allegro

Dense Retrieval for Allegro Search Engine

Aleksandra Chrabrowa, Machine Learning Research, Allegro

06/04/2024 GHOST Day

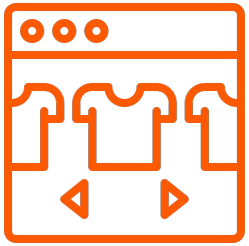
Intro to search at Allegro

Scale and business impact

Over 20m internet users[1] visit the site every month which is almost

70%

of Polish residents aged 15+ and around 75% of all internet users in Poland[1]



Allegro users can select from over **500m** offers/ **140m** products.[2]

Case study: **AI implementation directly translates into GMV[3] growth**

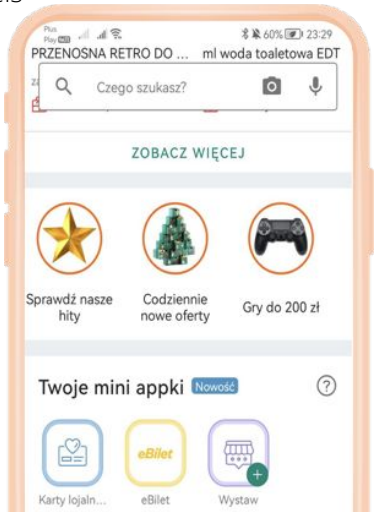
Search engine optimization

Since introduction of AI to Allegro search, we have moved towards more and more advanced intelligent algorithms to **help customers find what they need more easily.**

Current ML models result in over

PLN 1bn

Incremental GMV annually attributable to AI improvements



1. Source: Gemius Mediapanel as of Dec-23
2. Multiple merchant can sell the same products through their offers.
3. GMV = Gross merchandise value is the value of goods sold via e-commerce platforms, calculated prior to the deduction of any fees or expenses.

Dense Retrieval for Allegro Search Engine

Plan of the presentation

Abstract

Modern system designs for e-commerce search are complex, multi-layered systems that must adhere to various online and offline performance requirements. Retrieval is a key component of search engines. In the face of today's dominance by neural networks and large language models (LLMs), it may be surprising to discover that old-school lexical retrieval remains a tough opponent for dense retrieval, which leverages neural networks. During this talk, you will learn what dense retrieval is and delve into the dense retrieval training processes employed at Allegro.

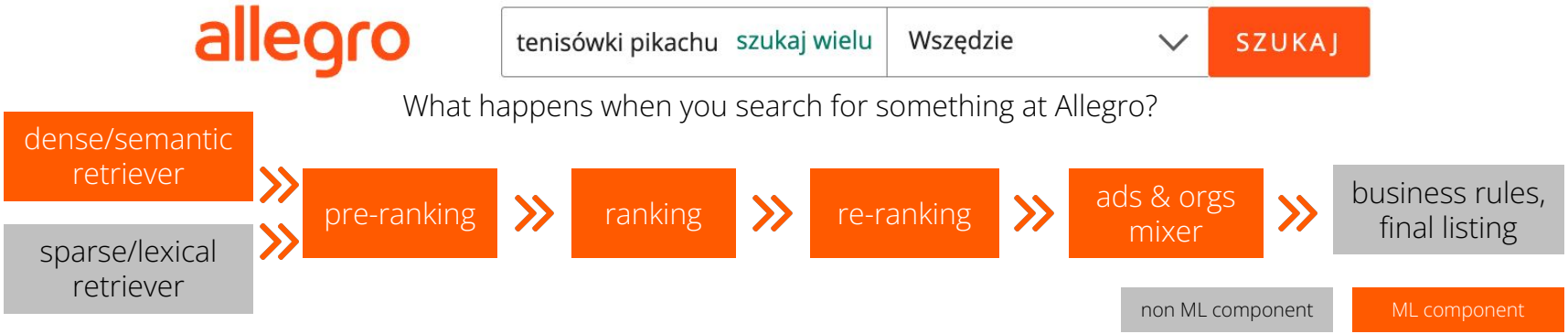
Plan of the presentation

1. ML in e-commerce search pipeline
   a. Overview of the envisioned system
   b. Dense vs. sparse retriever
   c. From ranking to the final listing
2. Dense retriever model
   a. Training data and metrics
   b. Two-tower architecture for representation learning
   c. Contrastive learning paradigm
   d. Experiments with large batch sizes
   e. Experiments with STAR hard negatives
3. Model deployment and serving
   a. FAISS approximate nearest neighbour search
   b. Hybrid with sparse retriever
4. Summary

