# One Hierarchy, Two Systems: Semantic Product IDs for Discovery-Surface Ranking and Search-Page Query Reformulation


Steven Xu*
steven.xu@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Sanjyot Thete*
sanjyot.thete@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Saathvik Dirisala*
saathvik.dirisala@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Raghav Saboo*
raghav.saboo@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Nimesh Sinha
nimesh.sinha@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Leo Shao
leo.shao@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Elyse Winer
elyse.winer@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Sudeep Das
sudeep.das2@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Martin Wang
martin.wang@doordash.com
DoorDash Inc.
San Francisco, CA, USA

Kyle MacDonald
kyle.macdonald@doordash.com
DoorDash Inc.
San Francisco, CA, USA





## Abstract

Multi-merchant e-commerce catalogs contain equivalent and related products under different merchant-scoped identifiers, fragmenting behavioral evidence across merchants. Expert-defined taxonomies, meanwhile, are often too coarse for fine-grained discovery. We investigate whether a single hierarchical Semantic ID (SID) representation can support personalized ranking and query reformulation. Learned once from product-content embeddings, the hierarchy defines product concepts at multiple granularities that each application combines with its own behavioral and serving context. For ranking, we aggregate consumer affinity and product performance over SID prefixes and derive sequence features for candidate products and consumer histories. Controlled ablations show improved offline relevance, while online evaluation of the full ranking treatment shows stronger top-slot add-to-cart engagement and broader exposure for less-popular products. For query reformulation, we ground queries and session transitions in SID concepts, use the hierarchy for navigation and refinement, and filter suggestions against the merchant's assortment. Offline evaluation shows finer intent preservation than taxonomy and higher-quality suggestions than raw query-string transitions; online evaluation shows reduced search effort and earlier access to purchasable products. These results show that a shared semantic product hierarchy can support both recommendation and search while preserving the task-specific context required by each application.



## CCS Concepts

• **Information systems → Recommender systems**; **Information retrieval**.

## Keywords

semantic IDs, hierarchical clustering, residual quantization, recommender systems, query reformulation

**ACM Reference Format:**
Steven Xu, Sanjyot Thete, Saathvik Dirisala, Raghav Saboo, Nimesh Sinha, Leo Shao, Elyse Winer, Sudeep Das, Martin Wang, and Kyle MacDonald. 2026. One Hierarchy, Two Systems: Semantic Product IDs for Discovery-Surface Ranking and Search-Page Query Reformulation. In *Proceedings of the First Workshop on Unified Search and Recommendation (USRW '26), October 2, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 8 pages.


## 1 Introduction

E-commerce marketplaces bring together the catalogs of grocery, convenience, and general retail merchants within a shared discovery experience. Equivalent and closely related products often recur across merchants under different listing IDs. A product representation must therefore capture relationships across listings while preserving the distinctions that matter to consumers. The representation used to organize products determines both the granularity at which behavioral evidence can be aggregated and the products across which that evidence can be shared.

*These authors contributed equally to this research.

This representation choice is particularly important for personalized ranking. A merchant-scoped listing ID preserves exact identity, but an interaction with a product at one merchant does not strengthen the preference signal for an equivalent product elsewhere. A consumer may therefore purchase the same product concept across multiple merchants without generating enough evidence on any individual listing to learn a reliable preference. Human-defined taxonomies share evidence across broader product groups, but their emphasis on interpretability and catalog organization can make them too coarse for fine-grained personalization.

A related representation problem arises in query reformulation, where candidate reformulations are often derived from transitions between query strings observed in search sessions. String-level transitions fragment evidence across misspellings, abbreviations, and synonymous expressions. They can also be dominated by broad, high-frequency queries and conflate distinct intents when the same query has different meanings across business verticals. Products associated with a query through downstream behavior provide a natural signal for grounding its meaning in the catalog, allowing transitions to be modeled between product concepts rather than raw strings. This requires a product representation that is fine-grained enough to preserve useful distinctions and hierarchical enough to support both lateral pivots, such as *milk* to *cereal*, and narrower refinements, such as *milk* to *whole milk*.

Semantic IDs (SIDs), introduced in TIGER [8] for generative retrieval, encode products as short sequences of discrete codes obtained through residual quantization of product-content embeddings. Learned SIDs often exhibit a hierarchical structure: products sharing a prefix tend to be semantically related, with longer shared prefixes corresponding to progressively finer-grained product groups. This structure provides multiple levels at which related products can be grouped while retaining fine-grained product distinctions.

