From Traditional Keyword Search to AI-Powered Search: Our Journey

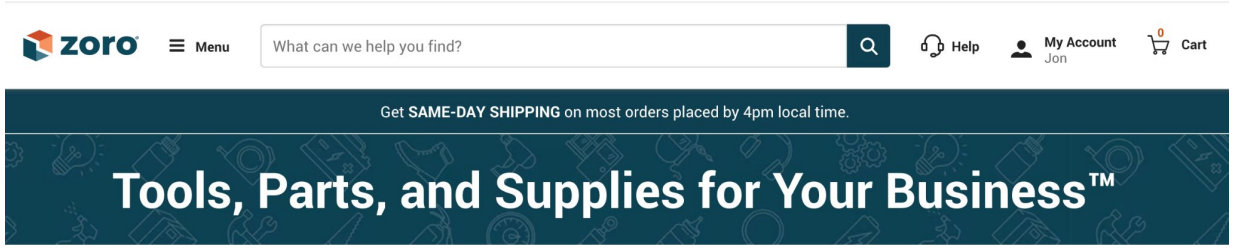
Jon Vivers, Meet Parekh, Jason Taylor

24th April 2025

zoro®

- Introduction
- Why we took search inhouse
- Understanding search traffic & measuring the search journey
- Search Infrastructure
- Offline testing
- Traditional Search
  - Browse
  - Identifier
  - Simple keyword search
- AI-powered enhancements
  - Head term reranking
  - Semantic expansion
  - Type ahead
- Advanced AI capabilities
  - LTR
  - KNN
  - Content enhancement
- Conclusion & Takeaways
- Impact

Introduction



As an online retailer specializing in industrial supplies, Zoro has a massive catalog with millions of SKUs. Helping customers quickly find the right product is critical to increasing conversions and ensuring a seamless shopping experience. Zoro leverages AI-powered search and discovery technologies to optimize this process.

Our product catalog currently exceeds 14M skus across 30 verticals

We are experiencing rapid growth.

Why we took Search In-House

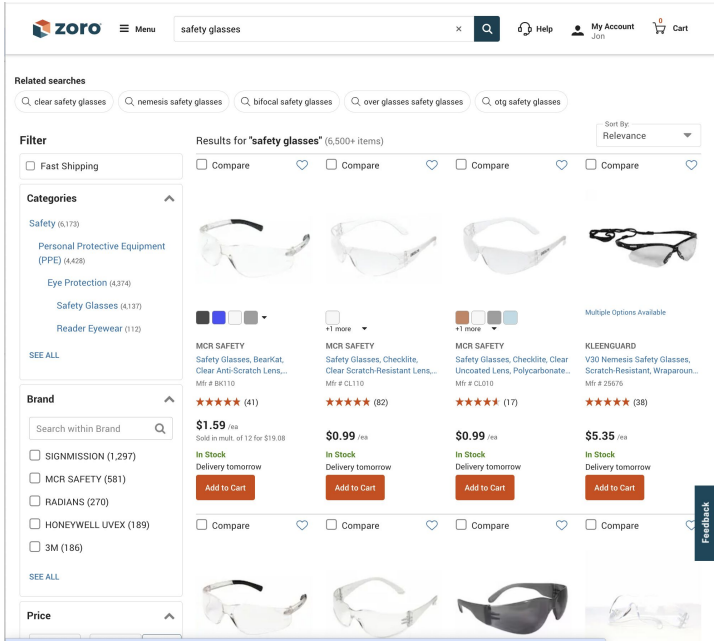
*Transitioning from a **third-party search provider** to an **in-house search solution** was a strategic decision driven by the need for **greater control, flexibility, scale and innovation** in search relevance and user experience.*

Limitations of Third-Party Search Providers:

- **Lack of control over relevance ranking** – Couldn't fine-tune search results for different customer needs.
- **Limited ability to innovate** – Feature development was dependent on the vendor's roadmap.
- **Generic algorithms** – Not optimized for identifier-heavy and B2B search behaviors.
- **Data insights locked away** – Couldn't fully leverage user behavior data for relevance improvements.
- **Inability to scale to larger catalog sizes** - Most 3rd party providers were unable / inexperienced with large catalogs

Why an In-House Solution?

- **Full control over ranking & relevance** – Adjust search ranking based on business priorities.
- **Customization for our specific use case** – Support identifier search, typeahead, and AI-driven personalization.
- **Ability to integrate AI & ML models** – Reranking, semantic expansion, LTR, and KNN search.
- **Leverage our own behavioral data** – Optimize based on real customer interactions.
- **Cost efficiency in the long run** – Reduce reliance on expensive third-party search services.