ML in e-commerce search pipeline

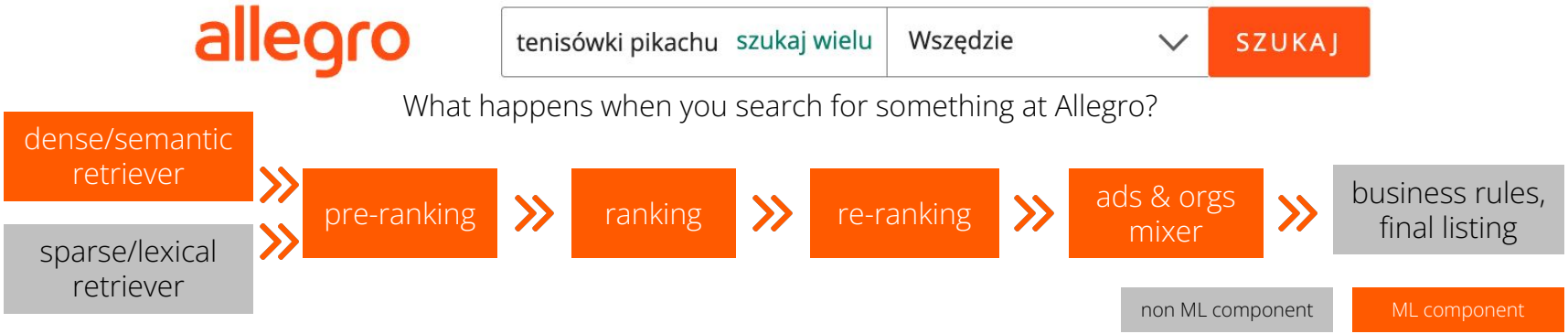
ML in e-commerce search pipeline

Overview of the envisioned system



ML in e-commerce search pipeline

Overview of the envisioned system

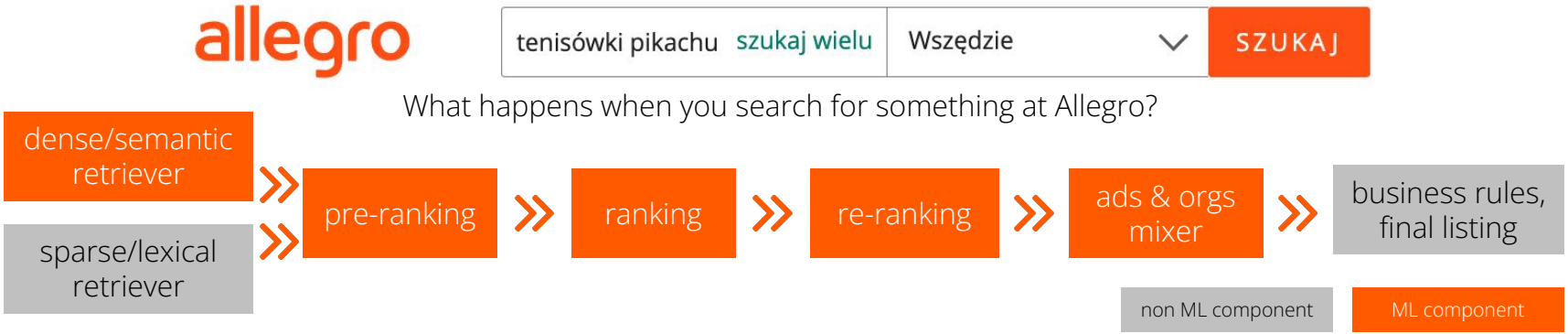


- Envisioned system is work in progress, compatible with state-of-the-art in companies such as Alibaba, Facebook, JD.[1]
- Real system is much more complicated. We focus here only on the ML part of the system.