In this paper, we propose using the learned hierarchy of SIDs as a shared product representation across personalized ranking and query reformulation. The hierarchy is constructed once from catalog content, while each application independently organizes its interaction data around the product concepts it defines. In personalized ranking, SID prefixes serve as keys for behavioral aggregates and as inputs to learned history representations, allowing preferences to be modeled at multiple semantic resolutions. In query reformulation, we map queries to SID concepts using associated products, model transitions between those concepts, and descend the hierarchy when a more specific reformulation is appropriate. The two systems use different levels of the hierarchy according to their respective objectives and do not share model parameters, training objectives, or serving architecture. Figure 1 summarizes the overall design.

Our contributions are threefold: (1) we present a production-scale study of the learned hierarchy of SIDs as a reusable product representation across recommendation and search; (2) we develop task-specific uses of this representation in two independently developed applications: personalized item ranking, where consumer interactions are aggregated over SID prefixes at multiple levels of granularity, and query reformulation, where the SID hierarchy grounds query transitions in product concepts and supports hierarchical refinement; and (3) we provide empirical evidence that SIDs offer a more effective representation than expert-defined taxonomy in both applications, improving personalized ranking and enabling more specific, assortment-aware query reformulations.

## 2 Related Work

*Semantic IDs.* Semantic IDs (SIDs) map items to short sequences of discrete codes that preserve structure from an underlying embedding space. Originally proposed in TIGER [8], SIDs are constructed through residual quantization of item embeddings and used as autoregressive targets for next-item retrieval. Much of the subsequent work continues to study SIDs in generative recommendation, search, and unified models of both [5, 7]. In industrial ranking, Singh et al. [9] learn SID subpieces using SentencePiece and use their embeddings to represent items and user histories, finding them more effective than manually defined (n)-grams. Zheng et al. [11] develop prefix (n)-gram parameterizations and deploy SID-based sparse and sequential features for ads ranking. We similarly use SentencePiece-tokenized SIDs to represent products and consumer histories. Whereas prior ranking work primarily uses SID-derived tokens to parameterize learned representations, we additionally use SID prefixes as shared concept units across both applications: personalized ranking aggregates consumer interactions over these concepts, while query reformulation aggregates query-grounding and transition evidence over the same hierarchy.

*Query suggestion and reformulation.* Behavioral query suggestion commonly represents queries as nodes connected by transitions observed in search sessions. The Query-Flow Graph [1] applies random walks over these transitions to identify useful successor queries, while context-aware methods incorporate click-through evidence to reduce sparsity and ambiguity [3]. Applied e-commerce systems also transfer behavioral evidence from frequent queries to semantically related tail queries [10]. Our approach retains the interpretability and batch-serving advantages of a transition graph, but grounds queries in product concepts and models transitions between concepts in the SID hierarchy. This allows evidence to be pooled across lexical variants, supports both lateral reformulations and descent to more specific concepts, and filters candidates against the merchant's active assortment.

## 3 A Shared, Learned Product Hierarchy

*Semantic ID construction.* For product $i$, we concatenate selected catalog fields, such as item name, brand, and size, into a textual profile $t_i$. We encode $t_i$ using a pretrained text encoder. In practice, we use `gemini-embedding-001` [4], which produces a 3,072-dimensional embedding $\mathbf{x}_i$.

We construct the SID using residual-quantization $K$-means (RQ-$K$-means) with $L = 3$ stages and $K = 512$ centroids per stage. Each stage operates on an $L_2$-normalized input. Starting with $\mathbf{r}_{i,0} = \mathbf{x}_i$, stage $\ell \in \{0, \ldots, L-1\}$ assigns the current residual to its nearest centroid in codebook $C_\ell$ and subtracts the selected centroid:

$$c_{i,\ell} = \arg\min_{k \in \{0,\ldots,K-1\}} \left\|\mathbf{r}_{i,\ell} - \boldsymbol{\mu}_{\ell,k}\right\|_2^2,$$
$$\widetilde{\mathbf{r}}_{i,\ell+1} = \mathbf{r}_{i,\ell} - \boldsymbol{\mu}_{\ell,c_{i,\ell}}.$$

