Building a Semantic Search Engine

SCAN ME

Outline

---

1. Sparse Embedding
2. Dense Embedding
3. Approximate Nearest Neighbor (ANN)
5. Production Considerations

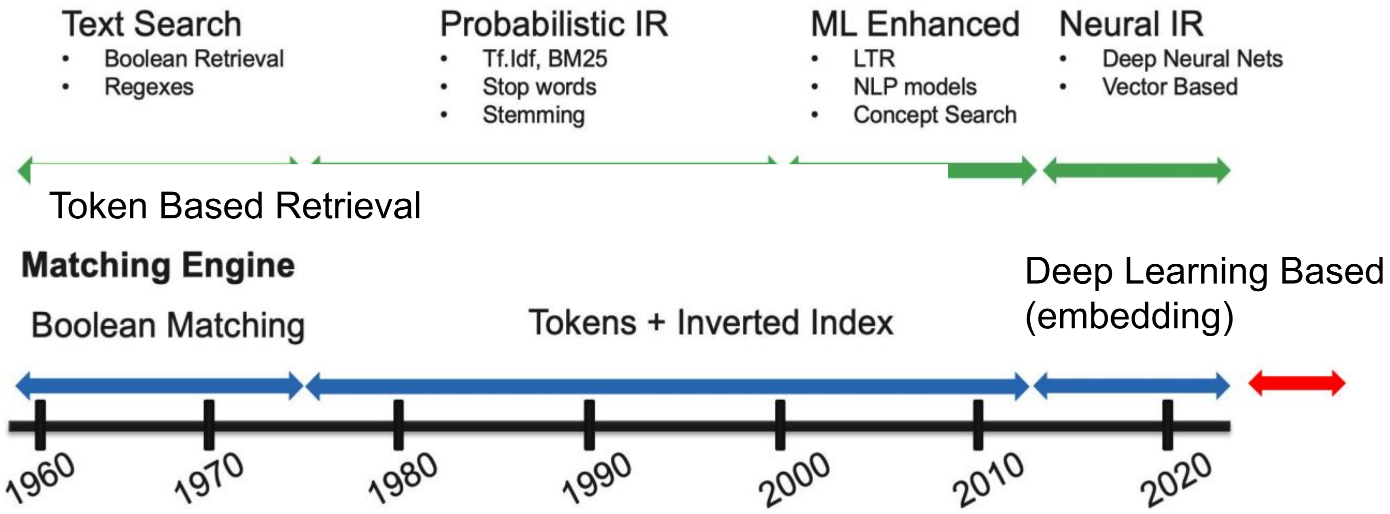
About Us

Ravi Yadav
[Linkedin]

Nidhin Pattaniyil
[Linkedin]