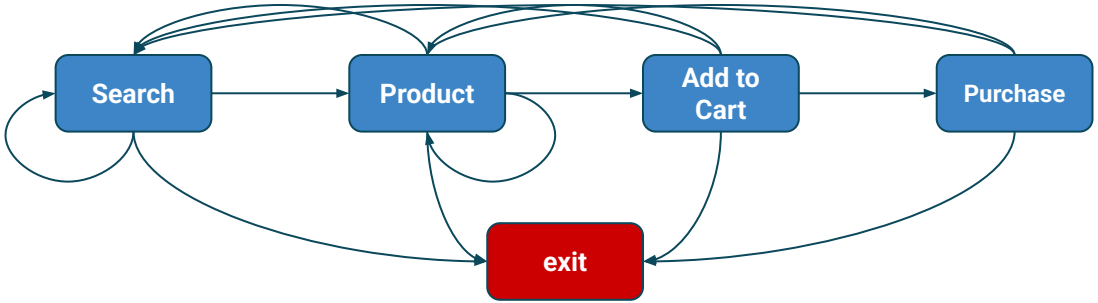


Understanding Search Traffic & Measuring the Search Journey

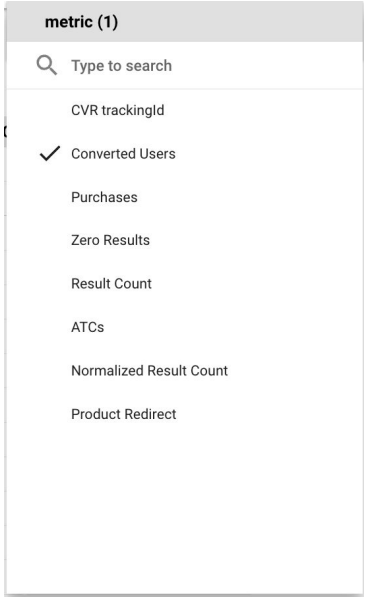
To **optimize search performance**, we must first **understand user behavior**. This requires **segmenting search traffic** and **capturing key data points** at every stage of the search journey.

Understanding Search Traffic

While shoppers rarely follow a straight path from search to purchase, they often fall into identifiable behavior loops. By analyzing these recurring patterns — like bouncing between search and product pages or abandoning carts — we can optimize the search experience to reduce friction, guide intent, and increase conversions.

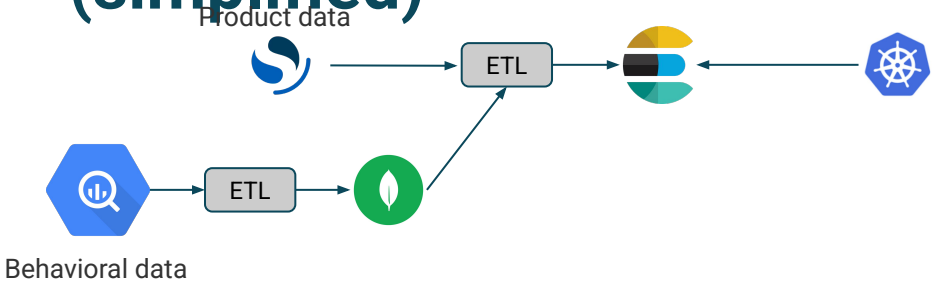


Query Segmentation



Search Infrastructure (simplified)

Where we started:



Elasticsearch Backbone

- Chosen for its flexibility, performance, and ecosystem support
- Powers all core search functionality: keyword search, KNN, ranking

Modular Indexing Pipelines

- Ingest product data from multiple sources (PIM, CMS, ERP)
- Normalize, enrich, and structure content before indexing
- Enables fast reindexing for experimentation & feature toggles

Query Processing Layer

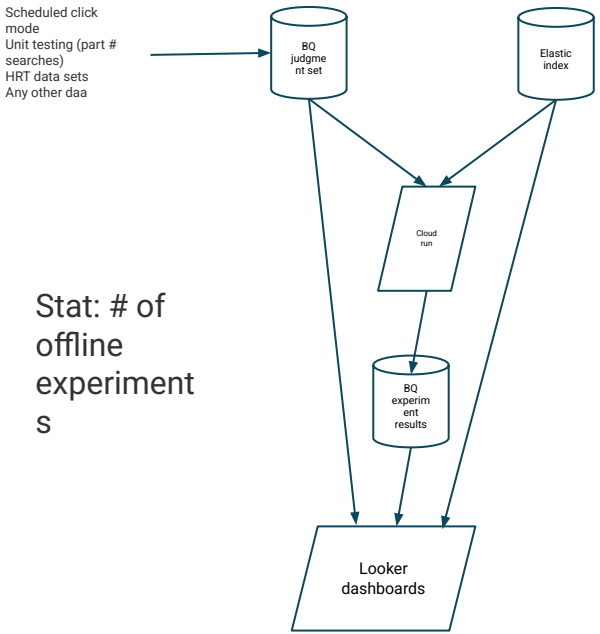
- Custom middleware parses and routes queries based on intent
- Injects context like ranking signals, feature flags, test legs

Where we are now:

ELT v's ETL

- By taking an 'ELT' approach we were able to fully utilize the power of our data lake and BigQuery in order to simplify and streamline adding features to our index
- Consolidating to elasticSearch as a feature store (in addition to a query engine) standardized our production processes

Offline testing



Stat: # of offline experiments

What Is Offline Relevance Testing?