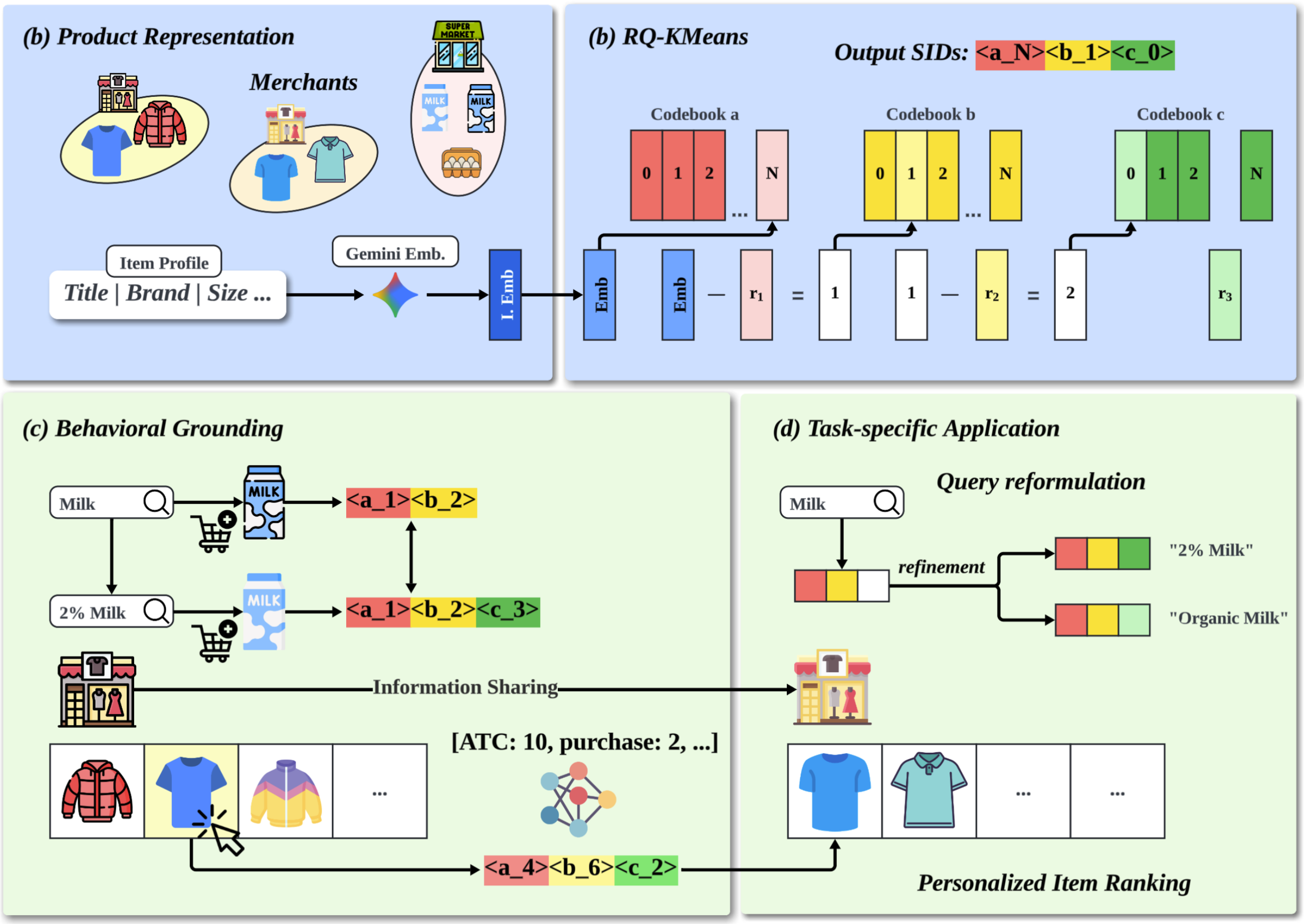


**Figure 1: Overview of shared SID construction and reuse across search and recommendation. Catalog product profiles are embedded and residual-quantized into a three-level SID hierarchy. Search queries and consumer interactions are then grounded and aggregated through SID prefixes, enabling cross-merchant preference transfer for personalized ranking and coarse-to-fine query reformulation through hierarchical descent.**

The resulting SID is $\mathbf{s}_i = [c_{i,0}, c_{i,1}, c_{i,2}]$. Let $\mathbf{s}_i^{(\ell)} = [c_{i,0}, \ldots, c_{i,\ell-1}]$ denote its prefix at depth $\ell$; we refer to depths 1, 2, and 3 as L1, L2, and L3, respectively.

