pro [16] for automatic assessment and selected 360 queries for human annotation through stratified sampling by domain, ASIN, query condition, and error type. We recruited five human annotators from Computer Science and Information Technology programs at Monash University, all possessing high English proficiency and relevant knowledge backgrounds in NLP, linguistics, or Information Retrieval.

Before the formal evaluation, all five annotators completed unified training and achieved at least 80% accuracy on a 20-sample qualification check; all five then participated in the formal evaluation. In the formal stage, each query is independently annotated by at least two annotators under blind conditions; annotators are not informed of the query-generation condition, the personalized selection procedure, or DeepSeek-V4-Pro judgments. Instances with annotator disagreement in classification tasks are adjudicated by a third annotator. Human–Human agreement is computed on the independent annotations prior to adjudication, and adjudicated labels are used to construct the final human reference. DeepSeek-V4-Pro and the human annotators use the same task definitions, evaluation criteria, and label space.

(1) **Personalized Syntactic Attribution.** The evaluator first views the anonymized historical expressions of the target user, then compares two queries that share the same product semantics and core attributes but differ in syntactic realization. One comes from personalized syntax-profile selection; the other is a matched control query. The order of the two queries is randomized, and the evaluator decides which one better matches the target user's historical expression style, with *No clear preference* also allowed. This paper reports decisive attribution accuracy and the no-preference rate separately. Let N_{pers}, N_{ctrl}, and N_{tie} denote the numbers of judgments selecting the personalized syntactic expression query, the control query, and *No clear preference*, respectively. Then

$$Acc_{\text{decisive}} = \frac{N_{\text{pers}}}{N_{\text{pers}} + N_{\text{ctrl}}}, \tag{24}$$

$$TieRate = \frac{N_{\text{tie}}}{N_{\text{pers}} + N_{\text{ctrl}} + N_{\text{tie}}}. \tag{25}$$

The random-choice baseline of Acc_{decisive} is 50%.

(2) **Personalized Writing-Error Consistency.** The evaluator views the anonymized historical writing-error evidence of the target user and judges how consistent the error pattern in the personalized writing-error query is with that user's historical writing-error behavior. Judgments use three ordered labels: *Pass*, *Minor Issue*, and *Fail*. *Pass* indicates that the generated error pattern is clearly consistent with the historical evidence; *Minor*

Issue indicates partial consistency but with weak or incomplete evidence; *Fail* indicates that the generated pattern lacks sufficient historical support.

(3) **Query Quality.** Naturalness and Syntactic Quality are rated on a unified 1–5 Likert scale; the two dimensions assess the linguistic naturalness of generated queries and the coherence of their syntactic organization, respectively.

(4) **Evaluation Agreement.** Human–Human reliability is computed using Krippendorff's α. Ordinal α is used for Naturalness, Syntactic Quality, and the three-level Personalized Writing-Error Consistency of *Pass* / *Minor Issue* / *Fail*; nominal α is used for Personalized Syntactic Attribution. LLM–Human agreement compares DeepSeek-V4-Pro judgments against the final human reference, using weighted Cohen's κ for ordinal tasks and Cohen's κ for Personalized Syntactic Attribution.

4.4.2 Results and Analysis. Table 4 and Table 5 present the benchmark quality evaluation results and annotation agreement of PQB.

Table 4. Benchmark quality validation results.

Dimension	Metric	LLM	Human
Query Quality	Naturalness (1–5)	4.38	4.21
Query Quality	Syntactic Quality (1–5)	4.31	4.16
Personalized Syntactic Attribution	Decisive Accuracy	70.1%	66.3%
Personalized Syntactic Attribution	No-preference Rate	5.2%	8.7%
Personalized Writing-Error Consistency	Strict Pass Rate	88.1%	84.6%
Personalized Writing-Error Consistency	Acceptable Rate	96.3%	94.8%

Table 5. Human–Human and LLM–Human agreement for benchmark quality evaluation.

Dimension	Human–Human α	LLM–Human κ
Naturalness	0.72	0.67
Syntactic Quality	0.69	0.64
Personalized Syntactic Attribution	0.74	0.68
Personalized Writing-Error Consistency	0.77	0.71

(1) **Personalized syntactic expression queries can exhibit identifiable user-specific expression features.** Both LLM and human evaluations of syntactic attribution accuracy exceed the random-selection baseline, with low no-preference rates. This indicates that under consistent target products and core attributes, generated queries still retain syntactic features consistent with users' historical expressions. Evaluators can distinguish queries from different users based on historical expressions, indicating that PQB's personalized syntactic modeling creates differences in query form while retaining identifiable features of users' expression styles.

(2) **Personalized writing-error queries can reflect users' historical writing-error habits.** Both LLM and human evaluations achieve high strict pass rates and acceptable rates, indicating that generated writing errors maintain high consistency with error mechanisms in users' histories. Unlike a common random error-injection scheme applied to all users, PQB constructs personalized writing errors based on users' historical error patterns,

so the generated queries reflect not only surface-level character changes but also user-specific preferences for error mechanisms. This provides data-quality support for analyzing changes in retrieval performance under personalized writing errors and for the robustness analyses reported above.

