# VECTORIZING CONSUMER ELECTRONIC GOODS

**MICES**

June 2024

# Uncovering The Problem Statement

# Problem analysis of zero-results

**SPELLING**

**11%**

Includes issues with spellings, wordbreaks and special chars that cannot be interpreted as of now (e.g. )

- *lenovo tap 11, i pad reparatur*

---

**SERIES**

**32%**

Includes all queries where customers are searching for products with certain series numbers

- Model number misspelled (*dreame l20s* ➔ *dreame l20, hisense 43e61kt* ➔ *hisense 43e6kt*)
- Specific model number/product not available (*apple pencil kappe, playstation adventskalender*)
- Product/model defined differently in MM (*john wick 1-4* ➔ *john wick*)

---

**SEMANTICS**

**24%**

Includes all queries where the search engine lacks semantic understanding

- Generic terms not appearing in product data e.g. for terms describing a size (*klein, groß*), a context (*homeoffice*), an attribute (*wireless, gebogen*)
- Different naming convention or synonyms (*bodenwaagen* ➔ *körperwaagen, falschgeldstift* ➔ *Geldscheinprüfstift*)

---

**LANGUAGE**

**6%**

Includes all queries using non-latin alphabet or a different language, e.g. French

- *робот пилосос, tondeuse (trimmer)*
- Behaviour has changed over time

---

**ASSORTMENT**

**33%**

Search is reasonable but unable to return valid products

- Either the products are not sold on MMS or lifecycle status of the product is not active
- *skyrim ps5, apple superdrive*

//

# How vector search could help

VS could solve following zero results query clusters:

- Spelling
  - lenovo tap 11 ➔ lenovo tab 11, *i pad reparatur ➔ ipad reparatur, iphone15 ➔ apple iphone 15*
- Series
  - *john wick 1-4 ➔ john wick, dreame l20s ➔ dreame l20, hisense 43e61kt ➔ hisense 43e6kt*
- Semantic
  - *seagate wireless externe festplatte ➔ seagate ~~wireless~~ externe festplatte,* vollkaffeeautomat ➔ kaffeeautomat
- Language
  - *tondeuse ➔ trimmer*

**Assortment**:

- Cannot be solved by finding semantically similar product descriptions. Need to find alternative products.
- Is it temporary? Can it be solved by change in availability status or adding simple business rules?
  - Head queries:
    - ~ 20% of head queries with zero results are resolved during 7d rolling window.
  - Long tail queries:
    - ~ 100% of long tail queries are not solved from retrieval perspective.

