The Journey to

Semantic Search in Omnichannel Retail

at dm-drogerie markt

MICES 2026 · Berlin
June 10, 2026 · w3.hub, Berlin · dm-drogerie markt

dm TECH

Speaker

Denise Schäfer

Fullstack Software Engineer

5 years part of search team for dm online shop & app

Research lead & technical architect for dm search

25+ years in software development & project management

Search Team
dm-drogerie markt

dm TECH

Speaker

Mike Dirnberger

Backend Software Engineer

5 years part of search team for dm online shop & app

Focus on search (design, implement, optimize)

15+ years in software development

divae CONCLUSION

dm TECH

Talk Overview

What We'll Cover

01 Keyword vs. Semantic Search

02 Technical Stack & Quality Assurance

03 5 Iterations to Production

04 Semantic Search Use Cases at dm

at a Glance

Germany's #1 drugstore chain — setting the standard in health, beauty and sustainability since 1973.

Our Search Challenge

- **22k+** products to discover across categories
- Highly diverse customer vocabulary & phrasing
- Complex attribute filtering (e.g. sugar-free)
- Zero-result queries hurting conversion rates
- Chatbot & AI assistant integrations

4,200+
Stores in Europe

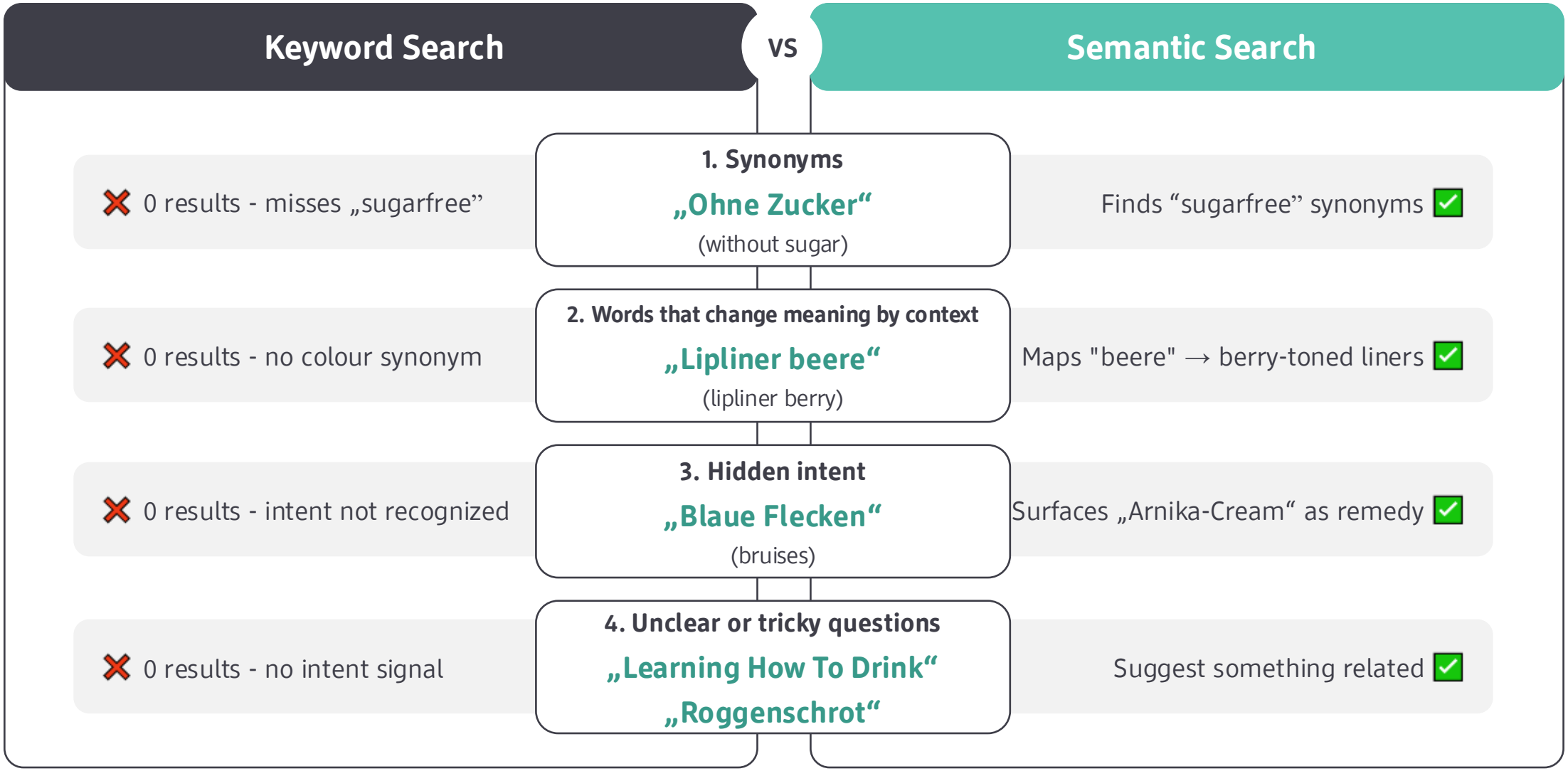
14
Countries
in Europe

22k+
Products online
in Germany

2M
Searches per day

dm.de & dm App

The Search Gap: **Keywords vs. Semantics**



Technical Foundation: **E5 + Qdrant**

Stack Details

Embedding Model

intfloat/multilingual-e5-small — 384 dim, multilingual, ONNX-exported for fast JVM inference via ONNX Runtime

Vector Database

Qdrant — high-performance ANN search with payload filters, named vectors

Service Stack