(3) **PQB maintains high language quality while preserving personalized expression features, and obtains consistent quality evaluation results.** Both naturalness and syntactic quality receive high scores in LLM and human evaluations, indicating that personalized expression modeling does not come at the cost of markedly reduced query language quality. Meanwhile, Human–Human and LLM–Human agreement reach good levels across all dimensions, indicating that different human annotators make similar judgments about query quality and personalized features, and that automatic evaluation maintains overall trends consistent with human references. This further supports the reliability of PQB's quality evaluation results.

4.5 User-Specific Syntactic Modeling Comparison

This section answers RQ4: how does the adopted rank-one-plus-diagonal Gaussian compare with alternative user-specific syntactic representations in discriminating users from their held-out expressions?

To evaluate the effectiveness of user-specific syntactic modeling, we compare the adopted rank-one-plus-diagonal Gaussian [56] against six alternative representations: a single embedding scored by cosine similarity [48], a shared isotropic Gaussian [60], a user-specific diagonal Gaussian [17, 28], a von Mises-Fisher distribution [43], a rank-two-plus-diagonal Gaussian [56], and a k-nearest-neighbor profile matcher [59]. All models are evaluated in three product domains using the same 16-dimensional syntax encoder and the same user histories. Three complementary measures are used:

- **Balanced Attribution.** The proportion of held-out expressions that are correctly attributed to their true user when the candidate sets of the two users are weighted equally, so that a user with many expressions does not dominate the average.
- **User AUROC.** The area under the receiver operating characteristic curve of the user-discrimination score, i.e., the probability that a held-out expression of one user is scored above a held-out expression of another user. It summarizes the separation between users without depending on any decision threshold.
- **Full-profile Attribution.** The attribution accuracy obtained when the full user profile is used as the reference budget, i.e., the full-budget endpoint of the profile-budget analysis, which shows how well the representation identifies users once all available history is used.

The table reports unweighted averages across the three product domains.

Table 6. Comparison of user-specific syntactic modeling methods (unweighted averages across the three product domains).

Model	Balanced Attr. (%)	AUROC	Full-profile Attr. (%)
Single Embedding + Cosine	64.10%	0.6051	63.60%
Shared Isotropic Gaussian	64.01%	0.6061	63.53%
User Diagonal Gaussian	62.76%	0.5996	62.49%
von Mises-Fisher (vMF)	59.61%	0.5731	59.52%
Rank-Two + Diagonal Gaussian	58.90%	0.5645	58.88%
k-Nearest Neighbor	56.22%	0.5556	56.31%
Rank-One + Diagonal Gaussian (Ours)	**65.55%**	**0.6265**	**65.52%**

The rank-one-plus-diagonal Gaussian achieves the best result on all three measures. In the user-attribution task, single embedding and the shared isotropic Gaussian perform comparably, and the discrimination ability of user diagonal, vMF, rank-two, and k-nearest-neighbor decreases in that order; AUROC and full-profile attribution accuracy show a similar trend, with the proposed method consistently ahead, indicating that it separates users' historical syntactic expressions more effectively. Overall, the method shows a consistent advantage in discriminating users from their held-out syntactic expressions. These findings highlight the role of user-specific syntactic distribution modeling in PQB's personalized query selection, while the writing-error profiles examined in RQ5 capture users' preferences for error mechanisms observed in their historical character edits, providing the basis for constructing personalized writing-error variants.

4.6 Personalized Writing-Error Profile Evaluation

This section answers RQ5: do the constructed writing-error profiles capture user-specific error mechanisms? To assess whether the constructed writing-error profiles capture user-specific characteristics, we conduct an experiment using sampled error-mechanism events. We compare personalized sampling against global, uniform, character-random, and wrong-user profile baselines. User attribution is evaluated using independently estimated reference error profiles. The table reports three measures:

- **Top-1 Accuracy.** The proportion of instances in which the correct user is ranked first among the candidate users, based on the sampled writing-error evidence.
- **MRR.** The average reciprocal rank of the correct user in that candidate ranking, which therefore reflects both whether the correct user is retrieved at all and how highly the user is ranked.
- **Single-query Top-1.** The proportion of instances in which the correct user is still identified when only the evidence provided by a single query is available, which tests the usability of the profiles when evidence is sparse.

Table 7. User attribution performance of different writing-error mechanism sampling strategies.