1.eugeneyan.com/writing/system-design-for-discovery/

ML in e-commerce search pipeline

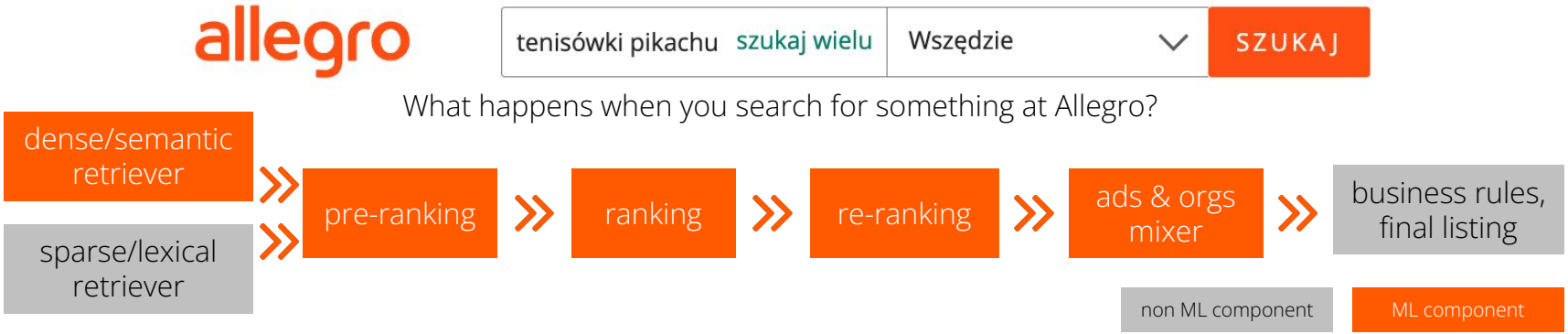
Overview of the envisioned system



each component gradually processes a smaller number of items
each component should work up to tens of milliseconds

ML in e-commerce search pipeline

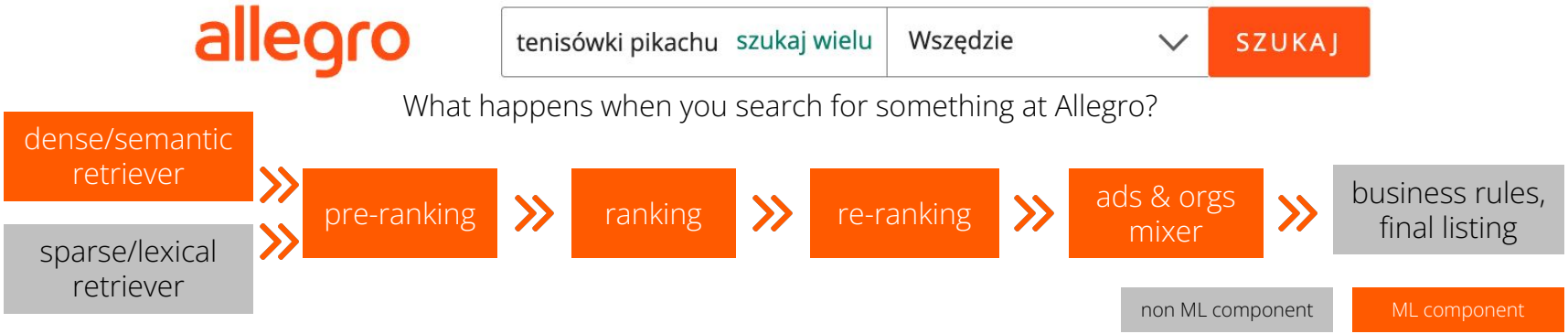
Dense vs. sparse retriever



- Retriever is the first, high-level search among all products/offers with high demands on the response time.
- Older technology: lexical (or sparse) retriever is based on keyword search (exact match).
- Newer technology: semantic (or dense) retriever is based on similarity score between query and product embeddings. Embeddings are computed by a neural network.
- Hybrid approach (combining results from both retrievers) gives best results.

ML in e-commerce search pipeline

From ranking to the final listing

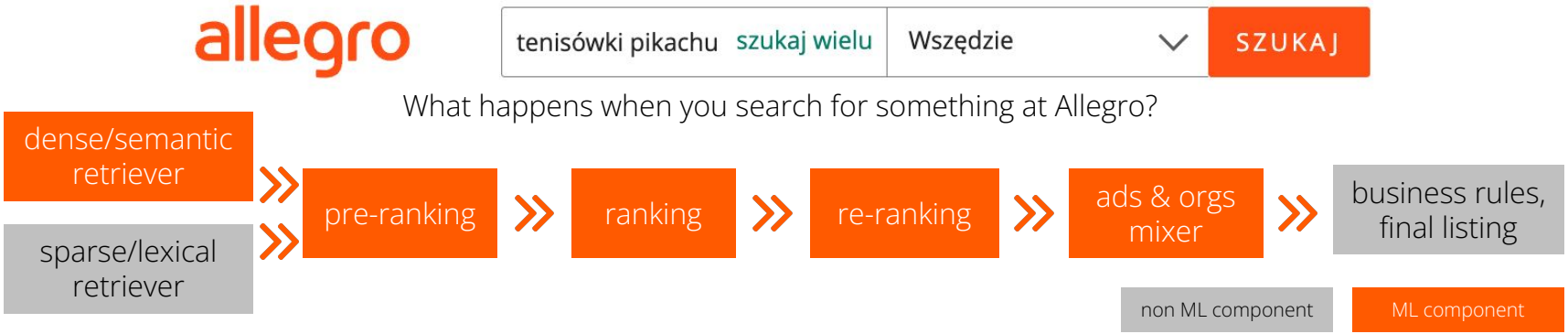


- We need to rank the retrieved offers, so that the most clickable offers are at the top of the list.
- Each ranking (pre-ranking, ranking, re-ranking) step processes a narrower selection of items.
- Both textual (query, offer title) and non-textual (price, popularity, personalization) features are used.
- Award winning Captain - ML Listing model computes the dynamic positions of advertisements.[1]