01

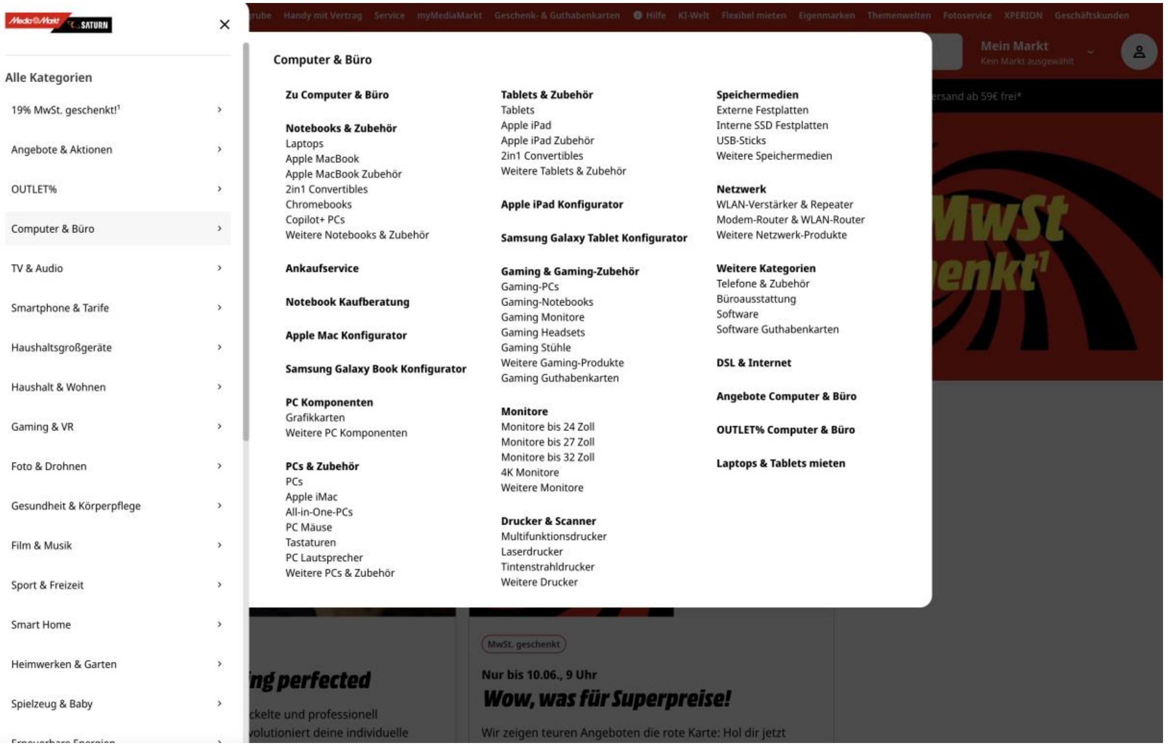
# First iteration

How suitable are public models?

# Offline Evaluation

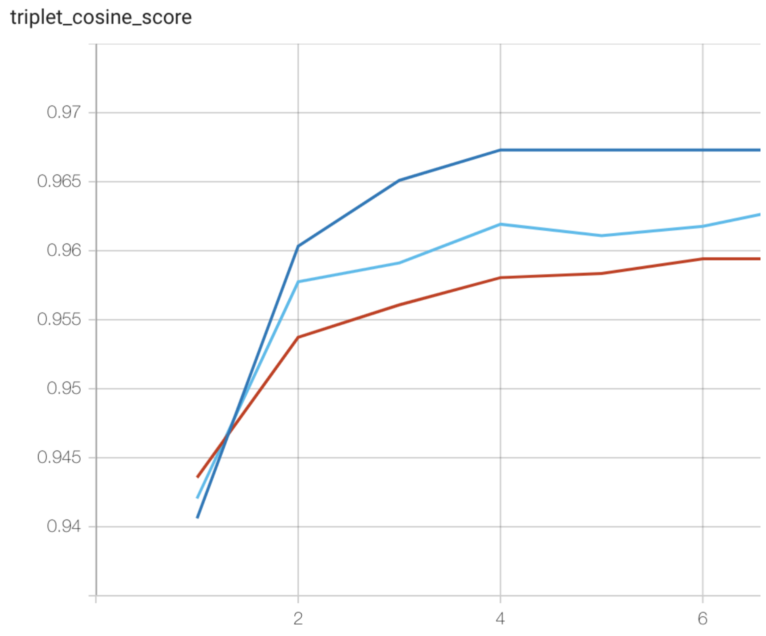
What's a good query-product match?

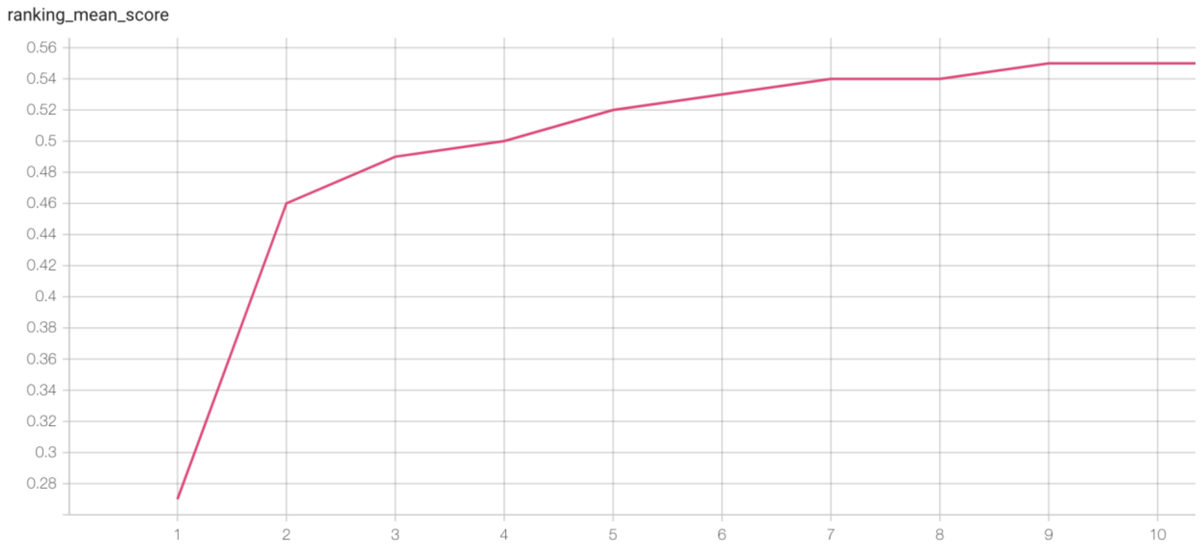
- Text match
  - Is every token equally important?
  - Does it address semantic queries?
  - Eg:
    - Search query: *spülmaschine 45 cm einbau* [dishwasher 45 cm built-in]
    - Product description: *BOMANN GSPE 7415 VI Geschirrspüler (Einbaugerät (Besteckkorb, 45,00 cm breit, 49 dB (A), E)* [BOMANN GSPE 7415 VI dishwasher (built-in appliance (cutlery basket, 45.00 cm wide, 49 dB (A), E)]
- User implicit feedback
  - Only popular products have implicit feedback
    - We might not always have enough products to map
    - What if the suggested product is not popular but still relevant? Does showing smartphone accessories on a smartphone query completely irrelevant?
- Category based result mapping
  - Deduce query category based on query impressions and user implicit feedback
  - Match query category with product's root category
    - Search query: *s23 case*
    - Matched product*: caseonline case22 backcover sony xperia 1 iv blau*