- A controlled, reproducible method to evaluate search performance using a fixed dataset of queries and labeled relevance judgments.
- Lets us test ranking changes, model updates, and query rewrites in isolation.
- Complements online A/B testing by offering faster iteration and lower risk.

Key Inputs for Offline Testing

- Query Set: A representative sample of real customer searches.
- Judgments (Labels): Relevance scores for query–document pairs, gathered via:
  - Human annotation
  - Click/engagement-based heuristics (e.g., position-normalized clicks)
- Ranking Outputs: Results from baseline and candidate models for side-by-side comparison.

Common Evaluation Metrics

- NDCG (Normalized Discounted Cumulative Gain): Rewards placing relevant items higher.
- Precision @ K / Recall @ K: Measures relevance coverage in the top K results.
- MRR (Mean Reciprocal Rank): Highlights how quickly the first relevant result is shown.

Why It's Valuable

- Faster iteration cycles than A/B testing
- Safe testing ground for experimental models
- Better targeting of relevance improvements
- Useful for training and validating Learning to Rank (LTR) models

Best Practices

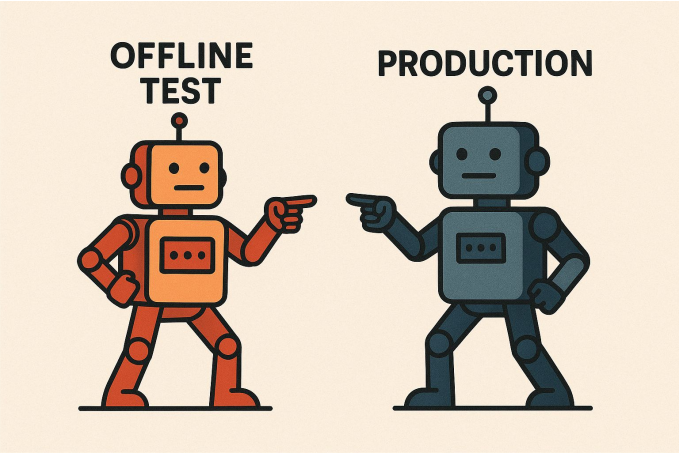
- Keep query sets diverse and updated regularly.
- Use a mix of critical, head, and long-tail queries.
- Make sure your judgement sets reflect your search segments
- Balance human-labeled and implicit feedback-derived relevance judgments.
- Pair offline results with online validation (A/B or shadow testing) before deployment.

Offline Testing

Offline testing is a reproducible, low-risk method for evaluating the relevance of search results using a fixed set of queries and labeled judgments—without needing live user traffic

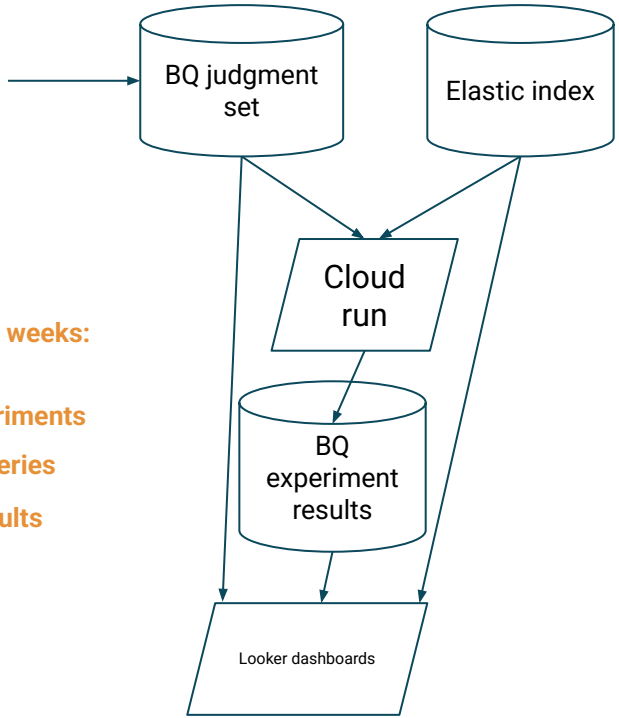
Key requirements of offline testing

- Directional consistency with A/B test results in production.
- Diagnostic, it must help uncover opportunities
- Repeatable
- Observable



Offline Testing: Ingredients

Scheduled click model
Unit testing (part # searches)
HRT data sets
Any other data



This last two weeks:

72 experiments

550k queries

19M results

1. Judgements

- Click ratings - Update on fixed cadence.
- Human annotation
- Synthetic data
- LLM as Judges etc.