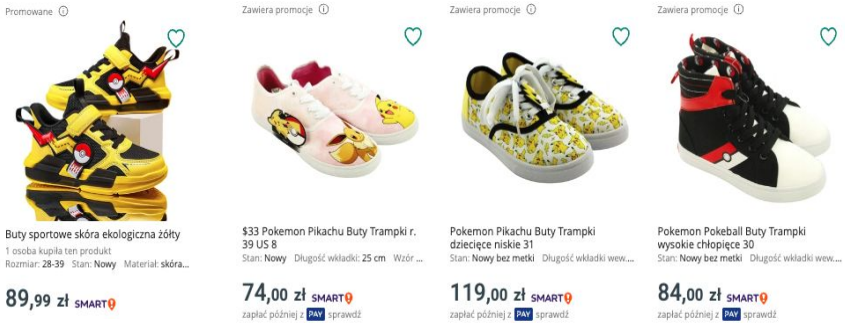
1. media.allegro.pl/297121-allegro-advertising-z-europejska-nagroda-e-commerce

ML in e-commerce search pipeline

From ranking to the final listing



… after all these steps we have the final listing:



Dense retriever model

Dense retriever model

Training data and metrics

- Training data:
  - query + relevant product pairs based on historical clicks in Allegro logs (i.e. user typed a query and clicked a product)
- Metrics:
  - for offline experiments we use mean reciprocal rank (MRR) and recall metrics
  - we calculate Recall@K and MRR@K for top-K products for k=10,60,100...
  - final evaluation via A/B tests

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}$$

Query	Proposed Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
torus	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

