Product Search NLP Evaluation

Christine Xu
Shio Huang
Yishi Wang
Lucy Xu

Table of contents

01

Problem Statement

02

Baseline Pipeline & Tagging Model

03

Multi-Agent Query-Time Enrichment

04

Pre-Tagging Enrichment

05

Claim-Level Evaluation & Insights

06

Recommendations & Future Work

Problem Statement:
Why Natural-Language Product Search Is Difficult

The product search task in GNPD is challenging because:

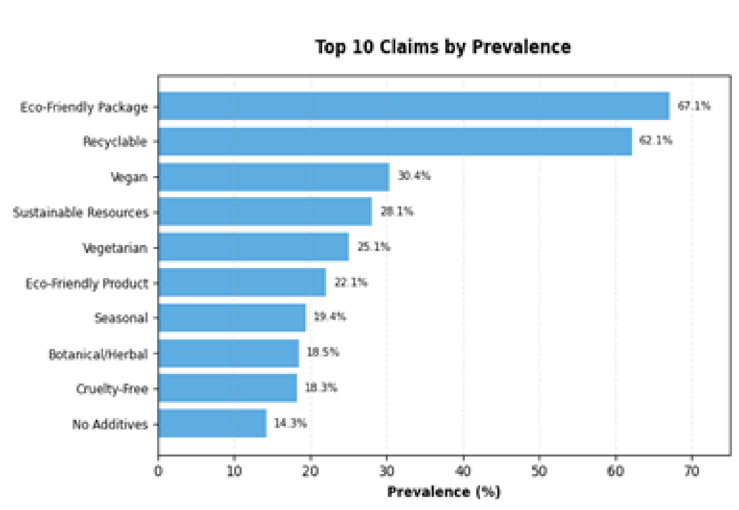
- **User intent is expressed in free text:** "eco packaging", "paraben free", "blueberry high-protein yogurt" do not map cleanly to GNPD's categories, claims, or ingredient fields.
- **Product Language is inconsistent and unstandardized:** "environmentally friendly", "paraben-free", "illuminating formula"
- **Model need clean, structured signals to perform well**: LLMs and embedding models perform better when products have normalized attributes.

Our Goal: Build a scalable and accurate natural-language product search system for GNPD by aligning user intent with heterogeneous product data through preprocessing, enrichment, tagging, embedding models, and LLM-based evaluation so users can retrieve relevant products without relying on GNPD's internal taxonomy.

Product Claims Distribution

- **Total Unique Claims:** 150+
- **Evaluated Claims:** 24 claims across 5 categories (>5% prevalence, >238 products each)

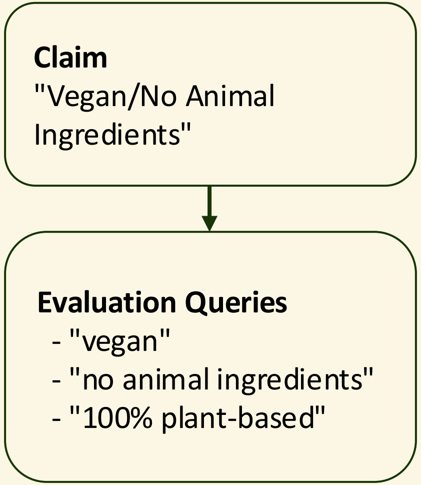
Category	# Claims	Example Claims
Dietary Restrictions	2	Gluten Free, Low Allergen
Lifestyle & Diet	3	Vegan, Vegetarian, Organic
Clean Label	4	No Additives, No Artificial Colors/Flavors, Sugar Free
Ethical & Sustainability	4	Recyclable, Eco-Friendly, Cruelty-Free, Sustainable
Nutritional Fortification	1	Vitamin/Mineral Fortified
Personal Care	2	Paraben Free, For Sensitive Skin
Specialty	8	Seasonal, Botanical, Moisturizing, Anti-Aging, Premium



Building Ground Truth from Claims

- Focus on **24 high-prevalence claims** (>5% prevalence, >238 products each)
- Generate **50 natural language queries** with 2-4 variations per claim
- Balanced Set: **50/50 positive/negative** (200+200 per query)