2. Judgement Sets

- Targeting particular strata (e.g. head terms, brand queries, identifier queries etc.)
- Mixed - replicating proportion we find in production is best way to replicate results in offline testing.**
- Update Judgement Sets regularly**

3. Evaluation

- Metrics: nDCG, Precision, Recall, MRR, ERR
- Evaluation Views

Offline Testing: Evaluation Views

Bigger Picture

type

search_volume_group

Zero Result

token_count

judgement_set_name: mixed_quer... (1)

Experiment Results

judgement_set_name	experiment_id	Avg Results	Avg Query Time	Record Count	Zero Results	No Rated Resu...	nDCG
mixed_query_set	0107d - roduct-search-1	0	71.51	4,582	75	238	0.6458
	14dfa - roduct-search-2	0	188.32	4,582	75	282	0.6137

Comparison View

search_term: 10' air hose (1)

	search_term	rank	live_product_search_1	live_product_search_2
1.	10' air hose	1	0.034926882275433759 G1543771 3/8" x 500 ft PVC Bulk Multipurpose Air Hose 300 psi RD	0 G3899741 Air Hose,1/2"I.D.,10 ft.
2.	10' air hose	2	0 G5240676 3/8" x 50 ft PVC Coupled Multipurpose Air Hose 300 psi, Red	0.025294580822949237 G4661140 Air Hose Reel with 3/8" x 50Ft w/ Rubber Hose
3.	10' air hose	3	0.036737813646661496 G3478851 3/8" x 250 ft EPDM Bulk Multipurpose Air Hose 200 psi BK	0 G2867955 Air Hose 1" x 50 Ft. Coupled 200 psi
4.	10' air hose	4	0.034926882275433759 G904186920 3/8" X 50' Automatic Air Hose Reel	0 G502661522 $3/8" 50' Flexzilla Air Hose Assembly
5.	10' air hose	5	0.031588778471649981 G1497291 Coiled Air Hose, 1/4 in Hose Inside Dia., Blue, Brass x Brass, MNPT x MNPT, 15 ft Hose Length	0 G3902690 Air Hose,1/4"I.D.,10 ft.
6.	10' air hose	6	0.031588778471649981	0.031588778471649981

Fine Grained View


```
    "productId": [
      "G510434590"
    ]
  },
  "_explanation": {
    "value": 10.164468,
    "description": "sum of:",
    "details": [
      {
        "value": 0.16446793,
        "description": "rescored using LTR model flatte
        "details": [
          {
            "value": 7.953125,
            "description": "first pass query score",
            "details": [
              {
```

```
    "productId": [
      "G301839184"
    ]
  },
  "_explanation": {
    "value": 10.616099,
    "description": "sum of:",
    "details": [
      {
        "value": 0.61609924,
        "description": "product of:",
        "details": [
          {
            "value": 10.6875,
            "description": "first pass query score",
            "details": [
```

Offline Testing: Value

1. Rapid Feedback Loops:

- Allows quicker evaluation of new models
- Supports fast iteration, reducing time between idea and impact

2. Deep Dive Diagnostics

- Helps capture finer nuances in model.
- Side-by-side comparisons surface specific strengths or weaknesses

3. Isolation of Problems

- Enables creation of targeted judgment sets.
- Isolate performance across query types.

4. Less Expensive than A/B Testing

- Doesn't require traffic splits or statistical wait times.
- Valuable for early stage model testing or validating improvements.

Offline Testing: Impact

- Get winning models to production faster
- Capture deeper insights into model performance
- Drive continuous, confident improvements in search relevance

Traditional Keyword Search – Our Starting Point

We began our journey with a traditional keyword-based search system – a solid but limited start.

🔍 How It Worked

- Lexical matching (exact/partial)
- Ranked by BM25 + product popularity
- Deterministic results

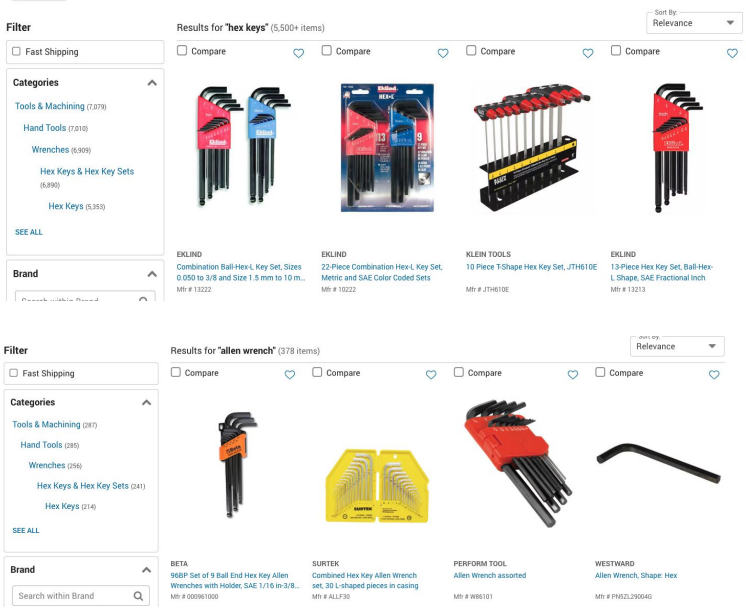
✅ What It Did Well

- Great for SKUs, MPNs (Our bread and butter)
- High return on effort: easy to implement, fast, predictable, and easy to debug

⚠️ Where It Fell Short

- No synonym/intent understanding
- No learning from behavior
- Fragile — typos = zero results

Impact: parity with 3rd party provider



Keyword approach

📄 Why It Matters - These foundational techniques gave us robust recall and cleaner inputs — a critical launchpad for layering in AI.