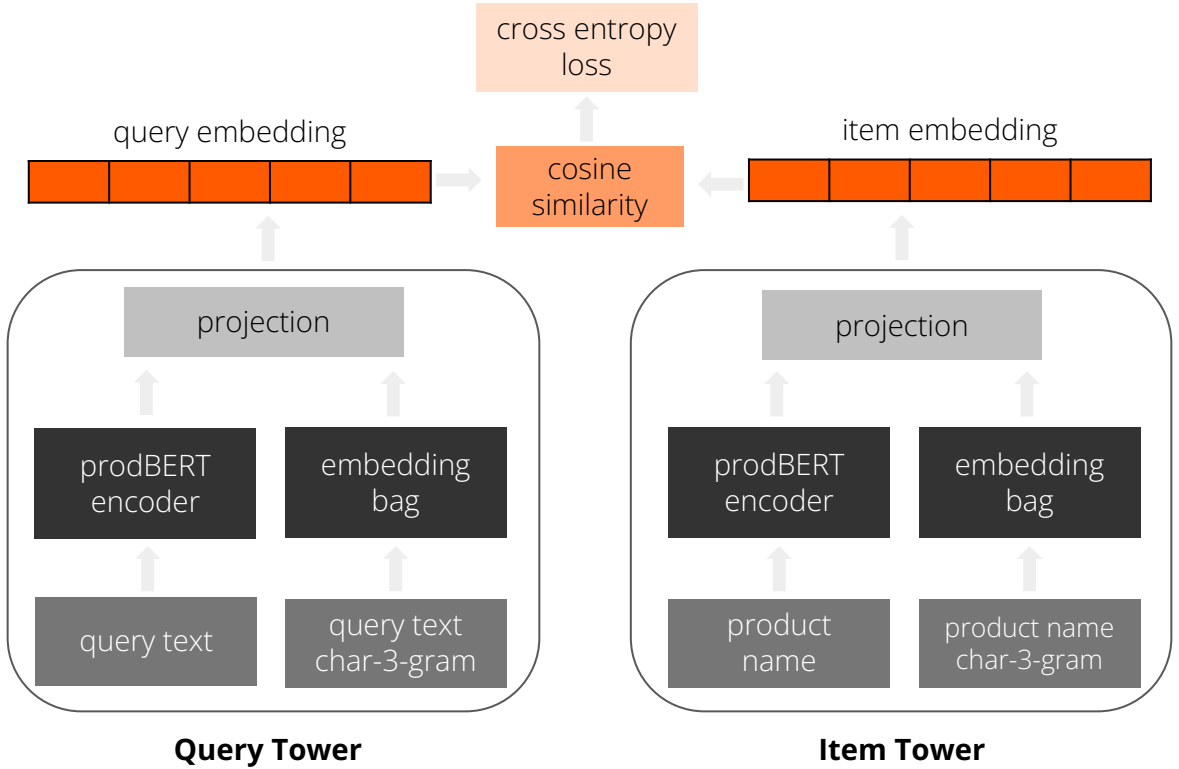
MRR@3=(⅓+½ +1)/3 = 0.61

$$Recall = \frac{1}{|Q|}\sum_{i}^{|Q|}\frac{\#retrieved\ relevant\ items\ for\ q_i}{\#relevant\ items\ for\ q_i}$$

Dense retriever model

Two-tower architecture for embedding

- Two-tower architecture is based on *Que2Search*[1] paper
- Query tower computes query embeddings
- Item tower computes product embeddings
- prodBERT is a BERT model pre-trained on Allegro product data
- char-3-gram is a very shallow model based on character trigrams (*tri, rig, igr, gra, ram, ams*)
- prodBERT is the main boost to the performance but char-3-gram also plays significant role



1. *Que2Search: Fast and Accurate Query and Document Understanding for Search at Facebook* Y. Liu t al. (Meta) 2021

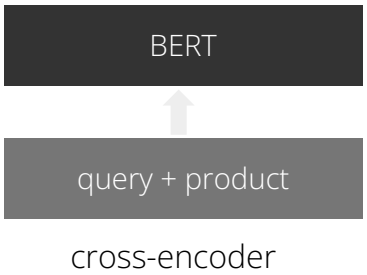
Dense retriever model

Two-tower architecture for embedding

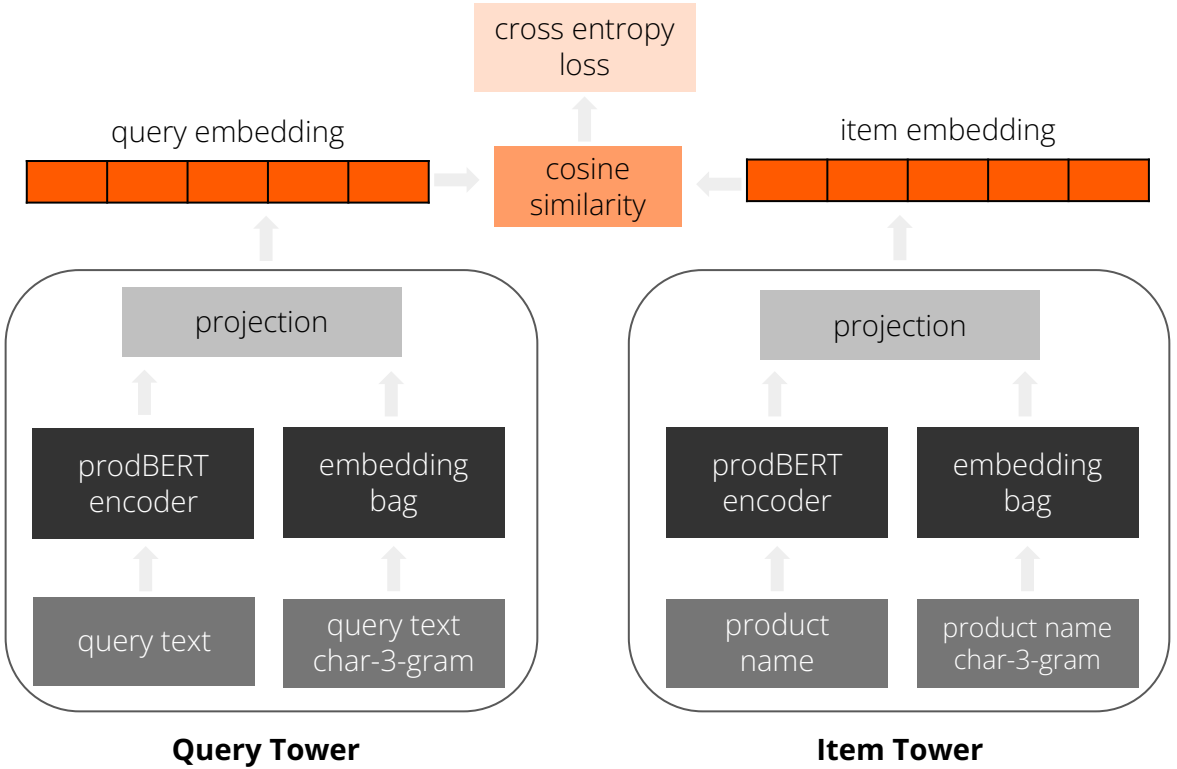
Why two-tower architecture?