Method	Top-1 Accuracy ↑	MRR ↑	Single-query Top-1 ↑
Global Sampling	5.76%	0.1948	6.53%
Uniform Sampling	4.68%	0.1874	4.52%
Character-random Sampling	4.32%	0.1781	4.52%
Wrong-user Profile Sampling	6.47%	0.2003	4.02%
Personalized Sampling (Ours)	**9.35%**	**0.2428**	**7.29%**

Personalized sampling achieves the highest performance across all three metrics. It outperforms global sampling on Top-1 accuracy and MRR, and also performs better than uniform and character-random sampling, indicating that user-specific error-mechanism preferences provide additional discriminative information. Compared with the wrong-user profile baseline, personalized sampling achieves higher Top-1 accuracy and MRR. These findings suggest that the constructed profiles retain identifiable user-specific writing-error patterns, providing empirical support for their use in subsequent personalized query generation.

4.7 Summary of Overall Experimental Results

Taken together, the experimental results of RQ1 and RQ2 show that both personalized syntactic expression differences and personalized writing errors affect retrieval results, but the ways in which they affect retrieval results differ. RQ1 shows that even when the target product and five core retrieval terms remain unchanged, personalized syntactic expression differences among users still lead to noticeable differences in retrieval results. These differences are observed across different retrieval systems and retrieval depths, and the differences in retrieval hits do not decrease monotonically as retrieval depth increases. RQ2 further shows that personalized writing errors lead to retrieval performance degradation. Even when the original queries and their personalized writing-error variants share the same retrieval target and maintain high semantic similarity, personalized writing errors still reduce retrieval hit rates and ranking performance, with performance degradation persisting across different retrieval depths. In addition, both experiments show that retrieval models exhibit different levels of robustness. Their relative robustness differs between the personalized syntactic expression and personalized writing-error evaluations.

RQ3, RQ4, and RQ5 further validate the quality of PQB and the effectiveness of its personalized modeling methods. The LLM and human evaluations in RQ3 show that the generated queries preserve identifiable user-specific expression characteristics, while personalized writing-error queries maintain high consistency with personalized writing-error mechanisms observed in users' historical texts. The generated queries also exhibit high naturalness and linguistic quality, with good agreement observed in both Human–Human and LLM–Human evaluations. RQ4 shows that the rank-one-plus-diagonal Gaussian achieves the best results across all three attribution and discrimination metrics, demonstrating its effectiveness in personalized syntactic modeling. RQ5 shows that personalized sampling outperforms other sampling methods across all three user attribution metrics, validating the effectiveness of user-specific writing-error profiles. Taken together, the results across the five research questions show that PQB can construct personalized queries from users' historical expressions and evaluate the effects of personalized syntactic expression differences and personalized writing errors on retrieval outcomes across different retrieval models.

5 Conclusion

PersonalQuery-Benchmark (PQB) is introduced to study how robust retrieval systems are to personalized query expressions. PQB builds personalized syntactic expression models and writing-error profiles from user review histories: for each query pair, PQB holds the target product, five core retrieval terms, and their occurrence frequencies constant, generates personalized syntactic queries together with their corresponding writing-error queries, and evaluates them across three product domains and seven retrieval configurations. The experiments show that personalized syntactic expressions from different users are associated with changes in the target product's hit status and ranking, while personalized writing errors lead to retrieval performance degradation across the evaluated models. These phenomena are observable across multiple retrieval models, indicating that personalized query expressions are an important factor determining how robust a retrieval model is. Overall, PQB provides a unified benchmark and evaluation framework for studying the robustness of retrieval systems to personalized query expressions along two dimensions: personalized syntactic expression and personalized writing errors. Future work will broaden the coverage of user expression differences and incorporate richer user histories and real query data for further validation.

References

[1] Kenya Abe, Kunihiro Takeoka, Makoto P. Kato, and Masafumi Oyamada. 2025. LLM-based Query Expansion Fails for Unfamiliar and Ambiguous Queries. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3035–3039. doi:10.

1145/3726302.3730222

[2] Nuha Abu Onq, Mark Sanderson, and Falk Scholer. 2025. Classifying Term Variants in Query Formulation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2395–2406. doi:10.1145/3726302.3729924

[3] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W. Bruce Croft. 2017. Learning a Hierarchical Embedding Model for Personalized Product Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 645–654. doi:10.1145/3077136.3080813

[4] Marwah Alaofi, Nicola Ferro, Paul Thomas, Falk Scholer, and Mark Sanderson. 2025. Demographically-Inspired Query Variants Using an LLM. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR 2025)*. 390–400. doi:10.1145/3731120.3744608

[5] Marwah Alaofi, Luke Gallagher, Mark Sanderson, Falk Scholer, and Paul Thomas. 2023. Can Generative LLMs Create Query Variants for Test Collections? An Exploratory Study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1869–1873. doi:10.1145/3539618.3591960

[6] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*. 725–728. doi:10.1145/2911451.2914671

[7] Elias Bassani and Gabriella Pasi. 2021. Semantic Query Labeling Through Synthetic Query Generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2278–2282. doi:10.1145/3404835.3463071

[8] Elias Bassani, Nicola Tonellotto, and Gabriella Pasi. 2024. Personalized Query Expansion with Contextual Word Embeddings. *ACM Transactions on Information Systems* 42, 2 (2024), 1–35. doi:10.1145/3624988