# Fine-tuning public models

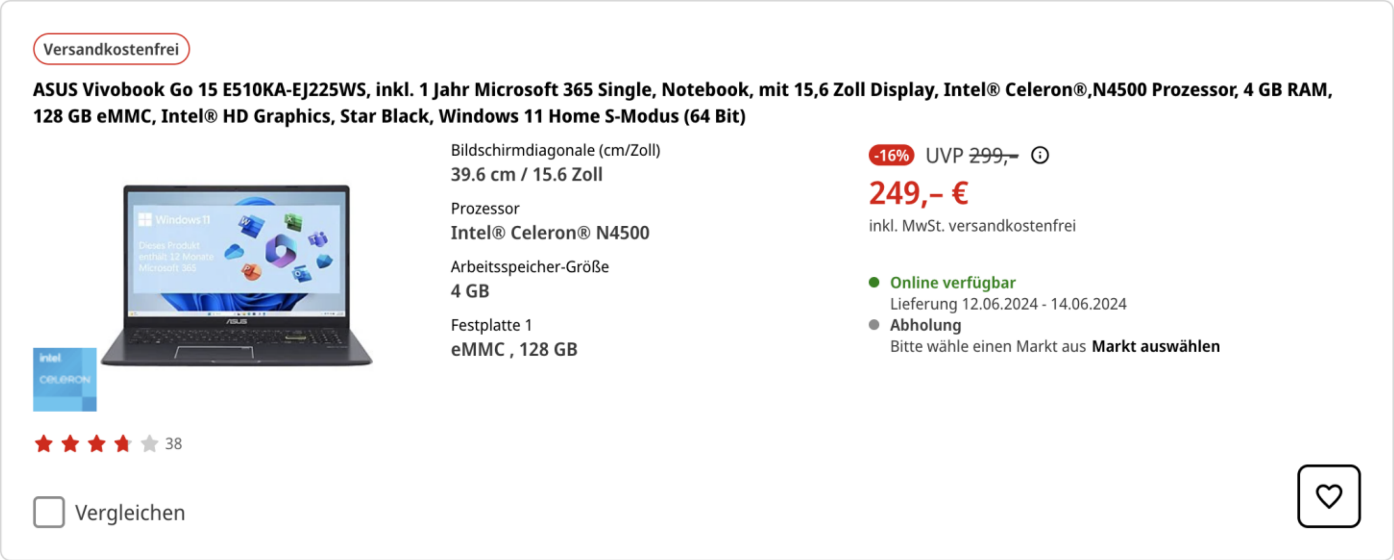
- How to choose a model?
  - Several models for retail in English – not in German
  - Multilingual
  - Size X Pretraining
- Models
  - sentence-transformers/all-MiniLM-L12-v2
  - Sakil/sentence_similarity_semantic_search
  - sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
  - Sahajtomar/German-semantic
  - PM-AI/sts_paraphrase_xlm-roberta-base_de-en
  - …
- Triplet training
  - 70k, 550k, 750k
  - Query – positive – negative
  - Naïve offline negative selection (random, avoiding substrings)