- cross-encoder would have the capacity to capture more interactions between query and product
- bi-encoder (two-tower) allows to pre-compute item embeddings offline[1]

Side note: *RocketQA*[2] paper, efficiently combines two approaches



cross-encoder



Query Tower **Item Tower**

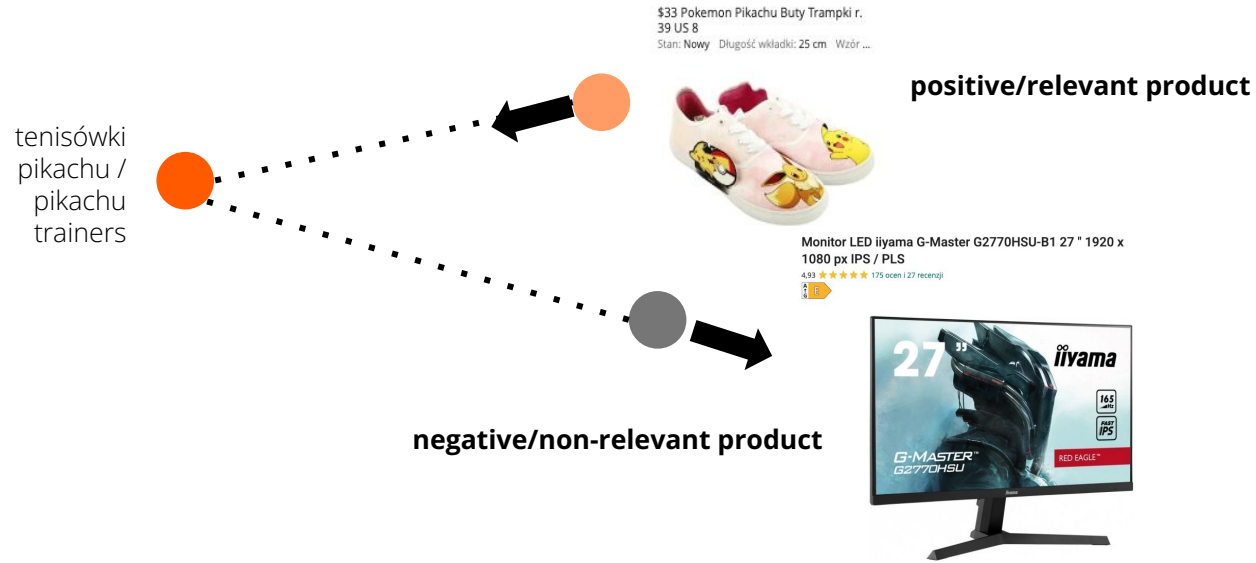
1. More about it in *Model deployment and serving* section
2. *RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking* R. Ren et al. 2021

Dense retriever model

Contrastive learning paradigm[1]

With two-tower architecture we can train the model with contrastive learning paradigm:

- the similarity between query and product embedding is measured with cosine similarity
- query and relevant product (positive pair) are pulled together during training
- query and non-relevant product (negative pair) are pushed away during training