*Learned hierarchy.* The prefix $n$-grams form nested partitions of the product catalog. In the learned code space, we observe that products sharing longer prefixes tend to be more semantically similar, suggesting that the hierarchy captures product concepts at increasing levels of specificity. These concepts provide an intermediate granularity between exact listing IDs and expert-defined taxonomy nodes. We use the same hierarchy in both applications: personalized ranking aggregates consumer interactions over prefixes at multiple depths, while query reformulation grounds queries and models transitions over the resulting product concepts.

*Characteristics of the learned hierarchy.* We examine how the product groups induced by SID prefixes change with shared-prefix length. Table 1 reports code-usage balance, cluster separation, and held-out semantic coherence for each prefix length.

**Table 1: Intrinsic characteristics of product groups induced by SID prefix levels. Lower Gini and DBI indicate more balanced code usage and better-separated groups, respectively; higher held-out cosine indicates stronger semantic coherence on unseen products.**

| Prefix level | Gini | DBI | Held-out cosine |
|---|---|---|---|
| L1 (coarse) | 0.472 | 3.934 | 0.957 |
| L2 (intermediate) | 0.508 | 2.065 | 0.965 |
| L3 (fine) | 0.467 | 0.976 | 0.981 |

Gini measures imbalance in prefix usage, with lower values indicating a more even distribution of products across groups. The Davies–Bouldin index (DBI) compares within-group dispersion with between-group separation, with lower values indicating more compact and better-separated groups. Held-out cosine measures the average cosine similarity between an unseen product and training products assigned to the same prefix group.

As shared-prefix length increases, DBI decreases and held-out cosine similarity increases, indicating that longer prefixes identify more compact and semantically coherent product groups. Code-usage imbalance remains comparable across the three prefix lengths. These observations suggest that the learned SID hierarchy captures product concepts at progressively finer levels of granularity.

## 4 Personalized Item Ranking with Semantic IDs

We first examine how the learned SID hierarchy supports personalized item ranking on discovery surfaces. On these surfaces, consumers browse a merchant's assortment through category- and theme-based carousels. A carousel may represent a grocery mission such as *Produce* or *Summer Grilling*, or a retail concept such as *Beauty*, *Apparel*, or *Household Supplies*. A retrieval stage selects candidate items for each carousel, and a personalized ranker determines their order for each consumer.

### 4.1 Ranking Model

The ranker uses a multi-task, multi-label neural network with CTR, ATCR, and CVR heads, corresponding to click, add-to-cart, and purchase outcomes, respectively. Its existing product-identity features include merchant-scoped listing IDs and expert-defined taxonomy prefixes. Listing IDs preserve exact identity within a merchant but cannot transfer a consumer's history to equivalent or related products at another merchant. Taxonomy prefixes support broader sharing but may group products that are too heterogeneous to represent a specific grocery or retail preference. Consequently, established fine-grained preferences may not be reflected near the top of a carousel, particularly at merchants from which the consumer has not previously ordered.

### 4.2 Feature Design

We construct two complementary families of SID-derived features. Dense features explicitly aggregate behavioral statistics over the product concepts defined by SID prefixes. This provides a strong inductive bias without requiring the ranker to learn each high-cardinality concept solely from a trainable embedding. Sequence features preserve information from the SID code sequence and support learned interactions between candidate products and consumer histories. Table 2 summarizes representative features from both families.

Following prior work on SID parameterization [9, 11], we use the prefix $n$-grams defined in Section 3 as aggregation keys for dense features and SentencePiece subwords as the units of item- and consumer-side sequence features.

*Feature identifiers.* Each prefix $n$-gram is mapped to a deterministic integer identifier:

$$p_i^{(n)} = \sum_{j=0}^{n-1} \left( (c_{i,j} + 1)K^j - 1 \right), \qquad n \in \{1, 2, 3\}, \quad K = 512.$$

The identifiers for L1, L2, and L3 are stored in separate fields, making the mapping collision-free within each prefix level.