Kotlin / Spring Boot — reactive, production-ready; p95 latency <50ms for semantic retrieval end-to-end

Evaluation

Offline tests · LLM-assisted checks · A/B tests – see slide 7

Quality Assurance: **How We Measure Progress**

01
Offline Metrics

- nDCG@10, nDCG@30
- judgements from click-stream: clicks + add-to-carts + orders -> relevance score
- position bias correction
- run on ranking or embedding changes

02
LLM-Assisted Relevance

- uses production queries
- LLM judges query-result pairs at scale
- provides granular relevance ratings with explanations

03
A/B Testing

- Controlled traffic splits on live queries
- Click-through rate (CTR) tracking
- Add-to-cart & conversion measurement
- zero result rate
- Long-term relevance trend analysis

All three pillars essential to validate ranking changes — offline metrics and LLM on pre-production, A/B tests validate business impact

Five Iterations to Production-Ready Semantic Search

01

Guardrails

Score Cutoff
+ Category restriction

02

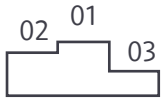
Business Signals

Relevance ranges
+ Top Sellers

03

Brand & Attributes

Brand Boost
+ Native Filters



04

Re-Ranking

Cross-Encoder Reranker + Score Filter

05

Fine-Tuning

Bi-Encoder trained on clickstream data

Iteration 1 **Guardrails: Score Cutoff + Category Restriction**

Problem: **Overly Creative Results**

Without guardrails, the bi-encoder retrieves anything semantically similar — even if it's the wrong category or product type.

”entwirrungshaarbürste” (= detangling brush)
Also returns other hair products

”roggenschrot” (= coarse rye)
Results include other products from categories unrelated to food