# Limitations

- Models adapt quickly and do not improve anymore
- Unsuitable base vocabulary
- Much noise in the product descriptions
- Too unsophisticated negative selection




```
['as', '##us', 'vivo', '##book', 'go', '15', 'e', '##51', '##0', '##ka', '-', 'e', '##j', '##22', '##5',
'##ws', ',', 'ink', '##l', '.', '1', 'ja', '##hr', 'microsoft', '365', 'single', ',', 'notebook', ',', 'mit',
'15', ',', '6', 'z', '##oll', 'display', ',', 'intel', '##®', 'ce', '##ler', '##on', '##®', ',', 'n', '##45',
'##00', 'pro', '##zes', '##sor', ',', '4', 'gb', 'ram', ',', '128', 'gb', 'em', '##mc', ',', 'intel', '##®',
'hd', 'graphics', ',', 'star', 'black', ',', 'windows', '11', 'home', 's', '-', 'mod', '##us', '(', '64',
'bit', ')']
```

02

# Second iteration

Custom model and product descriptions

# Product descriptions from user behaviour

- User queries are shorter in length
    - ~90% of the queries have up to 4 tokens
    - Avg. 10 tokens per product description
        - redundant information, noise, 100s of structured features per product

SEARCH QUERY: *apple macbook m1*

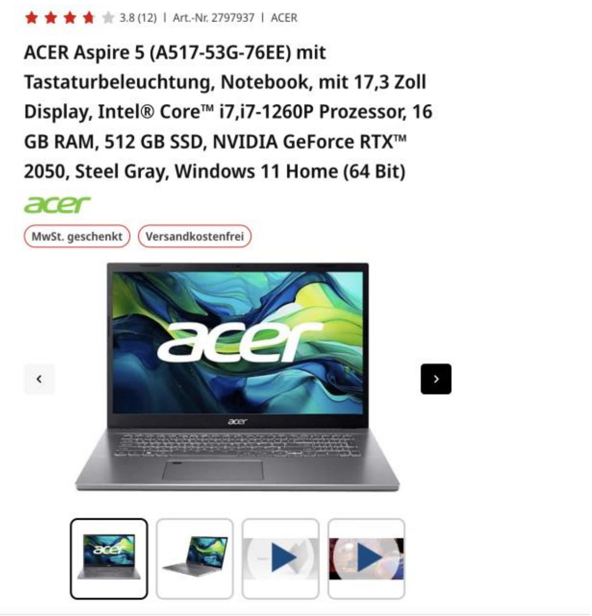

Goal is to find important components of the product description to reduce noise

- Iteration 1:
    - Find common terms between query and products
        - Loss of information
        - Product description might not always map to user queries
- Iteration 2:
    - Entity recognition from user queries to identify important features searched by users.
    - Use these features to build product description and model vocabulary
        - Different patterns per product category
            - Eg: prozessor-modell is important for Notebooks but not for smartphones while color might be important for smartphones but not for notebooks
        - Dynamic number of features per product category
            - Eg: drones can have up to 4 candidate features while notebooks might have up to 7 candidate features
        - Some common features across all categories like brand name, product type
        - Select top 4 features per product category