Example:



Claim
"Vegan/No Animal Ingredients"

Evaluation Queries
- "vegan"
- "no animal ingredients"
- "100% plant-based"

Combined Claims Strategy:

"Sugar Free/No Added/Low Sugar" combines:
- "Sugar Free" (65 products)
- "No Added Sugar" (116 products)
- "Low/Reduced Sugar" (63 products)

→ Total: 244 products (UNION)

Metrics & Evaluation Design (Query-Time Enrichment)

1. Conditions compared

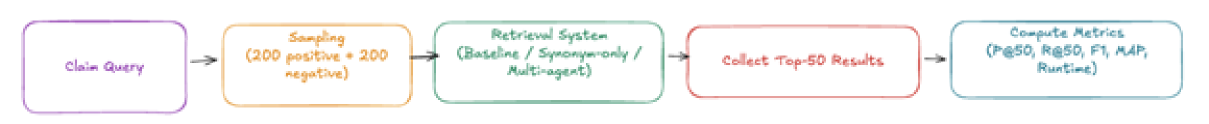
Use three query-processing strategies:

- **Baseline** (no enrichment)
- **Synonym-only** (lexical variants)
- **Full multi-agent system** (synonym + domain + semantic + fusion)

2. Dataset & Sampling

We validate across real claim categories:

- 50 claims across sustainability, ingredients, botanical, beauty, etc.
- Balanced sampling:
  400 products per claim (200 positive + 200 negative


Claim Query
Sampling
(200 positive + 200 negative)
Retrieval System
(Baseline / Synonym-only / Multi-agent)
Collect Top-50 Results
Compute Metrics
(P@50, R@50, F1, MAP, Runtime)

3. Metrics @k = 50

We evaluate how well each system retrieves relevant products:

- **Precision@50**: how clean the top 50 results are
- **Recall@50**: how many relevant items are covered
- **F1@50**: balanced precision/recall
- **MAP@50**: how early relevant items appear in ranking
  Runtime: cost & scalability of query-time enrichment

Keyword Extraction with KeyBERT

Pre-tagging

Enrich the product descriptions before tagging and indexing

Why KeyBERT

- Converts long product descriptions into clean, representative keywords
- Uses BERT sentence embeddings to capture semantic meaning

How It Works

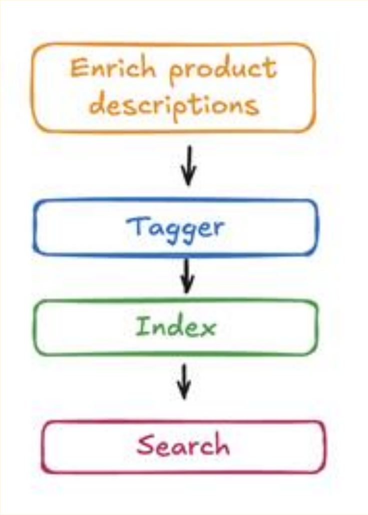
- Generate document embedding with Sentence-BERT
- Create candidate phrases and rank them by cosine similarity
- Select top-N phrases as product keywords

Benefits

- Produces consistent, high-quality tags for search and retrieval
- Improves BM25 + embedding hybrid search performance
- Automates metadata enrichment at scale

Example

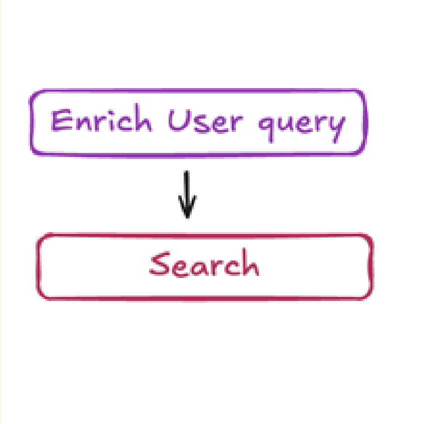
- “Organic vanilla yogurt… gluten-free, low sugar” →
- organic, vanilla yogurt, gluten free, low sugar



