From Keyword Search to Data Science

Search evolution at Kaufland e-commerce

Alexander Reelsen

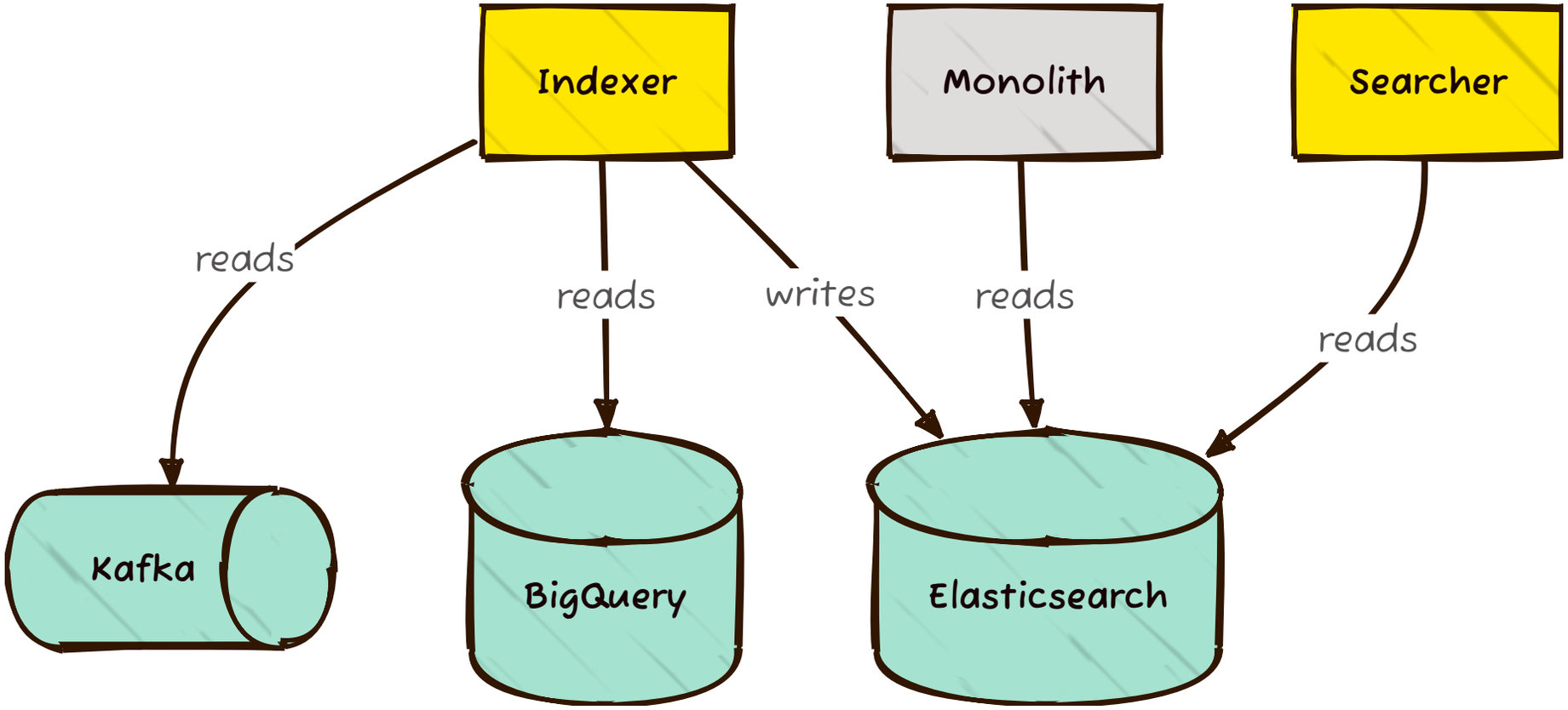
alexander.reelsen@kaufland-ecommerce.com

Introduction

Kaufland e-commerce

- Online B2C marketplace
- 11k sellers
- > 800 employees
- Five storefronts (de, cz, sk, pl, at)
- Triple digit million number of product offers stored

Architecture



Tech stack: K8s, gRPC, Typescript, Node, MongoDB, Redis, Elasticsearch, Kafka

Searches

- Phase 1: Analysis
  - Find shorter terms (agg)
  - Guess category (agg)
- Phase 2: Search
  - Product search: Find products (search)
  - Filter search: Find filters (aggs)

Phase: Analysis

- Use search learner data
- Information how users interacted with search terms and products
- Nested data structure


```
{
  "title": "Modernes Wandbild ... Kunstdruck New York",
  "search_stats": [
    {
      "id_item": 306593629,
      "rank": 0.00012203718164007487,
      "term": "modernes wandbild new york"
    }
  ]
}
```

Phase: Analysis - Category Guessing

- How do you guess a category id?
- Easy! Aggregate search on category id!
- 🚨 There are more iPhone accessoires than iPhones…
- Use search learner data, aggregate against the sum of the ranks!

Phase: Analysis - Shorter queries

- Use search learner data to find shorter queries
- `gummiseil gelb 8mm 20m`
  - What are the important parts of this query?
  - Can we extract the important parts?