Nonsense query
Returns products anyway — hurts brand trust

Solution: **Two Guardrails**

Score Cutoff

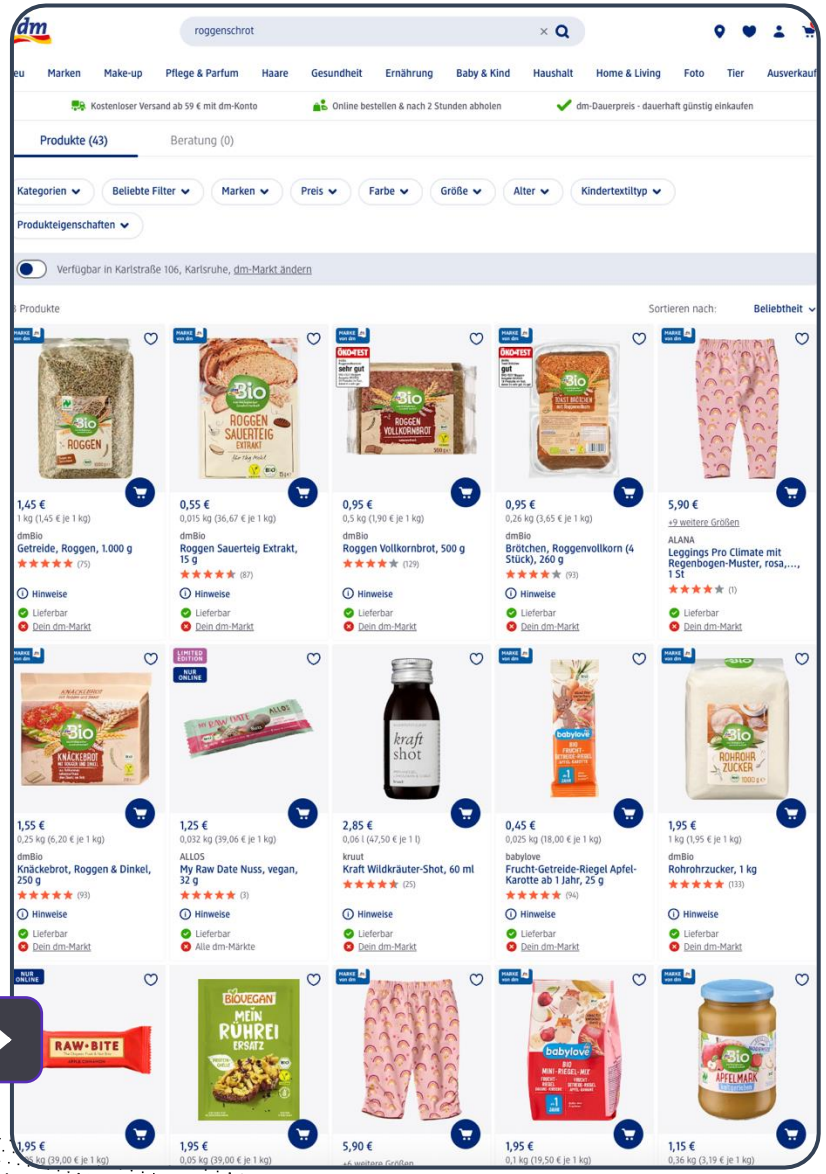
1. Define minimum cosine similarity threshold.
If no result exceeds the threshold
→ **rather show zero-result page**
2. Detect drops of similarity scores within results: find largest drop (greater than a drop threshold value) than and cut off after that

Category Restriction

Prerequisite: Hits should belong to max 1 or 2 top level product categories
→ **restrict results to those categories**

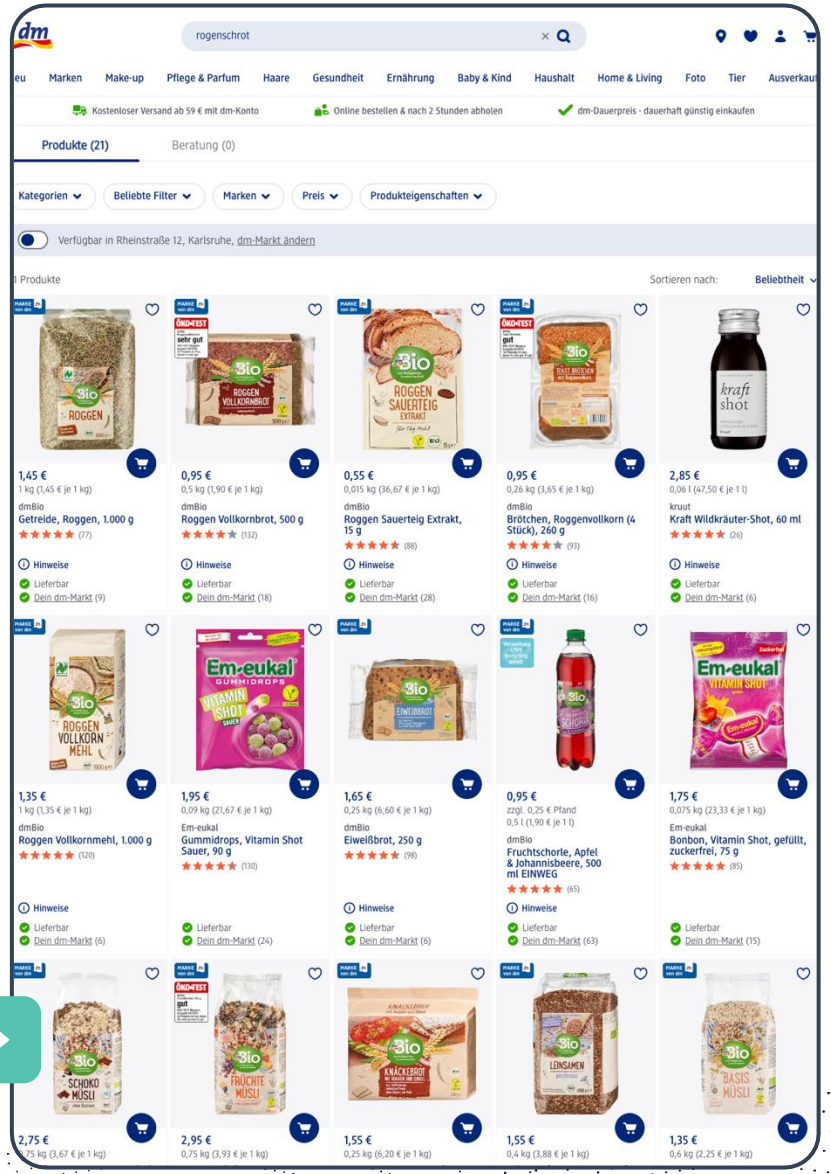
Example: ”roggenschrot" → top hits are from food category → restrict to that, excluding irrelevant cross-category hits.