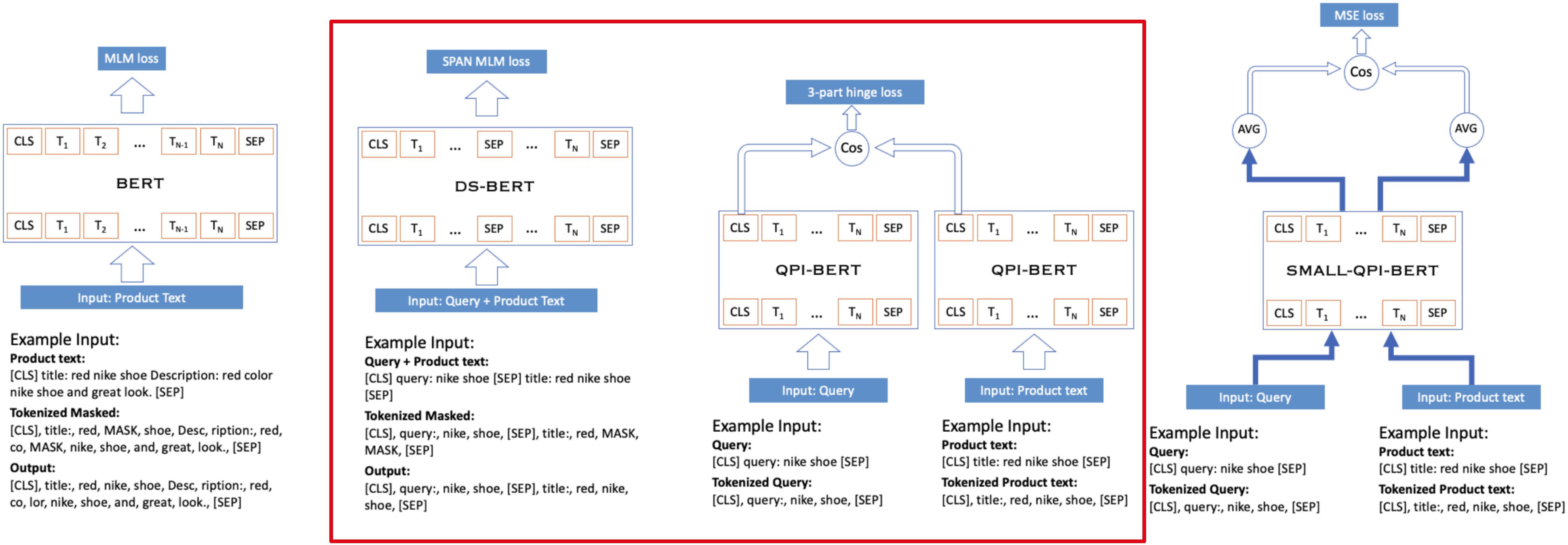
# User behaviour based product descriptions

- Notebooks
  - Candidate features:
    - Processor model
    - Series
    - Brand
    - Product type
    - Front camera resolution
    - Screen size diagonal
    - Total available graphics memory

  - Search query : *laptop 17 zoll windows 11*
  - Product display name: *ACER Aspire 5 (A517-53G-76EE) mit Tastaturbeleuchtung, Notebook, mit 17,3 Zoll Display, Intel® Core™ i7,i7-1260P Prozessor, 16 GB RAM, 512 GB SSD, NVIDIA GeForce RTX™ 2050, Steel Gray, Windows 11 Home (64 Bit)*
  - Generated product description : *acer notebook 17.3 zoll core™ i7*



# Muhamed et al. (2023)

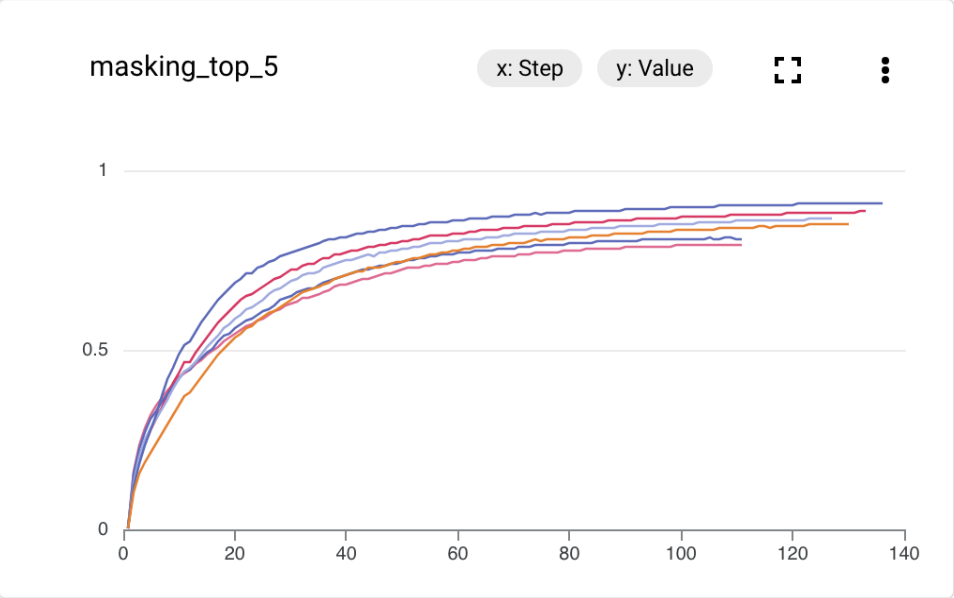



//

# Training from scratch

- Masked Language Modeling
- Dynamic masking (during training)
- Query-product tuples