Where Enrichment Sits

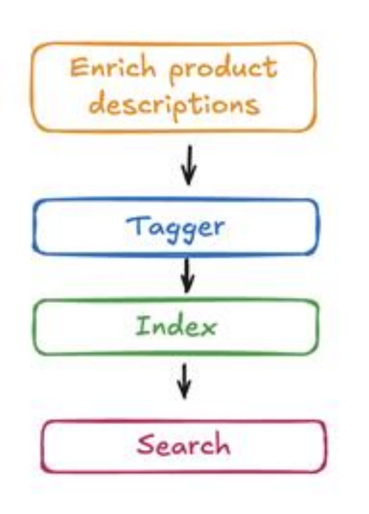
Query-time enrichment(A)

Enrich the user query before search



Pre-tagging enrichment(B)

Enrich the product descriptions before tagging and indexing



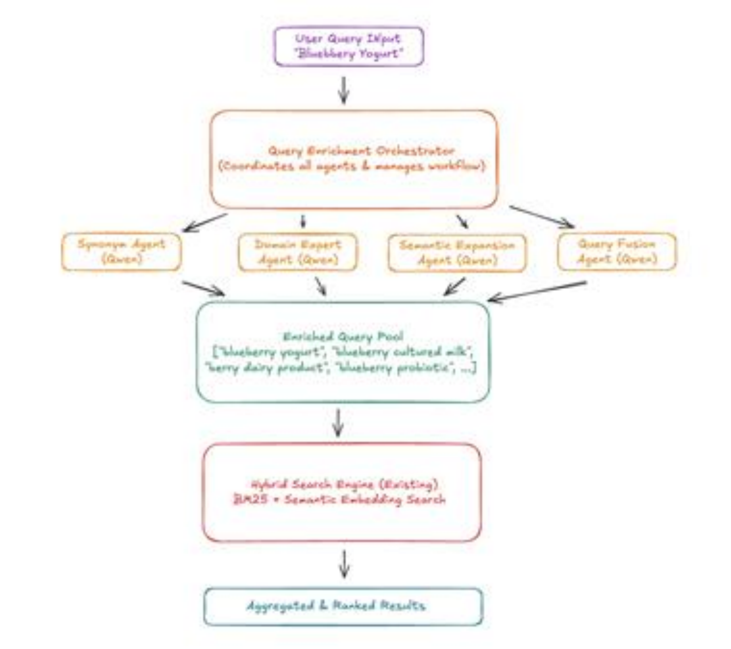
How Current Enrichment Works (Multi-Agent System)

A specialized multi-agent system expands queries in controlled ways to improve coverage while limiting semantic drift

Query-Time Pipeline

User Query

- Synonym Agent (lexical variants)
- Domain Expert Agent (label/certification language)
- Semantic Expansion Agent (paraphrases / conceptual neighbors)
- Fusion Agent (rank + select best candidates)
- Enriched Query Set
- Hybrid Retrieval (BM25 + embeddings) → Top-k results


User Query INput
"Bluebbery Yogurt"
Query Enrichment Orchestrator
(Coordinates all agents & manages workflow)
Synonym Agent
(Qwen)
Domain Expert
Agent (Qwen)
Semantic Expansion
Agent (Qwen)
Query Fusion
Agent (Qwen)
Enriched Query Pool
["blueberry yogurt", "blueberry cultured milk",
"berry dairy product", "blueberry probiotic", ...]
Hybrid Search Engine (Existing)
BM25 + Semantic Embedding Search
Aggregated & Ranked Results

Important: Enrichment produces a small, curated set of candidate queries rather than a large uncontrolled list, balancing coverage and relevance.

Query-Time Results (K=50)

What we see

- Full enrichment retrieves slightly more relevant results (Recall/F1).
- Early ranking quality drops (MAP).
- Runtime jumps from 17.5 → 205.7 sec (more than 10×).

What it means

- Vocabulary expansion helps recall, but introduces drift.
- Large runtime cost makes query-time enrichment impractical for large-scale search.