Iteration 1 **Guardrails: Category Restriction Example**



Before →

After →



Iteration 2 **Integrating Business Signals**

“Show relevant products — put the bestsellers at the top of each relevance group”

Ranking Algorithm

Score cutoff and category restrictions are applied before ranking

Score Grouping

threshold: 0.005

Semantic search finds relevant products,
then groups them by similarity (threshold: 0.005)

Within-Group Order

sales quantity

Inside each group,
products with higher sales count rank first

Purchasability

binary flag

Purchasable products always surface
before non-purchasable ones

Iteration 2 Integrating Business Signals: Offline Test Results

Relevance-only ranking vs. Relevance + Business Signals

Metric*	Relevance only	Relevance + Business Signals
Clicks + Add-to-Carts @10	72.7M	**78.1M (+7.4%)**
Clicks + Add-to-Carts @30	141.2M	**144.0M (+2.0%)**
nDCG@10 / nDCG@30	0.679 / 0.760	**0.694 / 0.771 (+2.2% / +1.5%)**

**Adding business signals improves all metrics —
both engagement and ranking quality improve together.**

* Clicks + Add-to-Carts = SUM across all top ~10,000 test queries (~1 month of traffic, total business impact)
| nDCG = AVERAGE across all queries (typical ranking quality)

Iteration 3 **Attribute-Filter & Brand Boost & Category Filter**

Qdrant Native
Attribute Filtering

Extract product attributes BEFORE embedding → apply attribute as native filter

"sulfatfreies Shampoo" (sulfate-free shampoo)

→ extract sulfate-free and search for shampoo, with a filter applied

Post-Search
Brand Boosting

User searches for a brand, all products from that brand are boosted in the results

"babylove Zahnhilfe" (babylove dental aid)

Only explicit brand names

Post-Search
Category Filtering

Stage 1 **Category Detection**

"duschgel vegan" (vegan shower gel) → detects "Duschgel" → result is filtered to that category

Multi-word queries only — falls back to Stage 2 if no category found

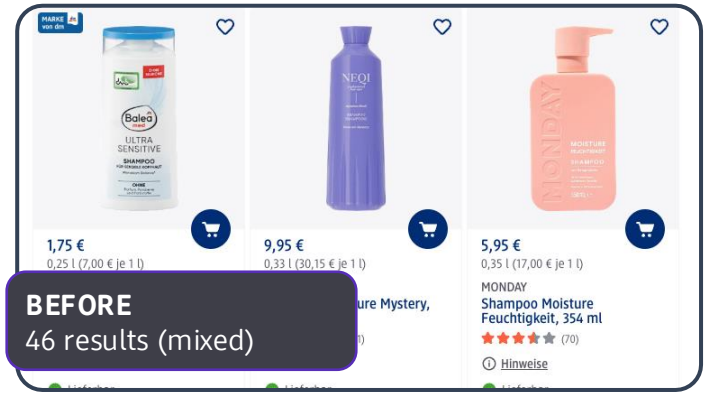
Stage 2 **Root Category Restriction**

"vegan bio" → no specific category → analyzes top results → filters to 2 root categories