To construct SentencePiece inputs, we map each (position, code) pair to a distinct symbol from an alphabet of $3K = 1{,}536$ symbols, converting each three-code SID into a three-symbol string. We train SentencePiece models [6] on an impression-weighted corpus of these strings. The resulting pieces may span one, two, or three adjacent SID symbols and therefore need not coincide with hierarchy prefixes.

*Dense aggregate features.* SID prefixes allow behavioral evidence to be shared at several semantic resolutions. Consumer-level aggregates capture individual affinity for the product concepts represented by each prefix, while global and regional aggregates provide overall and locally conditioned performance priors. We compute these signals at all three prefix levels and, where applicable, over multiple lookback windows. The ranker can therefore combine the broader coverage of shallow prefixes with the greater specificity of deeper prefixes.

*Sequence features.* We evaluated both the Unigram LM and BPE variants of SentencePiece. The Unigram LM vocabulary consisted largely of SID prefix $n$-grams and therefore overlapped substantially with the information already captured by the prefix-keyed dense features. We use BPE to obtain a more complementary set of subsequences and refer to the resulting units as SPM tokens.

On the item side, the candidate product is represented by the SPM tokens obtained from its SID. On the consumer side, we collect products ordered during the preceding 180 days, expand their SPM token lists, and aggregate order frequency by token. Distinct tokens are ranked by order count, with recency used as a tie-breaker, and the highest-ranked tokens are retained as a compact representation of recurring semantic preferences.

The item- and consumer-side tokens share an embedding table, allowing the ranker to relate candidate-product subwords directly to the consumer's historical preferences. Let $\mathcal{T}$ denote the SPM token list for either input. We use a vocabulary of $N = 2 \times 10^5$ learned tokens and reserve one additional entry for the null token, giving a shared trainable embedding table $\mathbf{E} \in \mathbb{R}^{(N+1)\times 64}$. We obtain the representation of each input by mean pooling:

$$\mathbf{h}(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbf{E}_t.$$

A separate mask indicates whether the input is empty. The pooled representations and embedding parameters are learned jointly with the ranking objective.

### 4.3 Offline Evaluation

The ranking model is trained on $M$ days of logged interactions and evaluated on data from day $M + 1$. The full candidate (FC) includes SID-derived features alongside concurrent non-SID feature updates, so comparison with the production baseline alone would not isolate the contribution of SID. We therefore construct an ablated candidate (FC-A) that removes all SID-derived features while holding the complete non-SID feature configuration fixed.

We evaluate FC and FC-A on the CVR head using MRR@$K$ and NDCG@$K$ for $K \in \{3, 5, 10\}$. We report $K = 5$ in Table 3 because five items are visible on the evaluated discovery surface without scrolling. Both metrics are computed over sessions containing at least one conversion.

**Table 2: Examples of SID-derived ranking features.**

| Feature family | Feature Scope | Description |
|---|---|---|
| Dense aggregates | Consumer | Order frequency, purchase recency, and subtotal statistics for each SID prefix in the consumer's history. |
| | Submarket | Impressions, clicks, ATC actions, purchases, and associated rates for each SID prefix within a submarket. |
| Sequence | Item | SPM token sequence obtained from the candidate item's SID. |
| | Consumer | SPM tokens aggregated from previously ordered products and ranked by their associated order counts. |

**Table 3: Offline ablation of SID-derived ranking features. All values are relative gains over the production baseline.**

| Model | MRR@5 | NDCG@5 |
|---|---|---|
| FC-A | +2.10% | +2.92% |
| FC | +6.98% | +6.76% |

**Table 4: Online results for the full ranking treatment. The treatment includes SID-derived features alongside concurrent non-SID updates.**

| Outcome | Relative gain |
|---|---|
| Subtotal | +0.31% |
| Average carousel ATC rate | +5.5% |
| Item ATC rate, position 1 | +8% |
| Item ATC rate, position 2 | +16% |
| Item ATC rate, position 3 | +6% |

As shown in Table 3, FC substantially outperforms FC-A on both reported metrics. The same pattern holds at $K = 3$ and $K = 10$. Since the two candidates differ only in their SID-derived features, the ablation indicates that these features contribute the majority of the full candidate's offline gain.