Analyzers (Understanding our most frequent search patterns)

- Min-length filters to reduce false part number matches
- Stop tokens to suppress brand stuffing in part numbers
- Regex-based measurement normalization (e.g., "10-ft" <-> "10")

Tf/Field Norms (Optimizing scoring for our unique index content)

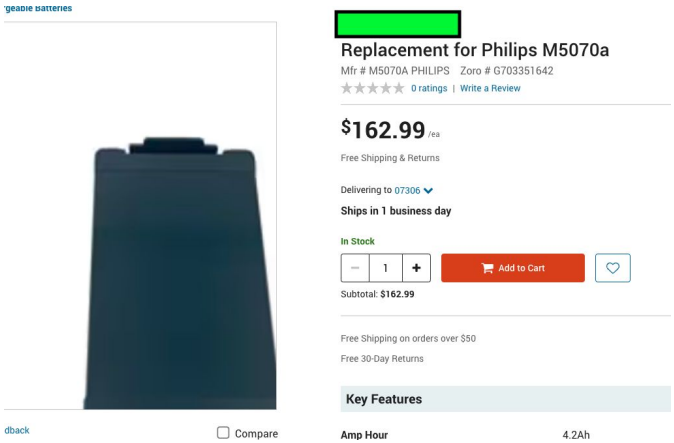
- TF / IDF with norms is biased towards shorter text.

Separating Ranking and Recall (Decoupling our Query Plan)

- This helped in isolating recall vs precision problem for us.

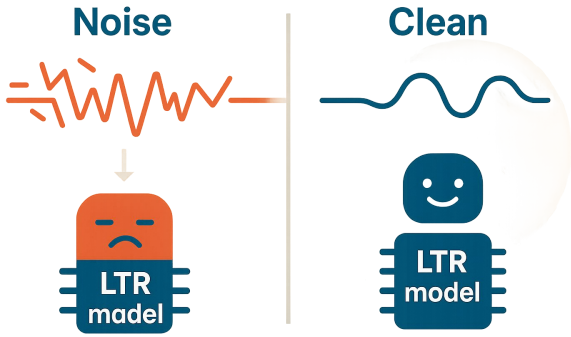
10 ft air hose -> 10-235-523

Milwaukee 0882 -> Milwaukee 0882-20



Keyword Search: Impact

- Removes Noise from Signals (better for training models)
- Less results (with same Recall) = Faster LTR + Vector Search models



AI-Powered enhancements

AI is transforming search from **static keyword matching** to a **dynamic, intelligent system** that **understands intent, reranks results, and expands queries** for better recall and relevance.

Semantic Expansion with Synonyms

Using a combination of subject matter expertise from our Merchandisers and observed search behavior, we can productionize synonyms in a targeted manner to blunt the impact of common search errors

Start simple

- Find all queries that just differs in space or one character.
- Triage with SMEs to solicit reported errors or industry jargon

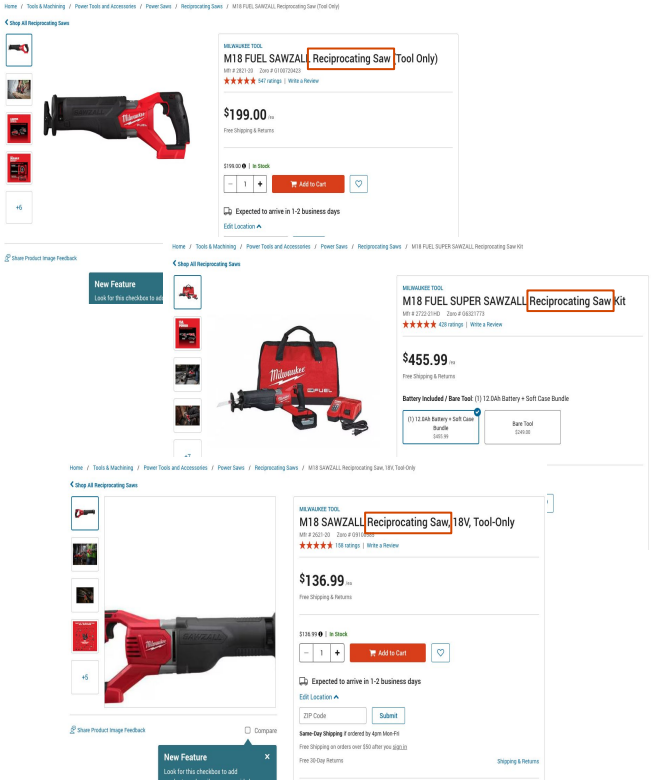
Sequential Searches (capturing patterns)