[9] Rodger Benham, Joel Mackenzie, Alistair Moffat, and J. Shane Culpepper. 2019. Boosting Search Performance Using Query Variations. *ACM Transactions on Information Systems* 37, 4 (2019), 1–25. doi:10.1145/3345001

[10] Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 793–805. doi:10.18653/v1/P17-1074

[11] David Carmel and Elad Yom-Tov. 2010. Estimating the Query Difficulty for Information Retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 911. doi:10.1145/1835449.1835683

[12] Andreas Chari, Sean MacAvaney, and Iadh Ounis. 2023. On the Effects of Regional Spelling Conventions in Retrieval Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2220–2224. doi:10.1145/3539618.3592030

[13] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*. 2318–2335. doi:10.18653/v1/2024.findings-acl.137

[14] Lingzhen Chen, Carlo Strapparava, and Vivi Nastase. 2017. Improving Native Language Identification by Using Spelling Errors. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*. 542–546. doi:10.18653/v1/P17-2086

[15] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A 'Pointwise-Query, Listwise-Document' based Query Performance Prediction Approach. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2148–2153. doi:10.1145/3477495.3531821

[16] DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. *arXiv preprint arXiv:2606.19348* (2026).

[17] Xue Dong, Xuemeng Song, Na Zheng, Yinwei Wei, and Zhongzhou Zhao. 2023. Dual Preference Distribution Learning for Item Recommendation. *ACM Transactions on Information Systems* 41, 3 (2023), 1–22. doi:10.1145/3565798

[18] Simone Filice, Guy Horowitz, David Carmel, Zohar Karnin, Liane Lewin-Eytan, and Yoelle Maarek. 2025. Generating Q&A Benchmarks for RAG Evaluation in Enterprise Settings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track) (ACL 2025)*. 469–484. doi:10.18653/v1/2025.acl-industry.33

[19] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. *arXiv preprint arXiv:2109.10086* (2021).

[20] Sahin Cem Geyik, Vijay Dialani, Meng Meng, and Ryan Smith. 2018. In-Session Personalization for Talent Search. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2107–2115. doi:10.1145/3269206.3272012

[21] Gregory Goren. 2019. Ranking Robustness In Adversarial Retrieval Settings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. 1446. doi:10.1145/3331184.3331414

[22] Tim Hagen, Harrisen Scells, and Martin Potthast. 2024. Revisiting Query Variation Robustness of Transformer Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 4283–4296. doi:10.18653/v1/2024.findings-emnlp.248

[23] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024). doi:10.48550/arXiv.2403.03952

[24] Zhiting Hu and Li Erran Li. 2021. A Causal Lens for Controllable Text Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 24941–24955. https://arxiv.org/abs/2201.09119

[25] Wei Huang, Keping Bi, Yinqiong Cai, Wei Chen, Jiafeng Guo, and Xueqi Cheng. 2026. How Do LLM-Generated Texts Impact Term-Based Retrieval Models?. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (WSDM 2026)*. 238–248. doi:10.1145/3773966.3777988

[26] Samuel Huston and W. Bruce Croft. 2010. Evaluating Verbose Query Processing Techniques. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 291–298. doi:10.1145/1835449.1835499

[27] Yuta Imasaka and Hideo Joho. 2024. Effect of LLM's Personality Traits on Query Generation. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP 2024)*. 249–258. doi:10.1145/3673791.3698433

[28] Junyang Jiang, Deqing Yang, Yanghua Xiao, and Chenlu Shen. 2019. Convolutional Gaussian Embeddings for Personalized Recommendation with Uncertainty. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 2642–2648. doi:10.24963/ijcai.2019/367

[29] Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P. de Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing Personal LAPS: LLM-Augmented Dialogue Construction for Personalized Multi-Session Conversational Search. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024)*. 796–806. doi:10.1145/3626772.3657815

[30] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2017. On Application of Learning to Rank for E-Commerce Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 475–484. doi:10.1145/3077136.3080838

[31] Roman Kern. 2013. Grammar Checker Features for Author Identification and Author Profiling. In *Proceedings of the Working Notes for the CLEF 2013 Evaluation Labs and Workshop*. 1–5. https://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-Kern2013.pdf

[32] Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an Author's Native Language by Mining a Text for Errors. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 624–628. doi:10.1145/1081870.1081947

[33] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv preprint arXiv:2308.03281* (2023).