Phase: Search

Next steps

- LLMs?
- RAG?
- Vector?
- Own models? Existing models?
- LTR?

Keyword search

- Easy to understand
- Fast
- Debuggable
- Users are trained to do keyword search
- Only a small amount would probably benefit from semantic search

ELSER

- Elastic Learned Sparse EncodeR
- Great idea!
- English only
- Not based on our own data
- Can we do something similar?

Query Expansion

Query Expansion

- Paper: Query Expansion by Mining User Logs
- Log based query expansion
- Correlate query terms and document terms
- Query expansion based on term correlations

Query expansion steps

- Find candidate to expand user input with
- Find a good way to integrate this in existing ES query

Query Example

user query: `nintendo switch controller grün`

- **Find longest candidates, has to include all query terms**
  - `grün tee` ❌
  - `nintendo switch bag` ❌
  - `nintendo switch controller case` ❌
  - `nintendo switch controller` ✅
  - `nintendo switch` ✅

Retrieve candidates

```
PUT query-expansion-phrases
{
  "mappings": {
    "properties": {
      "candidate": {
        "type": "text",
        "similarity" : "boolean",
        "fields": {
          "keyword" : { "type" : "keyword" }
        }
      }
    }
  }
}
```

```
POST query-expansion-phrases/_doc
{ "candidate": "nintendo switch controller" }
```

Retrieve candidates

```
GET query_expansion_phrases/_search
{
  "size": 500,
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "candidate": "nintendo switch controller grün"
          }
        }
      ],
      "filter": [
        {
          "script": { "script": { "source": ... } }
        }
      ]
    }
  }
}
```

Retrieve candidates

```
// params.query = "nintendo switch controller grün"

if (params.query.indexOf(doc['candidate.keyword'].value) < 0) {
  return false;
}

def asList = Arrays.asList(/ /.split(params.query));
def tokenizer = new StringTokenizer(doc['candidate.keyword'].value, " ");

while (tokenizer.hasMoreElements()) {
  def term = tokenizer.nextElement();
  def isTermInQuery = asList.contains(term);
  // if the query does not contain the current term, we bail
  if (!isTermInQuery) {
    return false
  }
}

return true;
```

Retrieve candidates

- Remember the scoring on the `candidate` field
- Boolean similarity: candidate with the most terms matched first

Retrieve scores

```
PUT query_expansion_cohesion_scores
{
  "mappings": {
    "properties": {
      "item_term": {
        "type": "text",
        "fields": { "keyword": { "type": "keyword" } }
      },
      "query_term": {
        "type": "text",
        "fields": { "keyword": { "type": "keyword" } }
      },
      "score": { "type": "float" }
    }
  }
}
```

Retrieve scores: Sample document

```
POST query_expansion_cohesion_scores/_doc
{
  "query_term": "nintendo switch controller",
  "item_term": "joy-con",
  "score": 0.02331638308509612 8
}
```

Note: Same `query_term` can map to different `item_term` with different scores.

Retrieve scores: Query

- Search for `query_term`
- Group by `item_term`
- Sum up scores
- Multiply summed up score with the number of item_term matches
- Finally sort by above multiplication score

Correlate query & document terms

```
FROM query_expansion_cohesion_scores |
WHERE (
  query_term.keyword LIKE "nintendo switch controller" OR
  query_term.keyword LIKE "switch controller" OR
  query_term.keyword LIKE "nintendo switch"
  ) |
STATS
  final_score=SUM(score) *
    POW(
      TO_DOUBLE(
        COUNT(item_term.keyword)
      )/3,
      0.01
    ),
  query_terms=VALUES(query_term.keyword)
  BY item_term.keyword |
SORT final_score DESC |
LIMIT 10
```

Correlate query & document terms


```
{
  "columns": [
    { "name": "final_score", "type": "double" },
    { "name": "query_terms", "type": "keyword" },
    { "name": "item_term.keyword", "type": "keyword" }
  ],
  "values": [
    [
      0.07394268180954819,
      [ "switch controller", "nintendo switch controller" ],
      "nintendo switch"
    ],
    [
      0.073837760835886,
      [ "switch controller", "nintendo switch", "nintendo switch controller" ],
      "controller"
    ],
    [
      0.05886813160032034,
      [ "switch controller", "nintendo switch", "nintendo switch controller" ],
      "joy-con"
    ],
```

Results

- ☹️ Unfortunately the model is not working well enough
- ☹️ Too many random terms returned

Next steps

- Query Rewriting
  - Semantic Equivalence of e-Commerce Queries (Mandal et al, 2023)
  - QUEEN: Neural query rewriting in e-commerce
  - Advancing query rewriting in e-commerce via shopping intent learning
- Automatic synonym generation using LLMs?

Summary

- With endless possibilities comes endless responsibility
- Knowing when to stop
- The art of measurement
- PoC ≠ MVP ≠ production
- Define acceptance criterias upfront
- Are we able to differentiate before query between keyword search and semantic search?

Thank you

Thank you

Questions? Answers! Let’s talk!

alexander.reelsen@kaufland-ecommerce.com

P.S. We’re hiring