Claim-Level Insights: Where Enrichment Helps (and Where It Hurts)

A. Winners: Where Enrichment Helps

(Vocabulary expansion aligns well with product descriptions)

- ❑ **Environmentally Friendly Product:** 0.008 → 0.040 (+400 percent)
  Reason: Captures "eco-friendly", "green", "sustainable"
- ❑ **Sustainable Resources:** 0.024 → 0.048 (+100 percent)
  Reason: Adds "sustainably sourced", "renewable materials"
- ❑ **Paraben Free:** 0.043 → 0.064 (+49 percent)
  Reason: Handles "paraben-free", "no parabens", "without parabens"
- ❑ **Botanical / Herbal:** 0.040 → 0.056 (+40 percent)
  Reason: Adds common botanical keywords
- ❑ **Moisturising / Hydrating:** 0.144 → 0.160 (+11 percent)
  Reason: Stable synonym space, low drift risk

Pattern: Works best for broad, descriptive, positive claims.

B. Losers: Where Enrichment Hurts

(Strict or negation-based claims suffer from drift)

- ❑ **Free from Artificial Colourings:** 0.064 → 0.024 (–62 percent)
- ❑ **No Additives / Preservatives:** 0.056 → 0.040 (–29 percent)
- ❑ **Vegan / No Animal Ingredients:** 0.056 → 0.048 (–14 percent)

Why these fail

- **Semantic drift:** "natural coloring" ≠ "no artificial coloring"
- **Over-generalization:** "plant-based" not always vegan
- **Fusion dilution:** weak expansions leak into final enriched set

Pattern: Constraint / negation claims are fragile.

C. Synonym-only: Too Noisy Without Domain Rules

(Mixed impact depending on claim type)

- ❑ **Helps:** Organic, Antioxidant, Botanical / Herbal
- ❑ **Hurts:** Brightening / Illuminating, Ethical: Animal, Ethical: Recycling

Why?

- **Synonym-only methods generate non-standard, domain-incorrect, or overly broad terms**
- **No guardrails → precision drop**

Pattern: Synonyms alone are unstable and inconsistent.

Main Diagnosis: Vocabulary Mismatch Between Query & Index

The enriched query vocabulary does not exist in the tagging index. This creates misalignment and limits the benefit of enrichment.