[34] Christina Lioma and Roi Blanco. 2009. Part of Speech Based Term Weighting for Information Retrieval. In *Advances in Information Retrieval – Proceedings of the 31st European Conference on IR Research (ECIR 2009)*. 412–423. doi:10.1007/978-3-642-00958-7_37

[35] Chang Liu. 2008. Exploring and Measuring Dependency Trees for Information Retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*. 892. doi:10.1145/1390334.1390567

[36] Jiongnan Liu, Zhicheng Dou, Guoyu Tang, and Sulong Xu. 2023. JDsearch: A Personalized Product Search Dataset with Real Queries and Full Interactions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2945–2952. doi:10.1145/3539618.3591900

[37] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Topic-oriented Adversarial Attacks against Black-box Neural Ranking Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1700–1709. doi:10.1145/3539618.3591777

[38] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Multi-granular Adversarial Attacks against Black-box Neural Ranking Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1391–1400. doi:10.1145/3626772.3657704

[39] Llama Team, AI at Meta. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[40] Kim Luyckx and Walter Daelemans. 2008. Personae: a Corpus for Author and Personality Prediction from Text. In *Proceedings of the 6th International Language Resources and Evaluation Conference (LREC 2008)*. 2981–2986.

[41] Aritra Mandal, Ishita Khan, and Prathyusha Senthil Kumar. 2019. Query Rewriting using Automatic Synonym Extraction for E-commerce Search. In *Proceedings of the SIGIR 2019 Workshop on eCommerce (eCOM 2019), CEUR Workshop Proceedings Vol-2410*. http://ceur-ws.org/Vol-2410/paper20.pdf

[42] Josiane Mothe and Ludovic Tanguy. 2005. Linguistic Features to Predict Query Difficulty – A Case Study on Previous TREC Campaigns. In *Proceedings of the ACM SIGIR Workshop on Predicting Query Difficulty*. 7–10. https://shs.hal.science/halshs-00287692

[43] Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA.

[44] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. 1659–1666.

[45] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – Grammatical Error Correction: Tag, Not Rewrite. In *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. 163–170. doi:10.18653/v1/2020.bea-1.16

[46] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *Advances in Information Retrieval: 44th European Conference on IR Research (ECIR 2022) (Lecture Notes in Computer Science, Vol. 13185)*. 397–412. doi:10.1007/978-3-030-99736-6_27

[47] Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating Adversarial Misspellings with Robust Word Recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5582–5591. doi:10.18653/v1/P19-1561

[48] 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)*. 3982–3992. doi:10.18653/v1/D19-1410

[49] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. doi:10.1561/1500000019

[50] Sonal Sannigrahi, Thiago Fraga-Silva, Youssef Oualil, and Christophe Van Gysel. 2024. Synthetic Query Generation using Large Language Models for Virtual Assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2837–2841. doi:10.1145/3626772.3661355

[51] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. 3715–3734. doi:10.18653/v1/2022.naacl-main.272

[52] Georgios Sidiropoulos and Evangelos Kanoulas. 2022. Analysing the Robustness of Dual Encoders for Dense Retrieval Against Misspellings. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2132–2136. doi:10.1145/3477495.3531818

[53] Munirathnam Srikanth and Rohini Srihari. 2003. Exploiting Syntactic Structure of Queries in a Language Modeling Approach to IR. In *Proceedings of the 12th ACM International Conference on Information and Knowledge Management (CIKM 2003)*. 476–483. doi:10.1145/956863.956952

[54] Efstathios Stamatatos. 2009. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology* 60, 3 (2009), 538–556. doi:10.1002/asi.21001

[55] Kalaivani Sundararajan and Damon Woodard. 2018. What represents "style" in authorship attribution?. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*. 2814–2822.

[56] Michael E. Tipping and Christopher M. Bishop. 1999. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61, 3 (1999), 611–622.

[57] Rob van der Goot, Nikola Ljubešić, Ian Matroos, Malvina Nissim, and Barbara Plank. 2018. Bleaching Text: Abstract Features for Cross-lingual Gender Prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, 383–389. doi:10.18653/v1/P18-2061

[58] Hemanth Vemuri, Sheshansh Agrawal, Shivam Mittal, Deepak Saini, Akshay Soni, Abhinav V. Sambasivan, Wenhao Lu, Yajun Wang, Mehul Parsana, Purushottam Kar, and Manik Varma. 2023. Personalized Retrieval over Millions of Items. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1014–1022. doi:10.1145/3539618.3591749

[59] Koen Verstrepen and Bart Goethals. 2014. Unifying Nearest Neighbors Collaborative Filtering. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 177–184. doi:10.1145/2645710.2645731

[60] Luke Vilnis and Andrew McCallum. 2015. Word Representations via Gaussian Embedding. In *International Conference on Learning Representations*. https://arxiv.org/abs/1412.6623

[61] Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2023. PRADA: Practical Black-box Adversarial Attacks against Neural Ranking Models. *ACM Transactions on Information Systems* 41, 4 (2023), 1–27. doi:10.1145/3576923

[62] Xiaohui Xie, Jiaxin Mao, Yiqun Liu, and Maarten de Rijke. 2020. Modeling User Behavior for Vertical Search: Images, Apps and Products. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2440–2443. doi:10.1145/3397271.3401423

[63] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388* (2025).

[64] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115* (2024).