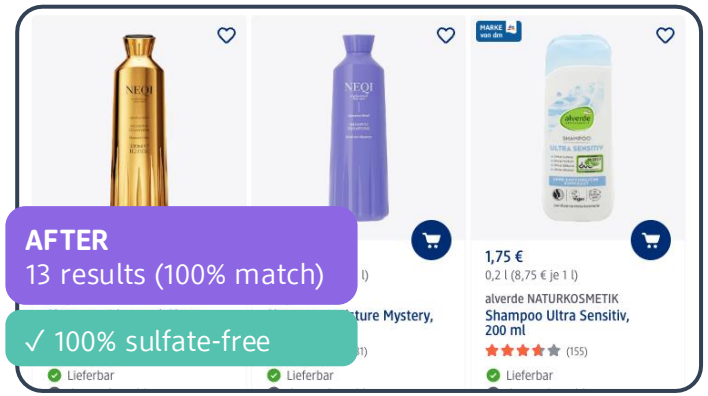
Iteration 3

Attribute Filtering
-71% Products (46 → 13)

Query: "shampoo ohne sulfat“
(shampoo without sulfate)



BEFORE
46 results (mixed)

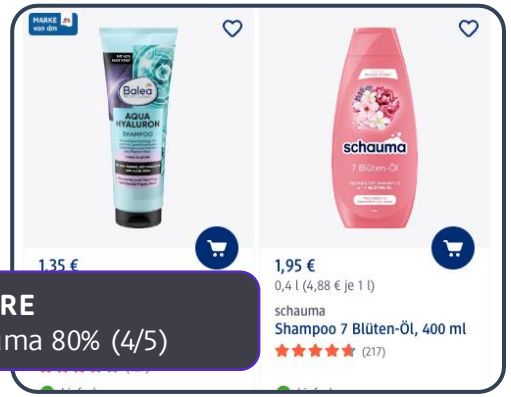


AFTER
13 results (100% match)

✓ 100% sulfate-free

Brand Boosting
+30% Score Boost

Query: "schauma trockenes haar shampoo“
(shauma dry hair shampoo)



BEFORE
schauma 80% (4/5)

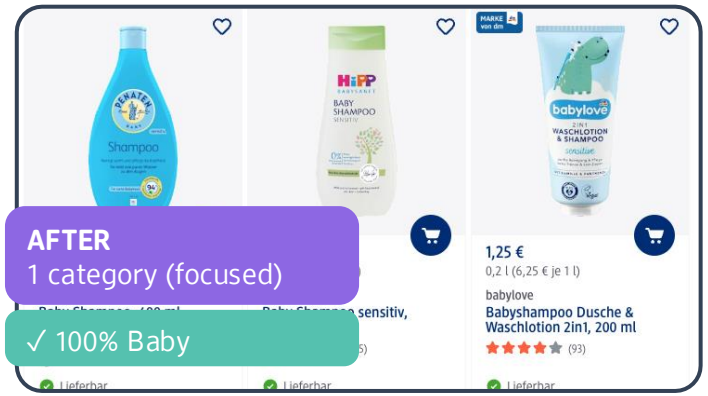
AFTER
schauma 100% (5/5)

✗ Balea PROFESSIONAL eliminated

Category Filtering
-67% Categories (3 → 1)

Query: "baby shampoo"

BEFORE
3 categories (mixed)



AFTER
1 category (focused)

✓ 100% Baby

Iteration 3 **Attribute-Filter & Brand Boost & Category Filter**

① **Offline Test:** top 10,000 search queries (1 month)

Metric*	Without improvements	With improvements	Change
Clicks + Add-to-Carts @10	63.2M	67.0M	**+6.0%**
Clicks + Add-to-Carts @30	112.8M	120.7M	**+6.9%**
nDCG@10 / nDCG@30	0.577 / 0.675	0.653 / 0.718	**+13.2% / +6.3%**

② **LLM-based Testing** — Chatbot Queries – top query improvements

Attribute Filter

"glutenfrei Haferflocken"

+1.2 Graded Precision**

isGlutenFree=true detected

Brand Boost

"dmBio Kidneybohnen Dose"

+1.5 Graded Precision

brand 'dmBio' detected

Category Filter

"Lipliner beere"

+1.3 Graded Precision

'Lipliner' category detected

* Clicks + Add-to-Carts = SUM across all ~10,000 test queries (~1 month of traffic) | nDCG = ranking quality, AVERAGE across all queries (LLM-evaluated)
** Graded Precision: results get a relevance grade on a scale — e.g. 0, 0.5, 2 — and the metric averages those grades across the result set