- Subsequent searches within same session differing in 1 character

Semantic Expansion with Synonyms

More advanced models (cart query expansion)

Demolition saw
Reciprocating saw
Handheld saw
Power saw
Sawzall
Cordless saw
Electric saw
Sabre saw
Recip saw

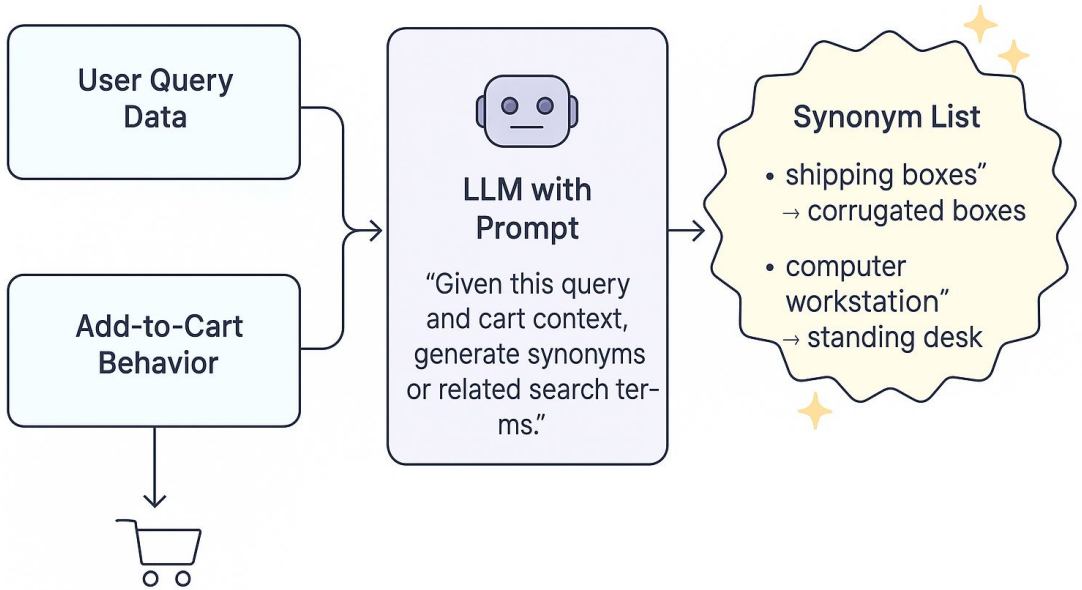


zoro

Semantic Expansion with Synonyms

More advanced models - generating synonyms with LLMs

These simple techniques generated over 11k synonyms and reduced our zero result rate by 40%



Term Association Models (feedback loops)

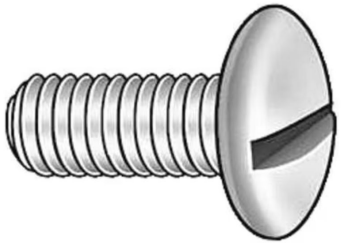
Using past success to bootstrap future success

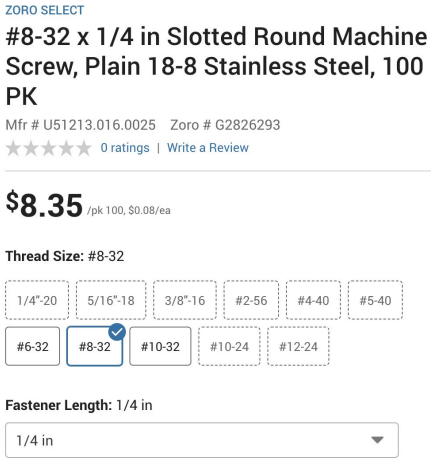
Term Association Model Usage

- We observe how users search for our products, which terms lead to conversions and which terms don't
- Put our thumbs on the scale and boost or deboost products based on successful or unsuccessful searches

Continuous Retraining of Association Models (Create feedback loops)

- As our customers search more, our models learn stronger positive and negative associations with products