ipad tablets apple ipad [MASK] 64 gb 10.2 [MASK] space grau

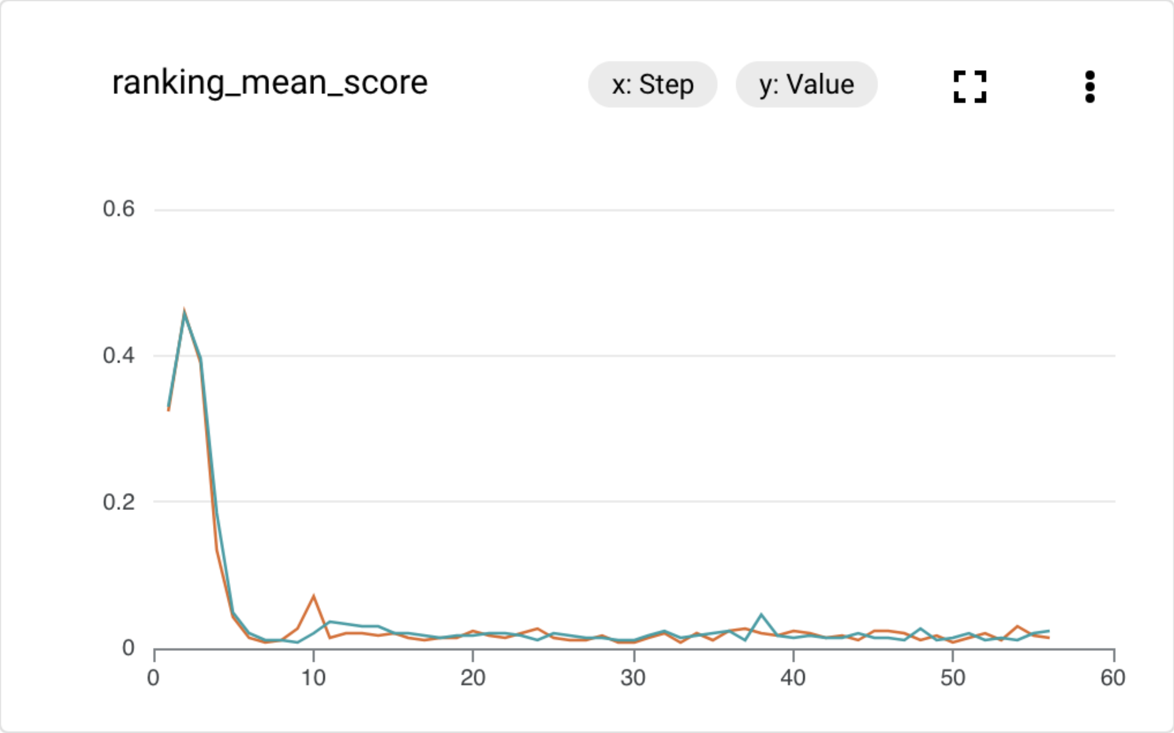



| Row | text |
|---|---|
| 1 | ipad tablets apple ipad wi-fi 9. generation 2021 tablet 64 gb 10.2 zoll space grau |
| 2 | ipad tablets apple ipad wi-fi 10. generation 2022 tablet 64 gb 10.9 zoll silber |
| 3 | mac safe kopfhörer & headsets apple a airpods pro 2nd generation in-ear kopfhörer bluetooth white |
| 4 | 03 wh wasserkocher smeg wasserkocher weiß |
| 5 | 1 tb samsung ssd touch festplatte (extern) samsung ww ssd t7 touch 1tb black 1 tb extern schwarz |
| 6 | 1.5 tb sandisk speicherkarte sandisk ultra plus microsdhc uhs-i adapter micro-sdxc speicherkarte 1.5 tb 160 mb s |
| 7 | 1.5v aa batterie batterien & akkus gp batteries aa mignon batterie alkaline super 1.5v 40 stück |

# How things can go wrong...

- Brutal negatives
  - Create an in-memory index at the beginning of an epoch
  - Take the highest-ranked unexpected products
- Small batch-size