Iteration 4 Re-Ranking: Two-Stage Pipeline with Cross-Encoder

Still failing for many examples: "feuerzeug", "linsenwaffeln" — bi-encoder returns too many irrelevant results, lacks precision

Pipeline:

Bi-Encoder (E5)

Encodes query & products independently
Fast ANN search in Qdrant (384-dim)
Returns **candidates**
Scales to millions of products
Limitation: No query-doc interaction

→

Cross-Encoder Reranker

Jointly encodes query + product text
Outputs new relevance score per pair
Reranks **Top-K** (e.g. Top-60)
Limitation: does not scale for large results

→

Score Filter

Use score from reranker to remove irrelevant results or return zero results (safety net for items not stocked at dm)

Model Candidates (constraint: non-commercial, ONNX, fast)

- cross-encoder/ms-marco-MiniLM-L6-v2
- cross-encoder/msmarco-MiniLM-L6-en-de-v1 (+ German)
- mixedbread-ai/mxbai-rerank-xsmall-v1

What We Learned

- Adds latency cost (try to restrict to 64 or 128 tokens)
- Fields to encode: title + category not enough, more context needed
- Should be fine tuned to own domain / product catalog
- Only for Top-K in small result sets (e.g. for zero results fallback)

Iteration 5 **Fine-Tuning E5**

Clickstream → Pairs + Grades (0–3) → **MNR Training** → **ONNX**

Training Data — Relevance Grades (from clickstream)

Grade 3 Perfect match (top ~33% engagement)
Grade 2 Good match (mid ~33%)
Grade 1 Relevant (bottom ~33% — critical for recall)
Grade 0 Not relevant (0–2 clicks, never engaged)

Grades are relative per query — engagement normalized to the top product

Why Fine-Tuning?

Domain-Specific
Understands dm taxonomy and customer language

No extra latency at inference time
Encoding happens anyway — no additional cost

MultipleNegativesRankingLoss (MNR Loss)

The model learns to rank the right product higher than all other products in the batch.
Maximize similarity to the correct product; all other products in the batch serve as negatives automatically.
Effective batch size 32

Iteration 5 **Fine-Tuning – A surprising discovery**

All 4 models saw the same **241,765 positive examples** — only the **Grade 0 negatives** (irrelevant products) differed.

Model	Grade 0 negatives	What We Expected	nDCG@10	nDCG@30	Clicks+A2C @10	Clicks+A2C @30	Results returned
Full	ALL (100%)	*"Too noisy" hurts quality*	+18% ✅	+13% ✅	+16.5% ✅	−2.3% ❌	20 avg
Positives-Only	0% (removed)	*"Perfect quality"*	−12.4% ❌	−9.4% ❌	−11.8% ❌	+1.9% ✅	44 avg
Balanced	Random (30%)	*"Cleaner is better"*	+2%	+2%	−1.6% ❌	+1.0%	44 avg
Smart	Hard negs* (45%)	*"Hard negs help"*	+2%	+2%	−0.8% ❌	+1.5% ✅	45 avg

The Puzzle: How do negative examples help?

Our best guesses:

- MNR is robust to noise (contrastive learning property)
- Low-quality pairs create diverse in-batch negatives
- Weak signals (0–2 clicks) still contain information

What We Learned

- Full dataset (302K incl. Grade 0) wins
- More data > Cleaner data (for contrastive learning)
- Don't over-optimize at training time

Full model + Adaptive Cutoff · best nDCG · 83% of clicks @10 · Adaptive Cutoff fixes result count

*Hard negs = confusing cases e.g. "essence mascara" → "essence cleaning"

Semantic Search Use Cases at dm

From zero results to full semantic understanding

Phase 00
Analysis

Baseline analysis of low-performer & zero-result queries. Understanding where keyword search fails and sizing the opportunity.

▲ +20% Interaction Rate on Low Performer Queries

Phase 01
Zero-Result Rescue

Semantic search activates as fallback when keyword search returns zero results.