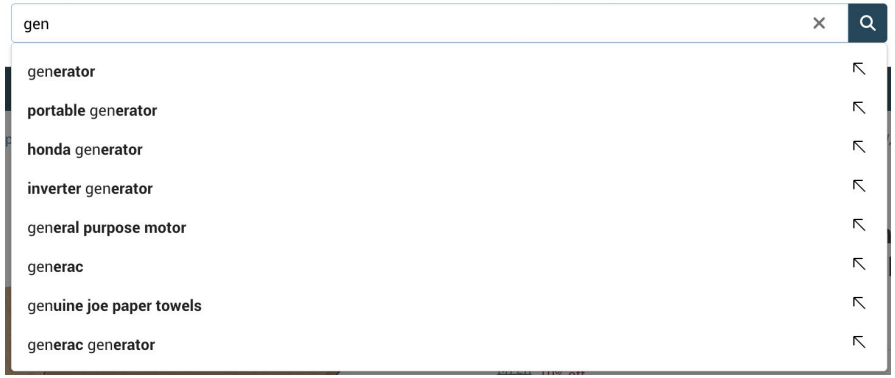


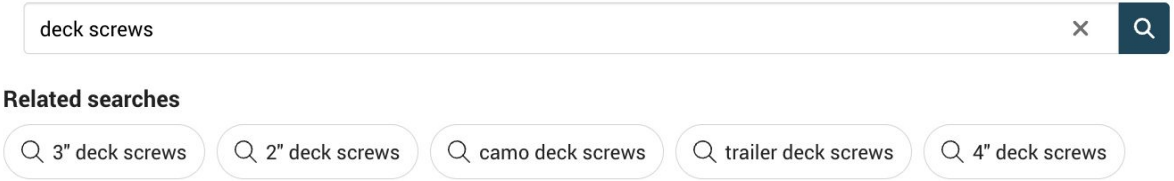
Positive: 8-32 machine screw
Negative: screws

Type Ahead / Related Searches

Now that we've identified keywords as a core entity in how users express intent, we can build systems that learn from successful search behavior to guide future searches.

- Elevate keyword suggestions that consistently convert
- Promote intent-rich, high-performing terms
- Break complex searches into modular, reusable keyword chunks
- Merge similar or redundant queries to reduce noise
- Ensure clean, consistent typeahead experiences





Advanced AI Capabilities in Search

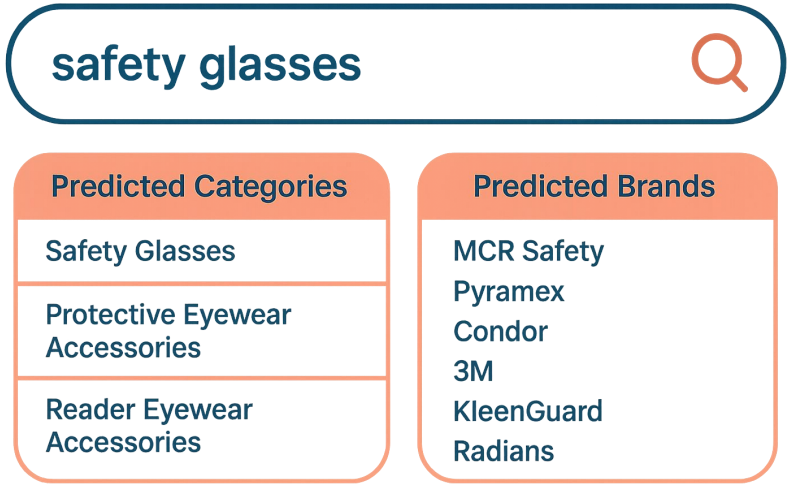
After laying the groundwork with basic AI enhancements, we expanded into more advanced, machine learning–driven techniques to further improve relevance, recall, and user experience.

Query Understanding

Developing models to find broader associations from query strings to other targets

Building General Associations

- Transformer Models (BERT) to find semantic associations between search terms and key product attributes like brands or categories
- Use to enrich queries or propagate model predictions to the frontend to power smart faceting



Learn To Rank

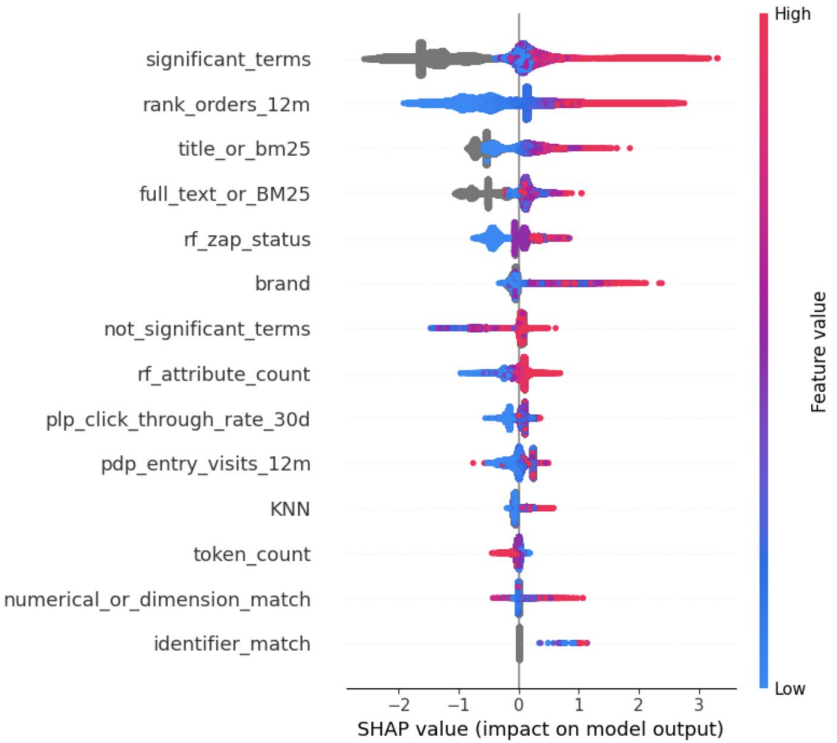
Why it matters to us:

- Semantic and lexical signals alone aren't good enough
- Signal explosion (we have a lot of ranking signals)
- Ranking signals are strong
- Interpretable/explainable features

How we use it:

- Trained on click model targets
- Includes query features, document features, and query-document features
- Continuously retrained to adapt to changing behavior

Impact: Significant lift in click-through rate and engagement on top-ranked results.