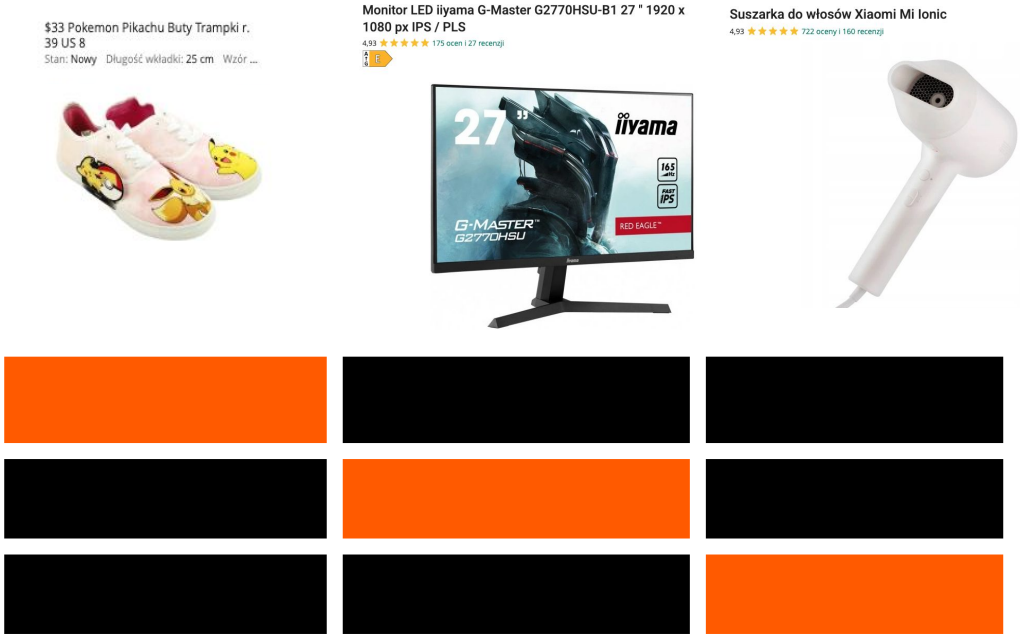


1. paperswithcode.com/task/contrastive-learning 2k+ papers on contrastive learning

Dense retriever model

Contrastive learning paradigm

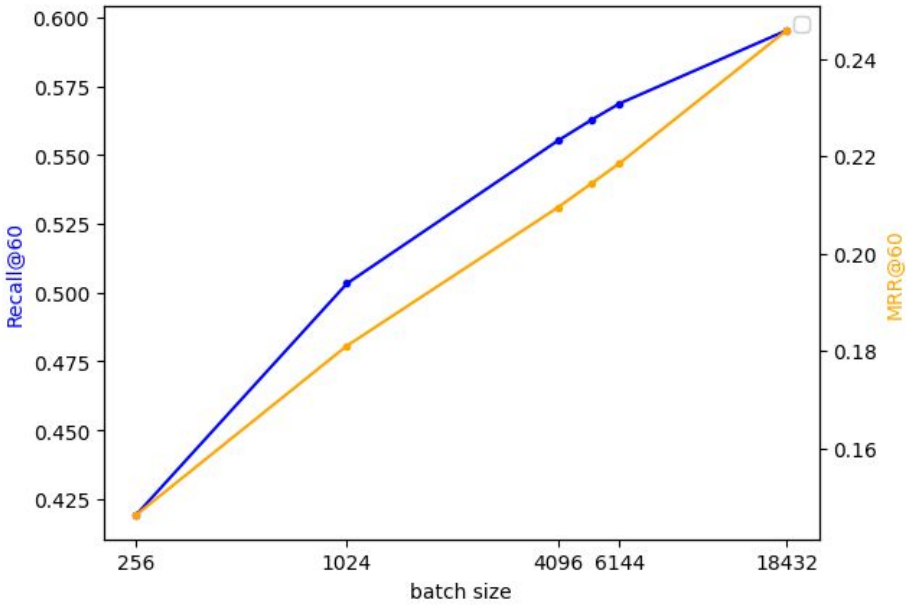
*Quick reminder: in contrastive learning, for batch size=3 we have **3 positive** and **6 negative** in-batch query-product pairs. Increasing the batch size, increases the number of query-product pairs in a batch quadratically.*



Dense retriever model

Experiments with large batch sizes

- If you increase the batch size 2x, Recall@60 will increase by 2.5pp and MRR@60 will increase by 1.5pp.
- The bottleneck for large batch sizes is GPU memory usage during training time.
- Batch size does not affect model deployment and serving
- Retrieval experiments in papers usually have smaller batch sizes
  - our query-products pairs are shorter compared to popular retrieval datasets
  - more powerful compute
  - sometimes the improvement described by a method in a paper holds only for smaller batch sizes



Dense retriever model

Experiments with STAR[1] hard negatives

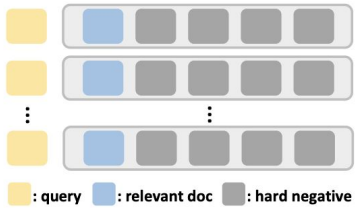
Problem: Are in-batch/random negatives to easy for the model?

Solution: hard negatives

1. Train model with in-batch/random negatives until (almost) convergence.
2. Retrieve STAR hard negatives: top N products most similar for each query with FAISS.[2]
3. Train model from 1. with hard negatives from 2.

Remarks:

- need to filter out highest ranked products in step 2.
- there are a few papers utilizing hard negatives but STAR paper solves some nuances best (training stability).



(a) Input: one relevant doc and multiple static hard negatives are sampled for each query.