Phase 02
Chatbot & MCP

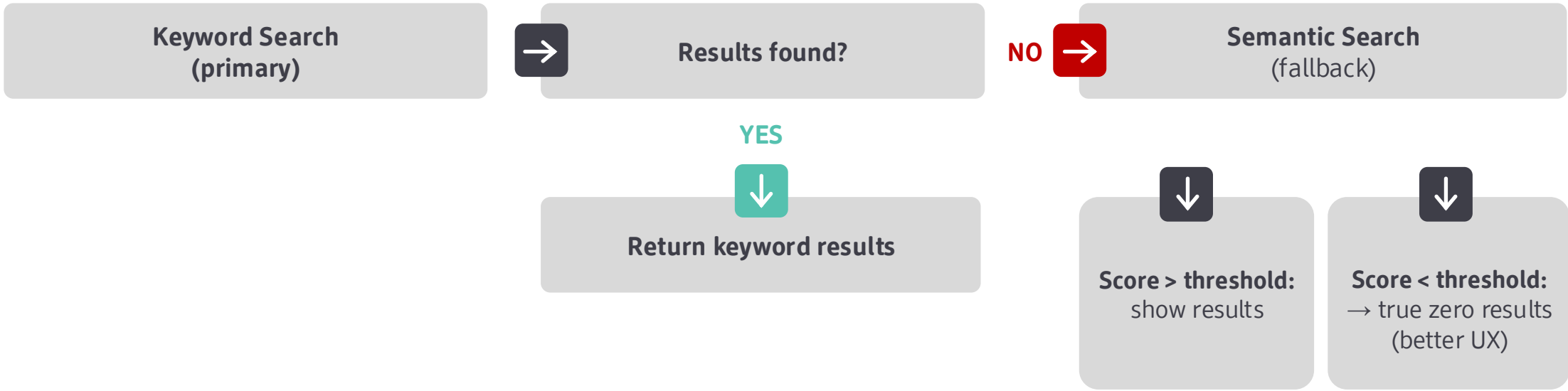
Full semantic integration into dm chatbot and external MCP server. Multi-stage intent detection for natural language queries.

Phase 03
Hybrid Search

Planned: combine keyword and vector search results to further strengthen low-performer queries and difficult edge cases.

Phase 1 **Zero-Result Query Handling**

Query Routing Logic



Business Impact

↓ Zero-Result Rate

Semantic fallback rescues queries that would otherwise show an empty page

↑ Conversion

Customers find relevant alternatives they wouldn't have discovered via keyword search

✓ Safe by Design

Score cutoff ensures we never show irrelevant results just to avoid an empty page

Phase 2 **Chatbot & MCP: Multi-Stage Intent Detection**

Multi-Stage Intent Detection Pipeline

01 **User Input**
Welches Shampoo ist gut für trockene Haare ohne Sulfat?"

02 **Intent Classification (LLM)**
Product search? General advice? FAQ?

03 **Attribute Extraction**
filter: sulfate_free=true, query: "Shampoo trockene Haare"

04 **Semantic Search**
Clean query embedded → vector search with attribute filters applied

05 **Response Generation (LLM)**
results are returned to the user with natural language explanation

One Interface, Two Consumers

Same MCP tool for chatbot and external AI

Consistent search quality across all consumers

New integrations need no changes server-side

External MCP Server

dm's search service is exposed as an MCP tool — external AI assistants like Claude can query the product catalog via natural language

https://mcp.dm.de/mcp

Phase 3 Outlook: Hybrid Search — Keywords + Semantics

Keyword Search
excels at precision

Strengths:
- Exact product name matching
- SKU / EAN lookup
- Brand + product type combos
- Well-established query patterns

Challenges:
- Fails on synonyms & variants
- No context understanding
- Zero results for paraphrases

Semantic Search
excels at understanding intent

Strengths:
- Intent & meaning understanding
- Synonym & variant handling
- Natural language queries
- Long-tail coverage

Challenges:
- Less precise for exact matches
- Higher compute cost
- May over-generalize

Hybrid Search (Best of Both)
combines both strengths

Strengths:
- Best of both worlds
- Precise + semantically aware
- Low-performer query rescue
- Robust to reformulations

Challenges:
- More complex ranking logic
- Score normalization needed
- Performance

Planned: merge keyword and vector rankings — fusion strategy to be evaluated.

Thank You!

Questions & Discussion

MICES 2026 · Mix-Camp E-Commerce Search

Conference

MICES 2026 · Berlin · June 10, 2026

Team

dm-drogerie markt · Product Search

Stack

E5 · Qdrant · Kotlin

dm TECH


dm
TECH