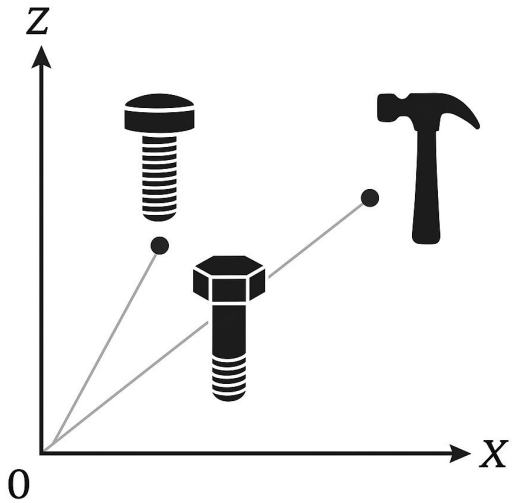


kNN - Vector Search

Why it matters: Our customers don't always use precise language and learned synonyms have a cold start problem—vector search helps find semantically relevant results, not just textual matches, using kNN score allows us to inject semantics into ranking features

How we use it:

- Bi-encoder configuration: embed queries and product representations in same vector space
- ANN search to find products similar to the query
- Used as supplement to keyword search for recall and as a ranking signal in LTR



Impact: Increased recall and relevance for vague, long-tail, or many-token natural language queries

Content Enrichment Using LLMs – Fueling Better Matching

What it is: Using large language models (LLMs), to enhance product content for improved discoverability and search experience.

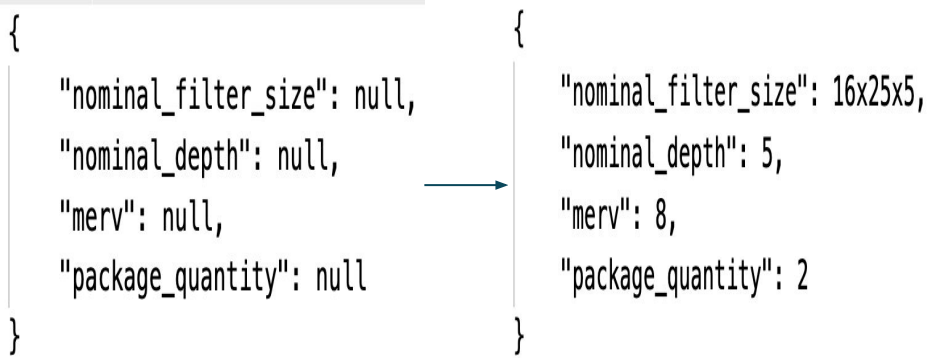
Why it matters: Better data = better search relevance, better filters, better user experience.

How we use it:

- Normalize inconsistent product data
- Enhance titles and descriptions for better search term match
- Fill in missing metadata to enable better filtering and retrieval

Impact

1. Improved Faceting - 250 bps increase in facet usage
2. Better Recall



What We've Learned from a Year of Improving Search

🛡️ **1. Bot Traffic Distorts Everything**
Bots can heavily skew behavioral data, and it's more than a reporting problem. If you're not filtering them out, you're optimizing for fake users. Invest in strategies that let you focus on real customer behavior.

🧰 **2. Simplify Your Stack**
The best search teams focus on relevance, not racking up infrastructure complexity. Streamlining lets you iterate faster and stay focused on what actually matters.

⚡ **3. Quick Wins Matter**
Sometimes a simple solution really is good enough. Taking the easy wins early helps reduce scope, deliver immediate value, and build momentum.

📈 **4. Start Simple, Add Complexity Later**
Early solutions don't need to be perfect — they need to teach you something. Simpler approaches get results and deepen your team's understanding of the problem space.

🧪 **5. Offline Testing Is a Superpower**
Online testing time is precious. Build a strong offline testing pipeline so you can experiment fast and save your online cycles for the changes that really move the needle.

🚨 **6. But Don't Over-Rely on Offline Tests**
Offline testing isn't gospel. Click-based judgment sets only work if the right results are visible to begin with. Sometimes a test fails offline — and still wins with customers. Be ready to trust your instincts and ship anyway.

🔁 **7. Test Everything**
No exceptions. If it touches the customer, it gets tested.

Impact