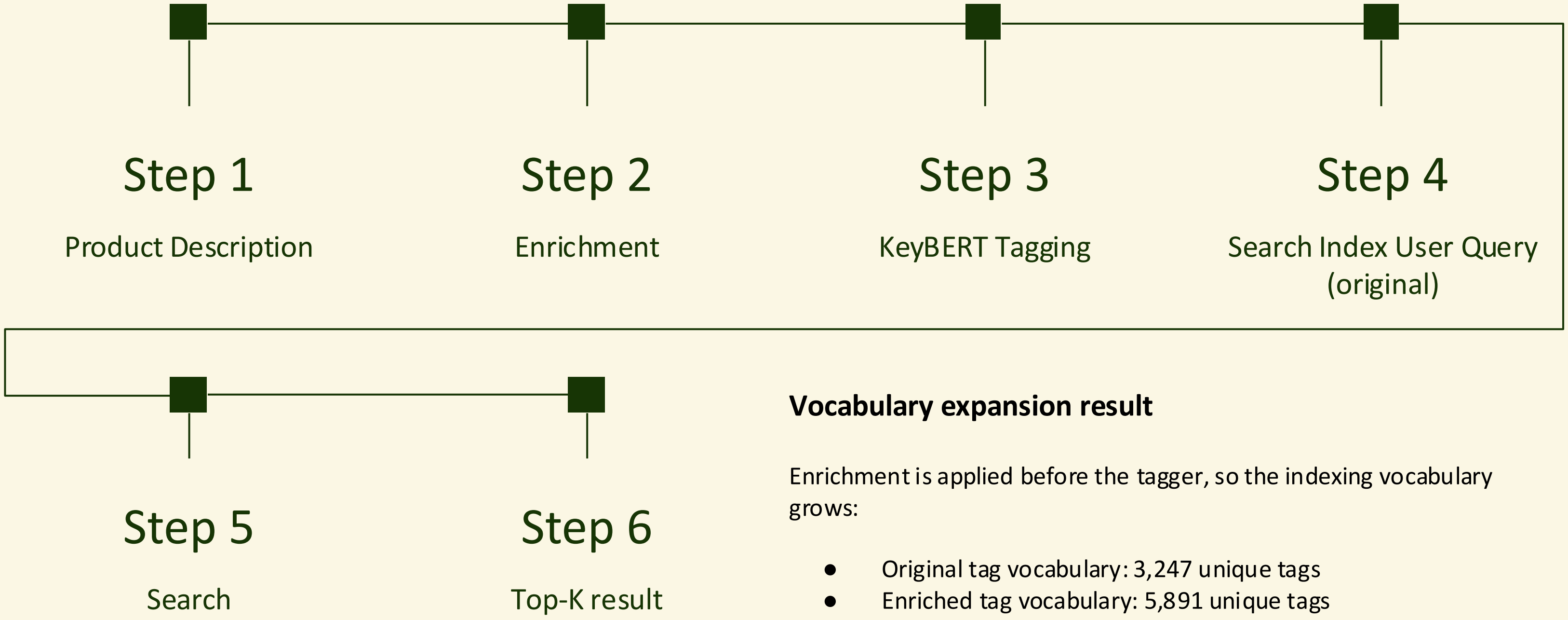
A. What actually happens

Enrichment expands the user query, but the search index relies on tags generated from original product descriptions. Since tag vocabulary is much smaller, enriched expansions often have no matching tag, leading to poor alignment.

B. Mismatch example

Component	Text
Product text	“environmentally friendly packaging”
Baseline tags	packaging green
Enriched query	“eco-friendly”, “sustainable”, “renewable”
Problem	**None of these enriched tokens exist in tag vocabulary → retrieval drifts**

Test: Enrichment Before Tagging (Pre-Tagging Enrichment for 500 samples)



Step 1

Product Description

Step 2

Enrichment

Step 3

KeyBERT Tagging

Step 4

Search Index User Query (original)

Step 5

Search

Step 6

Top-K result

Vocabulary expansion result

Enrichment is applied before the tagger, so the indexing vocabulary grows:

- Original tag vocabulary: 3,247 unique tags
- Enriched tag vocabulary: 5,891 unique tags
- Total expansion: +81.4 percent

Performance Comparison @ K = 50 Metrics

Metric @50	Improvement	% Change
Precision@50	+0.0637	**+35.1 percent**
Recall@50	+0.0164	**+36.6 percent**
F1@50	+0.0259	**+36.1 percent**

Why This Is Decisive

- **Fixes the mismatch:** enriched vocabulary is now in the index, so enriched queries align with searchable tags.=
- **Quality jumps:** Precision, Recall, and F1 all improve by about 35 percent.
  No latency cost: LLM enrichment happens offline once, not during query-time.
- **Scalable:** Works for millions of products because enrichment is precomputed.

Conclusion: Pre-tagging enrichment delivers substantially better retrieval and is production-viable, unlike query-time enrichment.

Future Direction (What We Recommend)

1. Strengthen Product Data Normalization and Preprocessing

Cleaning and standardizing OCR text, descriptions, and structured fields improves all downstream methods.

- Reduce noise in OCR text
- Normalize claims, ingredients, and packaging language
- Standardize metadata formats

Purpose: Provide models with consistent, high-quality signals.

2. Use Query-Time Enrichment Only as a Selective Fallback

Query-time LLM enrichment can help, but should not be the default due to latency and drift.

- Baseline retrieval is sparse or low-confidence
- Claims historically benefit from vocabulary expansion (for example, sustainability, botanical, ingredient-heavy queries)

Purpose: Improve recall in specific cases without incurring system-wide cost.

3. Add Claim-Aware Routing & Guardrails

Some claims are fragile, especially negation or constraint-based ones, so LLM expansions must be controlled.

- Negation detection ("no X", "free from X") → block risky semantic expansions
- Similarity thresholds + formatting rules to avoid drift
- Downweight drift-prone expansions during fusion

Goal: Reduce semantic drift and protect precision.

4. Evaluate at Claim-Class Granularity

Averaged metrics hide important differences.

- Report descriptive vs constraint/negation claims separately
- Move from balanced offline evaluation → prevalence-weighted or online A/B testing

Outcome: Ensures future improvements target the claims that matter most.

Thanks!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**