## 4.4 Online Experiment

The online experiment evaluates a production feature bundle that combines the SID-derived features with concurrent non-SID updates. Consumers were randomized approximately evenly among three arms: the existing ranker, the feature-bundle treatment, and the same treatment with an additional serving optimization. We report the 21-day comparison between the existing ranker and the feature-bundle treatment.

As shown in Table 4, the treatment improves add-to-cart engagement across the carousel and at each of the first three positions. These engagement gains translate into a 0.31% relative increase in subtotal.

The treatment also reduces popularity concentration. The average historical popularity of the item displayed in the first carousel position decreases by 18.1%, while the share of first-position impressions assigned to blockbuster items decreases by 2.1 percentage points. Thus, the treatment improves engagement while allocating less top-position exposure to historically dominant products.

**Table 5: Online impact of SID-based query reformulation on search efficiency. Changes are relative to control; lower ATC position and scroll depth are better.**

| Metric | Relative change | 95% CI |
|---|---|---|
| Purchase MRR | +0.558% | [+0.294, +0.823]% |
| ATC position | −1.571% | [−2.159, −0.982]% |
| Search scroll depth | −1.866% | [−2.262, −1.470]% |

Because the production bundle includes both SID and non-SID updates, the online experiment alone does not isolate the contribution of SID. However, both the offline ablation study and post-experiment analysis indicate that SID-derived features account for most of the observed improvement.

# 5 Query Reformulation with Semantic IDs

Beyond personalized item ranking, the same SID representation can support search-time intent discovery. We study query reformulation for suggested-query pills displayed alongside search results on a merchant's Store Page. A useful suggestion must advance the consumer's shopping mission and correspond to products in the merchant's active assortment. Shopping sessions may involve lateral basket-building moves, such as *milk* to *cereal*, or refinements to a product type, brand, or variant.

Prior work on query reformulation often mines query-to-query transitions directly from search sessions. Representing queries as raw strings fragments behavioral evidence across misspellings, abbreviations, and synonymous expressions, while pooling the same string across business kinds can conflate different meanings. We instead use the SID hierarchy as a catalog-grounded concept space that supports both lateral navigation and progressive refinement.

## 5.1 Methodology

*Query-to-concept grounding.* For each query and business vertical (BV), we aggregate associated ATC events by SID prefix. For a query–BV pair with at least five events, we assign the dominant L2 prefix if it accounts for at least 30% of the evidence. Otherwise, we apply the same criterion at L1. If neither level qualifies, a guarded fragmented-query path retains the raw query node rather than assigning an unsupported concept. Conditioning on BV allows the same query string to resolve to different product concepts across retail contexts.

*Lateral navigation.* Given consecutive queries $(q_t, q_{t+1})$ in a session, we replace the string pair with a transition between their grounded SID prefixes. We estimate transition counts and marginals separately within each BV and rank candidate edges by normalized pointwise mutual information (NPMI) [2]. We retain an edge when its observed count is at least 5, its expected count under independence is at least 1, and its NPMI is at least 0.1. This construction pools lexical variants that resolve to the same concept while preserving BV-specific interpretations of ambiguous queries.

*Hierarchical refinement.* Queries grounded to the same SID prefix share a graph node, causing transitions between finer-grained intents to collapse into self-loops and be discarded. This is particularly limiting at L2, where the graph captures lateral pivots across concepts but misses refinements within a shared parent. We therefore add a parallel path that descends from an L2 prefix to its L3 children, ranking them with query-specific ATC evidence when available and parent-level popularity otherwise.

*Query rendering.* Because SIDs are internal identifiers, a language-model prompt renders target concepts from both candidate-generation paths as short consumer-facing queries using representative products from each concept. A second pass removes unusable source–target pairs, and the remaining candidates are ranked by embedding similarity between the source query and the rendered query. Language generation is confined to this rendering step; observed behavior and catalog concepts determine the candidate structure.

*Assortment-aware filtering.* At serving time, candidates are filtered against the merchant's active assortment. A target concept is eligible when the assortment contains at least one active item assigned to that concept, preventing suggestions for intents the merchant cannot fulfill.

### 5.2 Offline Evaluation

Our offline evaluation focuses on two complementary questions: whether SIDs preserve fine-grained intent distinctions better than the product taxonomy, and whether transitions over SID concepts produce better reformulations than transitions over raw query strings.