[65] Jayanth Yetukuri and Ishita Khan. 2025. Intent-Aware Neural Query Reformulation for Behavior-Aligned Product Search. Proceedings of the SIGIR 2025 Workshop on eCommerce, Padua, Italy. arXiv:2507.22213 doi:10.48550/arXiv.2507.22213

[66] Oleg Zendel, Sara Fahad Dawood Al Lawati, Lida Rashidi, Falk Scholer, and Mark Sanderson. 2025. A Comparative Analysis of Linguistic and Retrieval Diversity in LLM-Generated Search Queries. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM 2025)*. 4014–4023. doi:10.1145/3746252.3761382

[67] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards Personalized and Semantic Retrieval: An End-to-End Solution for E-commerce Search via Embedding Learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*. 2407–2416. doi:10.1145/3397271.3401446

[68] Yingbo Zhou, Utkarsh Porwal, and Roberto Konow. 2019. Spelling Correction as a Foreign Language. In *Proceedings of the SIGIR 2019 Workshop on eCommerce (eCOM 2019), CEUR Workshop Proceedings Vol-2410*. http://ceur-ws.org/Vol-2410/paper28.pdf

[69] Shengyao Zhuang, Linjun Shou, and Guido Zuccon. 2023. Augmenting Passage Representations with Query Generation for Enhanced Cross-Lingual Dense Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1827–1832. doi:10.1145/3539618.3591952

[70] Guido Zuccon, João Palotti, and Allan Hanbury. 2016. Query Variations and their Effect on Comparing Information Retrieval Systems. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM 2016)*. 691–700. doi:10.1145/2983323.2983723

A Experimental Configurations and Construction Statistics

A.1 History Threshold Sensitivity

Table 8 summarizes eligible user–product pairs and same-product cohort coverage under different minimum-history thresholds.

Table 8. Minimum-history thresholds vs. eligible user–product pairs and same-product cohort coverage.

$h_{\min}$	Musical Eligible Pairs	Musical ASINs ≥ 2 Users	Video Eligible Pairs	Video ASINs ≥ 2 Users	Baby Eligible Pairs	Baby ASINs ≥ 2 Users
10	168,385	23,068	359,159	25,540	219,161	22,704
30	**11,162**	**1,667**	**48,835**	**5,780**	**7,462**	**1,219**
50	2,235	223	14,738	2,216	1,224	142
80	399	19	3,997	661	174	9
100	178	3	1,982	307	66	2

A.2 Syntax Encoder Configuration

Table 9 lists the syntax encoder configuration. The StyleMLP output is L2-normalized; Gaussianity of these learned representations is a modeling approximation.

Table 9. Syntax encoder configuration.

Parameter	Setting
SVD dimension	256
Hidden dimension	128
Latent dimension	16
Contrastive temperature	0.1
Learning rate	3×10^{-3}
Training epochs	30

A.3 Gaussian Validity Filter Thresholds

A.4 Candidate Query Generation Configuration

Table 12 lists the candidate query generation model and configuration parameters.

A.5 Writing-Error Construction Statistics

Table 13 reports the personalized writing-error profile statistics of the selected users rather than the broader upstream population, and Table 14 reports the emitted personalized writing-error query statistics of the three product domains.

A.6 Retrieval and Reranking Configurations

Table 15 lists the core configurations of the five first-stage retrievers and the two LLM-based rerankers.

Table 10. Gaussian validity filter thresholds.

Parameter	Value
Audit distance $D_u^2(z)$	$(z-\mu_u)^\top(\lambda_u v_u v_u^\top + \mathrm{diag}(\sigma_u^2))^{-1}(z-\mu_u)$
Rank-1 variance and residual variances	> 0
Fitting floors for both variance terms	10^{-4}
Stored log-determinant and validation distances	finite
Validation membership in nominal reference region	$\geq 80\%$
Ideal 16D Gaussian reference boundary	$\chi^2_{16,0.95} = 26.296$
Retained profile fraction per subsample	80%
Repetitions	20
Valid-fit fraction over repetitions	≥ 0.90
Median Spearman distance-rank correlation	≥ 0.80
Median nominal-gate agreement	≥ 0.80

Table 11. Gaussian validity filtering results across the three product domains, with percentages calculated relative to the initial Gaussian-profile cohort within each domain.

Filter Stage	Requirement	Baby	Musical	Video	Total
Initial cohort	Entering filtering	2,154 (100.00%)	2,804 (100.00%)	7,727 (100.00%)	12,685 (100.00%)
Numerical validity	Valid parameters	2,142 (99.44%)	2,797 (99.75%)	7,712 (99.81%)	12,651 (99.73%)
Held-out calibration	+ calibration gate	1,889 (87.70%)	2,550 (90.94%)	6,805 (88.07%)	11,244 (88.64%)
Repeated-sampling stability	+ stability gate	**1,683 (78.13%)**	**2,283 (81.42%)**	**6,466 (83.68%)**	**10,432 (82.24%)**

Table 12. Candidate query generation configuration.