(b) Reusing other document embeddings when computing pairwise loss.

Experiment	Recall@60	MRR@60
in-batch negatives (step 1.)	57.6%	22.9%
STAR hard negatives (step 3.)	**65.5%**	**31.1%**

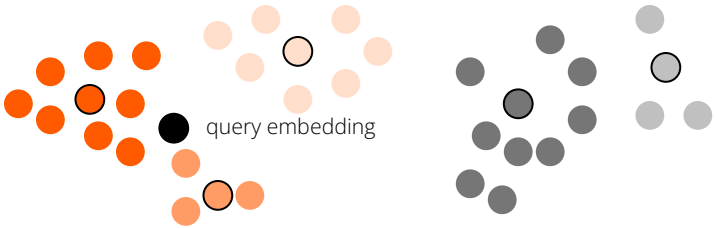
1. *Optimizing Dense Retrieval Model Training with Hard Negatives* J. Zhan et al. 2021
2. Approximate nearest neighbour search - will be covered later.

Model deployment and serving

Model deployment and serving

FAISS[1] approximate nearest neighbour search

- Thanks to two-tower architecture we can split computing embeddings:
  - **Offline**: precompute embeddings offline with **item tower for all Allegro products**
  - **Online**: compute embedding with **query tower per query**
- Task: given a query embedding, find products with most similar embeddings (measured by cosine similarity):
  - Naive solution, exact nearest neighbour search: compute cosine similarity for all products, find most similar products. It would take minutes to compute for every query.
  - Better: **approximate nearest neighbour (ANN) search with FAISS** library: precompute index of all Allegro products offline (hours) to keep the online compute times within tens of milliseconds.
  - Challenge: choosing the right similarity threshold to cut off nearest neighbour candidates -> knee method.
  - Challenge: choosing the optimal FAISS configuration among many available options.



General idea of ANN algorithms:

1. Compute index: cluster item embeddings with KMeans
2. Exact search among cluster centroids
3. Exact search among members of top clusters

1. ai.meta.com/tools/faiss/ and *The Faiss library* M. Douze et al. (Meta) 2024

Model deployment and serving

Hybrid with sparse retriever

Why do we use **hybrid approach** (combine results from dense and sparse retriever)? Despite better performance of dense retrievers, sparse retrievers are:

1. good for exact match queries
2. more explainable and easier to enforce specific queries to work
3. last but not least: fine-tuned with a lot of domain knowledge (because they have been running in production for some time)

Personal opinion: in scientific literature many novel state-of-the-art dense retrieval methods are typically compared again BM25,[1] which is also exact term matching system. However, due to 3. in real e-commerce system, the gap may be smaller.

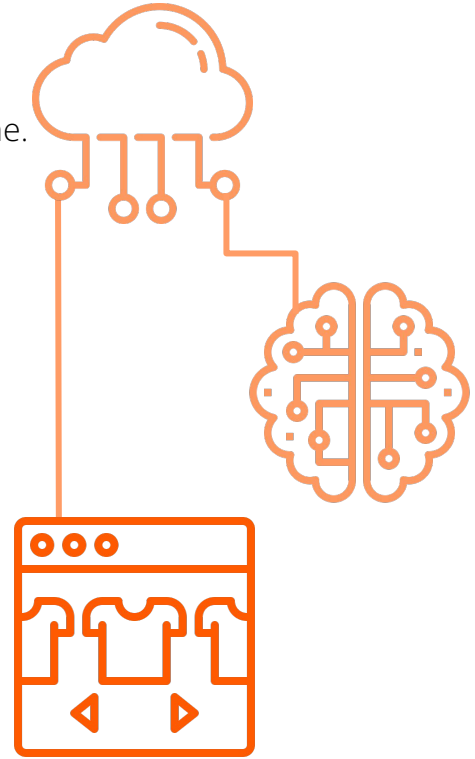
1. *Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval* S. Robertson and S. Walker, 1994

Summary

Summary

Dense Retrieval for Allegro Search Engine

- 20m monthly Allegro users can select from 140m products/500m offers.
- Retriever is the first, high-level search from all products/offers in ML search pipeline.
- There are two types of retrievers: dense/semantic and sparse/lexical.
- Sparse/lexical retriever is based on keyword search (exact match).
- Dense retriever model:
  - is a neural network with two-tower architecture.
  - uses BERT trained on Allegro product data
  - is trained with contrastive learning paradigm.
  - benefits from large batch sizes and hard negatives.
- Model in production:
  - FAISS/approximate nearest neighbour search is used.
  - hybrid solution with sparse/lexical retriever is used.



Thank you

Any questions?