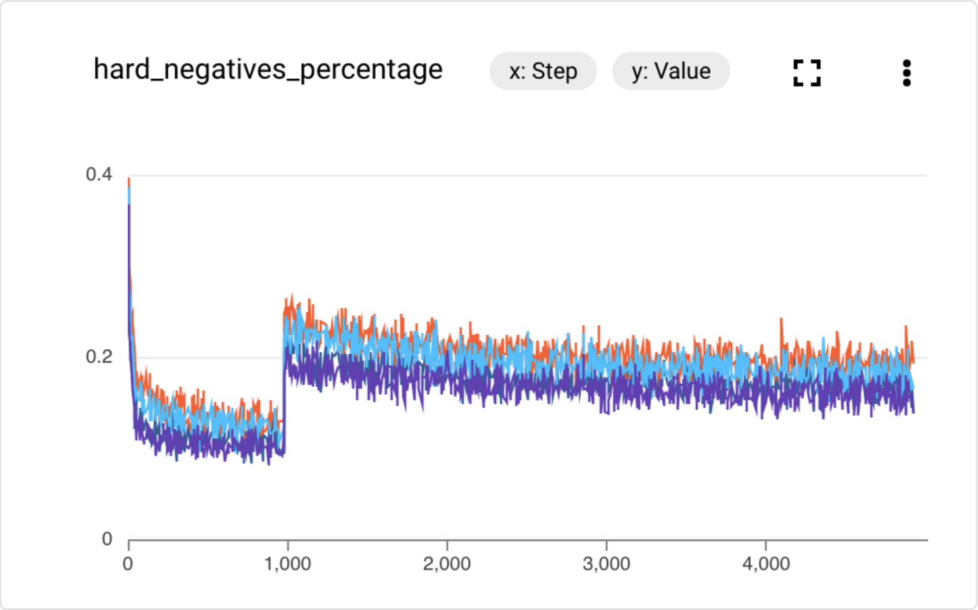
Model over-adapts!

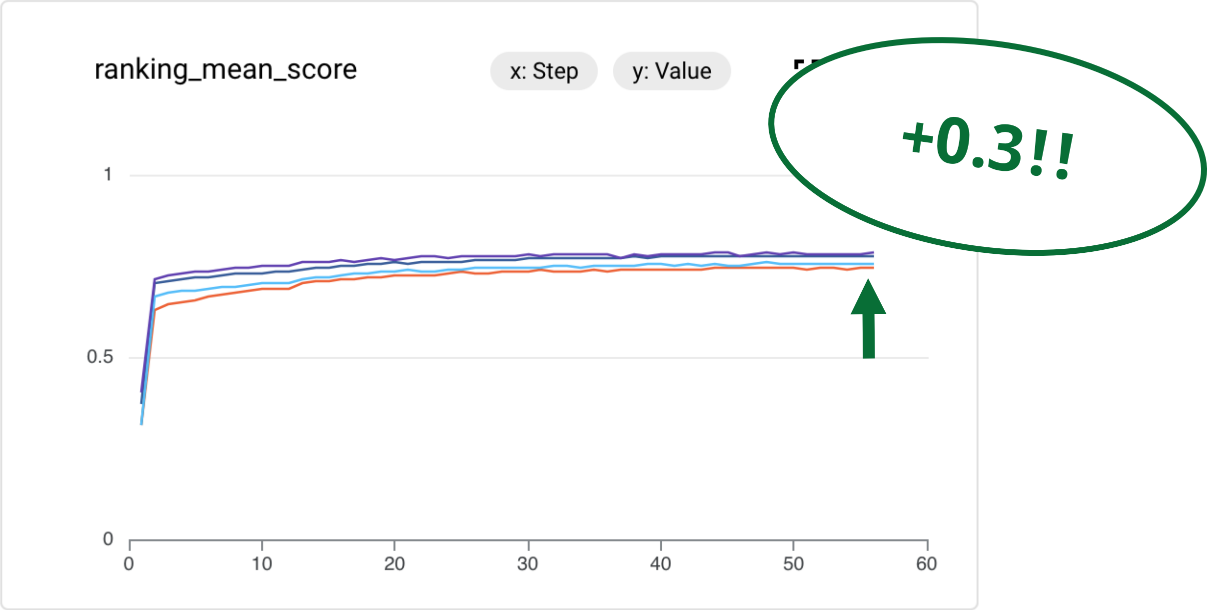


# Fine-tuning with online negatives (Schroff et al., 2015)

- Creation of online negatives
  - Create a batch of anchor-positive tuples (query + relevant product)
  - Create embeddings for all queries and products in the batch
  - A semi-hard negative is farer away from the query than the positive, but still close
  - A hard negative is closer to the query than the positive
- Start soft with semi-hard negatives, continue with hard negatives