Parameter	Value
Generation model	Qwen2.5-0.5B-Instruct
Temperature (T)	0.7
Top-p	0.95
Maximum generation length	40 tokens
Candidates per round (K)	50
Minimum retained candidates ($K_{\min}$)	30
Maximum generation rounds	2
Maximum raw candidates per product	100
Candidate pool truncation	None

Table 13. Personalized writing-error profile statistics for the selected users across product domains.

Statistic	Baby Products	Musical Instruments	Video Games	Total
Distinct users within each domain	273	487	1,517	2,277
Historical sentences analyzed	13,650	24,350	75,850	113,850
Sentences with ≥ 1 retained writing-error edit	4,870	8,498	29,220	42,588
Total retained writing-error edits	7,410	13,116	48,728	69,254
Retained edits per user (mean)	27.14	26.93	32.12	30.41
Retained edits per user (median)	25	23	26	26
Distinct syntactic contexts	350	486	816	924

Table 14. Personalized writing-error query construction across product domains.

Statistic	Baby Products	Musical Instruments	Video Games	Total
Personalized syntactic expression query rows entering error generation	398	742	2,514	3,654
Edit events in emitted changed variants	888	1,700	5,911	8,499
Rows with a genuinely changed, nonempty personalized writing-error query	298	573	1,991	2,862
Rows with no surviving personalized writing-error query	100	169	523	792
Clean–Typo pairs rejected by the 0.90 semantic similarity constraint	198	353	1,429	1,980

Table 15. Core configurations of the evaluated retrieval and LLM-based reranking models.

Model	Checkpoint / Implementation	Core Configuration
First-stage retrievers		
BM25	bm25s (Lucene)	$k_1 = 1.5$, $b = 0.75$
SPLADE	naver/splade-cocondenser-ensembledistil	max length = 128
GTE-base	thenlper/gte-base	768-d, L2-normalized cosine similarity
ColBERTv2	colbert-ir/colbertv2.0	128-d token representations, MaxSim
BGE-M3 (Hybrid)	BAAI/bge-m3	dense/sparse/multi-vector weights = 1.0/0.3/1.0
LLM-based rerankers		
Qwen3-Reranker-8B	Qwen3-Reranker-8B	BM25 candidates, top-25 reranking, P(Yes) on first-token Yes/No logits, $T = 0$
Llama-3.1-8B	Meta-Llama-3.1-8B-Instruct (4-bit)	BM25 candidates, top-25 reranking, P(Yes) on first-token Yes/No logits, $T = 0$
Final ranking score of both rerankers: $s_{\text{final}} = 1.0s_{\text{LLM}} + 0.1s_{\text{rank}}$		

B Additional Absolute Effectiveness Results

B.1 Absolute Ranking Effectiveness

Table 16 reports absolute results on the full personalized syntactic expression query cohort.

Table 16. Absolute ranking effectiveness on all canonical personalized syntactic expression queries across three product domains and their all-domain average.

	Baby Products					Musical Instruments				
Model	**Hit@1**	**Hit@5**	**Hit@10**	**Hit@20**	**MRR**	**Hit@1**	**Hit@5**	**Hit@10**	**Hit@20**	**MRR**
BM25	14.07%	32.16%	42.71%	51.51%	23.33%	13.21%	28.30%	35.04%	42.72%	20.97%
SPLADE	14.32%	34.67%	42.96%	49.75%	23.50%	14.42%	29.65%	35.85%	41.37%	21.89%
GTE-base	8.29%	16.08%	18.84%	23.12%	12.25%	9.43%	19.95%	24.80%	30.59%	14.93%
ColBERTv2	7.54%	17.84%	24.62%	29.40%	12.65%	7.95%	15.36%	19.95%	25.34%	11.90%
BGE-M3 (Hybrid)	12.31%	31.16%	40.95%	49.25%	21.16%	16.17%	28.03%	34.23%	40.97%	22.49%
Llama-3.1-8B	16.33%	38.44%	45.23%	52.26%	25.98%	16.71%	30.59%	38.27%	44.61%	23.49%
Qwen3-Reranker-8B	15.83%	38.44%	45.73%	53.02%	25.69%	18.33%	33.29%	40.16%	44.20%	25.19%

	Video Games					All-domain-average				
Model	**Hit@1**	**Hit@5**	**Hit@10**	**Hit@20**	**MRR**	**Hit@1**	**Hit@5**	**Hit@10**	**Hit@20**	**MRR**
BM25	9.35%	19.25%	24.18%	29.47%	14.40%	12.21%	26.57%	33.98%	41.23%	19.57%
SPLADE	10.22%	21.20%	25.30%	30.79%	15.58%	12.99%	28.51%	34.70%	40.64%	20.32%
GTE-base	5.29%	11.54%	14.92%	18.58%	8.50%	7.67%	15.86%	19.52%	24.10%	11.89%
ColBERTv2	5.97%	11.85%	14.84%	17.98%	9.12%	7.15%	15.02%	19.80%	24.24%	11.22%
BGE-M3 (Hybrid)	10.46%	21.04%	24.94%	29.63%	15.58%	12.98%	26.74%	33.37%	39.95%	19.74%
Llama-3.1-8B	10.42%	20.92%	26.01%	30.27%	15.22%	14.49%	29.98%	36.50%	42.38%	21.56%
Qwen3-Reranker-8B	10.70%	22.00%	26.81%	30.39%	15.72%	14.95%	31.24%	37.57%	42.54%	22.20%

In each metric column, the best (highest) value is highlighted with a red background and the worst (lowest) value with a blue background, computed across all seven evaluated models. All-domain-average is the equal-weight average of the three domains.