For the taxonomy comparison, we measure how often queries expressing different intents map to the same concept. Such transitions collapse into self-loops and cannot generate reformulation candidates. Taxonomy collapses 18.8% of intent-changing transitions, compared with 10.9% for SIDs. The finer SID representation therefore preserves more observed behavioral signal for candidate generation.

We next compare end-to-end suggestion quality against a query-string transition graph. An LLM judge scores usefulness, target-text quality, and distinctiveness using the labels *bad*, *acceptable*, and *good*, which are mapped to 0, 0.5, and 1, respectively. On a human-labeled set of 200 query pairs, the judge achieves 78% exact agreement and 80% agreement on usefulness. Among queries served by both systems, rank-one judged quality increases from 0.522 for the query-string graph to 0.734 for the catalog-grounded SID system.

Together, these results indicate that SIDs provide a more discriminative representation of product intent than taxonomy and a stronger foundation for mining behavioral query transitions than raw query strings.

### 5.3 Online Experiment

We evaluate the complete SID-based reformulation module in a consumer-randomized experiment against a control without suggested-query reformulations. Table 5 reports selected outcomes as relative changes from control.

The reformulation module improves purchase MRR while reducing both the position of ATC and search scroll depth. Together, these changes indicate that consumers reach relevant, purchasable items earlier and with less search effort.

## 6 Qualitative Analysis

Appendix Table 6 illustrates two properties of the SID representation that benefit both applications. First, products with similar semantics share codes even when they belong to different merchant-scoped listings, allowing behavioral evidence to transfer across merchants. Second, the prefix hierarchy exposes multiple levels of granularity: coarser prefixes pool evidence, while deeper prefixes distinguish more specific product intents.

These examples also expose the trade-off introduced by semantic compression. Mapping many products or queries to one prefix increases statistical support, but an overly broad or semantically mixed prefix can connect unrelated behaviors. In reformulation, such prefixes may become high-degree graph hubs that propagate irrelevant suggestions. Product-based query grounding can also capture the concept ultimately added to cart rather than the intent expressed by the original query. In ranking, products near a quantization boundary may receive different codes despite being useful substitutes, while products sharing a prefix may still differ on attributes that matter to a particular consumer.

The applications address this trade-off by restoring information outside the SID itself. Ranking combines prefixes at multiple depths with consumer-specific and product-specific signals. Query reformulation conditions grounding and transitions on BV, uses query-specific evidence when descending to L3, and filters candidates against the merchant's assortment. The SID hierarchy therefore supplies a transferable semantic prior rather than a complete representation of task intent.

## 7 Cross-System Findings

Appendix Table 7 summarizes how the two applications use the same SID hierarchy. In both cases, SIDs replace a fragmented behavioral unit with a semantic unit over which evidence can be pooled. The applications differ in which hierarchy depths they use, which behavioral signals they attach, and which information they restore before producing an output.

The shared pattern is to use SIDs for semantic evidence pooling and recover task-specific context before acting. Coarser prefixes provide support and transferability, while deeper prefixes recover specificity. The appropriate operating depth is application-dependent: ranking can consume several depths jointly, whereas reformulation assigns distinct roles to L2 navigation and L3 refinement.

This separation also clarifies the role of SIDs relative to existing identifiers. Exact product IDs remain necessary for identity and serving, and taxonomy remains useful for business organization. SIDs complement these representations by supplying a fine-grained, transferable hierarchy that can be reused across discovery tasks without requiring the applications to share the same model or decision logic.

## 8 Conclusion

We show that a single hierarchical Semantic ID vocabulary can support discovery systems spanning recommendation and search. Personalized item ranking uses SID prefixes as transferable feature keys over consumer–product interactions, while query reformulation uses the same hierarchy to ground queries, mine concept transitions, and generate coarse-to-fine suggestions within a merchant's assortment.

Across both applications, SIDs improve the task-specific outcomes enabled by the shared hierarchy. In ranking, SID-derived features add incremental predictive signal and contribute materially to relevance gains. In query reformulation, SID concepts preserve finer intent distinctions than taxonomy, improve suggestion quality over query-string transitions, and reduce search effort online. These findings show that search and recommendation can share a semantic product representation without requiring a jointly trained model. SIDs complement exact identifiers and product taxonomies with a fine-grained, transferable hierarchy that each application can combine with its own behavioral signals and serving context.