- Model variations without significant impact (vocab size, model size)



03

# Third iteration

Striving for the MVP

# Integrating business logic and user behaviour for product descriptions

Shortcomings from user behaviour approach:

- Product catalogue not well maintained  (missing field values/ too long descriptions)
- Feature disambiguation
- Hand crafted rules to extract feature information (processor models: m-series vs M1, smartphone model names, drone model series, etc)
- Redundant features (like maximum storage capacity, delivery information, package information, etc)

Solution: Integrate business knowledge

- Catalogue managers have set of rules to build product descriptions from product features based on business knowledge
- Consider global feature importance on search result page

| Business features | User behaviour-based features | Product description |
|---|---|---|
| - Name<br>- Color<br>- Category<br>- Brand<br>- Storage capacity<br>- Dual sim | - Series<br>- Color<br>- Model<br>- Brand<br>- Storage capacity<br>- Mobile radio standard<br>- Model year | - User behaviour-based product description: ***apple iphone 14 pro 128gb space schwarz***<br><br>- Integrated approach: ***Apple iphone 14 pro 128 gb space schwarz dual sim*** |

# Leanest way of running vector search in production

- Base image
- Code + Dependencies

**Scheduled Job**
- Load model
- Create Embeddings
- Create in-memory index
- Dump index

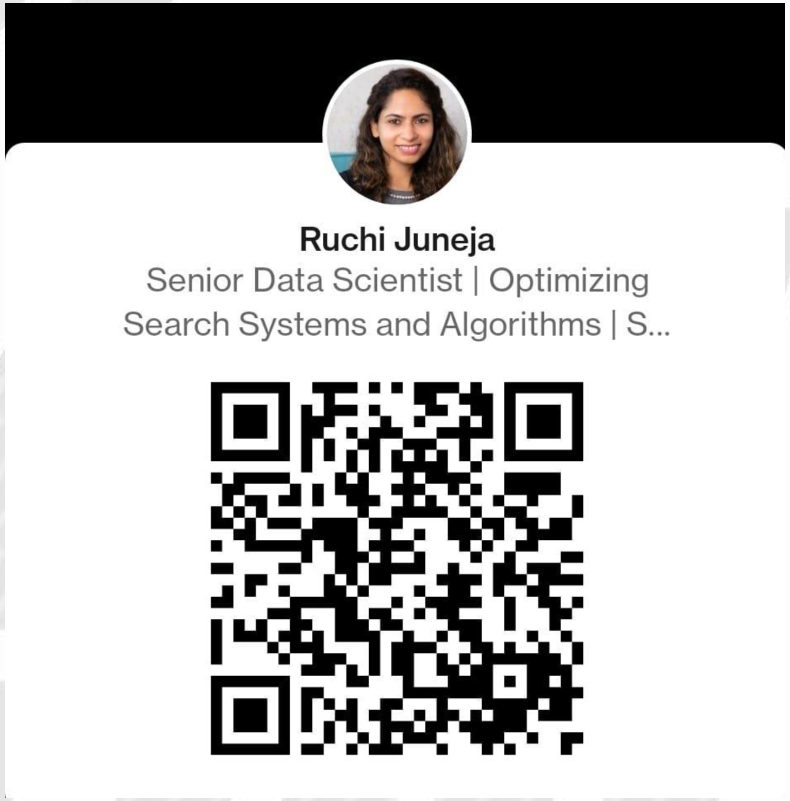
- Code, model, index
- Runs index in-memory

- Run containers in serverless engine
- No database needed
- Low latencies
- Easy scaling
- Indexing = Redeployment

# Future perspectives

- Done
  - ✓ Main implementations
  - ✓ Infrastructure integration
  - ✓ Setup of different training options
  - ✓ CI/CD for models

- Produce data for
  - Spellings / (de)compound
  - Addressing multi-language
  - Synonym relations
- Evaluating zero-results
  - Iteration 1: Manual evaluation followed by AB testing
  - Iteration 2:
    - Use similar queries
    - Ensemble technique using OpenAI


Dr. Johannes Peter
Principal Search Consultant

Ruchi Juneja
Sr. Data Scientist
MediaMarkt-Saturn

# THANK YOU!