B.2 Absolute Ranking Effectiveness on Personalized Syntactic Expression and Personalized Writing-Error Queries

Table 17 reports absolute results on both sides of the same paired records.

Table 17. Absolute effectiveness on the personalized syntactic expression and personalized writing-error sides of the same paired records.

Model	Baby Products									
	Hit@1		Hit@5		Hit@10		Hit@20		MRR	
	Clean	Error	Clean	Error	Clean	Error	Clean	Error	Clean	Error
BM25	15.10%	8.72%	33.56%	22.82%	45.30%	29.87%	53.69%	35.23%	24.46%	15.54%
SPLADE	14.77%	10.07%	36.58%	30.87%	45.64%	38.26%	52.35%	45.97%	24.60%	19.50%
GTE-base	9.06%	4.03%	16.78%	10.40%	19.13%	14.77%	24.50%	21.14%	12.88%	7.68%
ColBERTv2	7.38%	4.70%	19.46%	13.09%	26.17%	17.11%	30.87%	23.49%	13.25%	8.99%
BGE-M3 (Hybrid)	12.75%	9.40%	32.21%	25.50%	41.28%	35.23%	48.32%	41.95%	21.72%	17.55%
Llama-3.1-8B	17.45%	12.08%	40.94%	28.86%	47.99%	34.90%	55.03%	38.59%	27.60%	19.43%
Qwen3-Reranker-8B	16.78%	9.40%	39.93%	27.85%	46.64%	34.90%	55.37%	39.26%	27.17%	17.29%

Model	Musical Instruments									
	Hit@1		Hit@5		Hit@10		Hit@20		MRR	
	Clean	Error	Clean	Error	Clean	Error	Clean	Error	Clean	Error
BM25	12.74%	8.73%	27.57%	20.42%	34.55%	25.83%	42.23%	31.24%	20.43%	14.79%
SPLADE	15.18%	11.87%	29.32%	23.73%	35.25%	29.32%	40.66%	34.90%	22.39%	17.51%
GTE-base	10.30%	5.58%	20.07%	13.96%	25.13%	17.45%	30.37%	21.12%	15.51%	9.59%
ColBERTv2	8.20%	5.24%	15.88%	11.17%	19.55%	14.66%	25.83%	17.10%	12.14%	8.47%
BGE-M3 (Hybrid)	16.58%	14.31%	28.62%	23.91%	34.03%	29.67%	41.01%	36.65%	22.70%	19.67%
Llama-3.1-8B	17.10%	13.61%	30.19%	23.39%	37.70%	27.75%	44.15%	31.94%	23.62%	18.06%
Qwen3-Reranker-8B	17.80%	14.14%	32.98%	25.65%	39.62%	30.72%	43.80%	32.98%	24.73%	19.18%

Model	Video Games									
	Hit@1		Hit@5		Hit@10		Hit@20		MRR	
	Clean	Error	Clean	Error	Clean	Error	Clean	Error	Clean	Error
BM25	8.89%	6.73%	18.88%	14.06%	24.01%	17.98%	29.53%	22.00%	14.00%	10.45%
SPLADE	9.74%	7.28%	20.54%	15.72%	24.86%	20.04%	30.14%	24.46%	15.07%	11.57%
GTE-base	5.22%	3.57%	11.50%	6.88%	14.97%	9.49%	18.43%	12.41%	8.45%	5.58%
ColBERTv2	5.83%	3.06%	11.90%	7.38%	14.77%	9.79%	17.68%	12.46%	8.92%	5.47%
BGE-M3 (Hybrid)	10.30%	8.39%	20.49%	17.28%	24.36%	20.99%	29.08%	25.06%	15.26%	12.73%
Llama-3.1-8B	9.99%	6.83%	20.69%	15.77%	25.92%	19.94%	30.39%	23.20%	14.94%	10.75%
Qwen3-Reranker-8B	10.45%	6.98%	22.05%	16.62%	26.82%	20.29%	30.44%	23.20%	15.60%	11.12%

In each metric column, the best (highest) value is highlighted with a red background and the worst (lowest) value with a blue background, computed across all seven evaluated models. Clean columns report the personalized syntactic expression query side, and Error columns report the personalized writing-error query side.