## Acknowledgments

We thank Kevin Zhai, Abhishek Tambat, Camrick Solorio, Akshad Viswanathan, Danielle Rommerdahl, Natalia Dougan, Vivek Paharia, Kyle Hsiao, and Tito Anammah for their support in this work.

## References

[1] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. 2008. The Query-Flow Graph: Model and Applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. 609–618.

[2] Gerlof Bouma. 2009. Normalized (Pointwise) Mutual Information in Collocation Extraction. In *Proceedings of the Biennial GSCL Conference*. 31–40.

[3] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-Aware Query Suggestion by Mining Click-Through and Session Data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 875–883.

[4] Google. 2026. Gemini API: Embeddings. https://ai.google.dev/gemini-api/docs/embeddings. Accessed July 27, 2026.

[5] Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. 2025. Generative Recommendation with Semantic IDs: A Practitioner's Handbook. *arXiv preprint arXiv:2507.22224* (2025).

[6] Taku Kudo and John Richardson. 2018. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 66–71. doi:10.18653/v1/D18-2012

[7] Gustavo Penha, Edoardo D'Amico, Marco De Nadai, Enrico Palumbo, Alexandre Tamborrino, Ali Vardasbi, Max Lefarov, Shawn Lin, Timothy Heath, Francesco Fabbri, and Hugues Bouchard. 2025. Semantic IDs for Joint Generative Search and Recommendation. *arXiv preprint arXiv:2508.10478* (2025).

[8] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, Vol. 36. arXiv:2305.05065

[9] Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, Ed H. Chi, and Xinyang Yi. 2024. Better Generalization with Semantic IDs: A Case Study in Ranking for Recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 1039–1044. doi:10.1145/3640457.3688190

[10] Ziqi Zhang, Yupin Huang, Quan Deng, Jinghui Xiao, Vivek Mittal, and Jingyuan Deng. 2024. Towards Scalability and Extensibility of Query Reformulation Modeling in E-Commerce Search. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. arXiv:2402.11202 https://arxiv.org/abs/2402.11202

[11] Carolina Zheng, Minhui Huang, Dmitrii Pedchenko, Kaushik Rangadurai, Siyu Wang, Fan Xia, Gaby Nahum, Jie Lei, Yang Yang, Tao Liu, Zutian Luo, Xiaohan Wei, Dinesh Ramasamy, Jiyan Yang, Yiping Han, Lin Yang, Hangjun Xu, Rong Jin, and Shuang Yang. 2025. Enhancing Embedding Representation Stability in Recommendation Systems with Semantic ID. *arXiv preprint arXiv:2504.02137* (2025).

# A Qualitative Examples and Cross-System Comparison

**Table 6: Illustrative effects of SID transferability and hierarchical granularity.**

| SID property | Example | Observed effect |
|---|---|---|
| **Personalized item ranking** | | |
| Cross-merchant transfer | Prior purchases of mushrooms and chicken at another merchant | Matching products rise toward the top despite having different merchant-scoped identifiers. |
| Fine-grained semantic affinity | Prior purchase of a makeup brush | A related brush set rises above a makeup sponge, and more brush products appear among the top results. |
| **Query reformulation** | | |
| Hierarchical refinement | Source query: *clay mask* | L3 descent introduces more specific intents, including *sheet mask* and *hydrating mask*. |
| Coarse-to-fine navigation | Source query: *car clean* | L3 adds *interior wipes*, *interior cleaner*, and *leather wipes*, while L2 retains a useful lateral pivot. |

**Table 7: How personalized ranking and query reformulation reuse the shared SID hierarchy.**

| SID role | Personalized item ranking | Query reformulation |
|---|---|---|
| Input unit | Merchant-scoped product listings | Raw query strings |
| Mapping into SID space | Products in consumer interaction histories | Queries grounded through associated products |
| Evidence pooled | Cross-merchant interactions among semantically related products | Transitions across lexical variants of the same concept |
| Hierarchy use | L1–L3 prefixes used jointly as ranking features | L2 transitions for navigation; L3 descent for refinement |
| Task-specific signals | Affinity, recency, product performance, and interaction sequences | Transition strength, query evidence, and concept popularity |
| Context restored | Consumer, candidate-product, and regional context | BV, source query, and merchant assortment |
| Main representation risk | Shared prefixes may hide preference-relevant attributes | Broad prefixes may collapse intents or become graph hubs |