Walmart: ML Engineers on the Search team

History of Information Retrieval

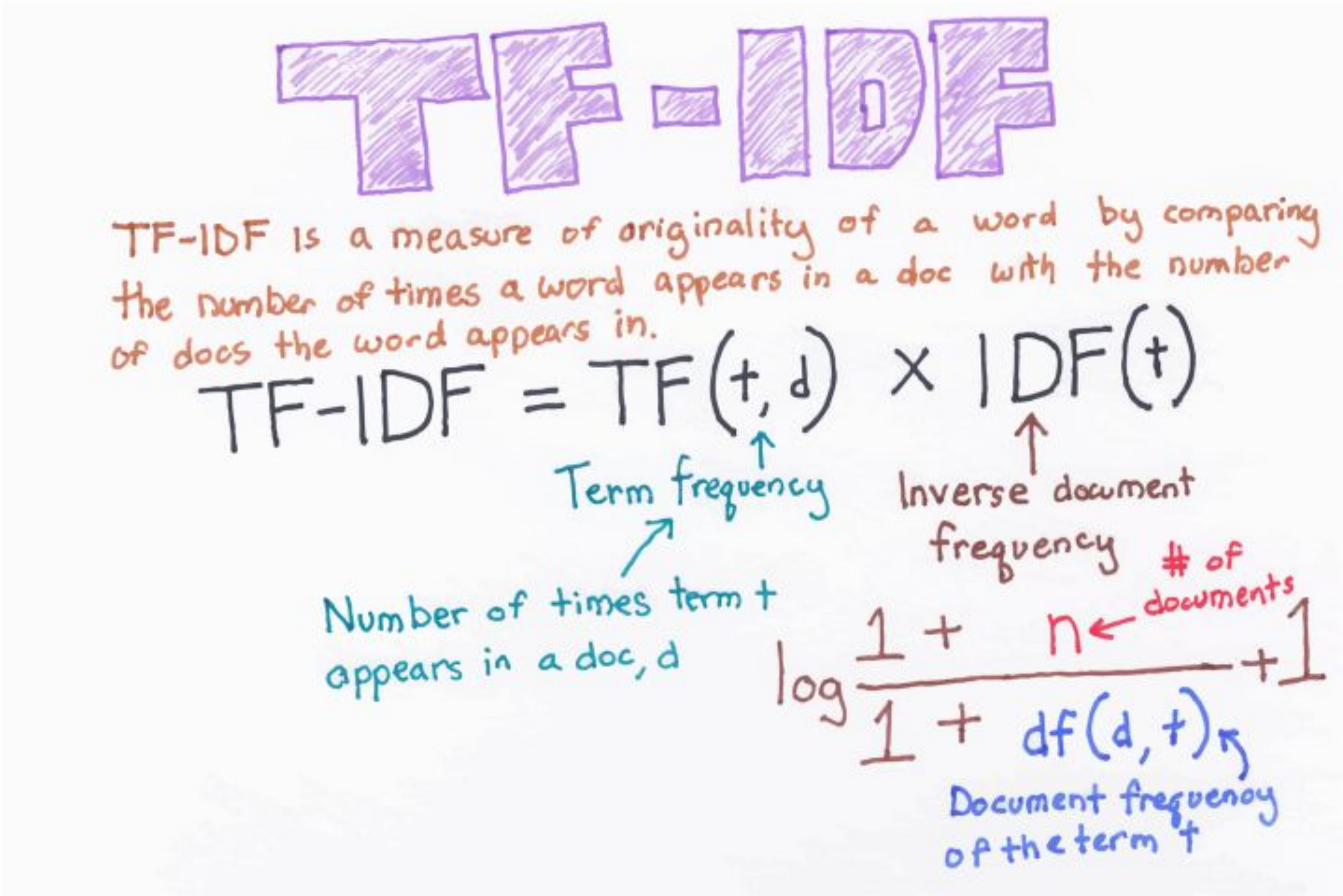


Hughes, Simon. "Semantic Product Search – Vector Search for E-Commerce." Conference Presentation at Haystack 2021, https://haystackconf.com/files/slides/haystack2021/Hughes-Haystack_2021_Semantic_Product_Search.pdf, September 29, 2021.

Token based retrieval

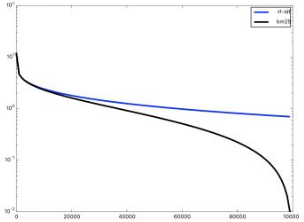
Boolean Retrieval

- Queries and documents are represented as bag of words
- Query terms are connected with boolean operators
- Scans through inverted index and retrieves documents that satisfies the condition
- Query: (beef **OR** chicken) **AND** stroganoff **AND** recipe
- Disadvantages:
    - Filtering more than retrieval
    - Terms have same weights
    - No ranking



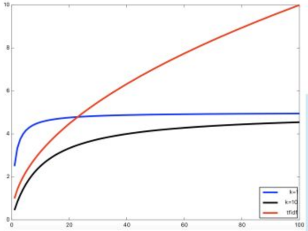
Credit : Chris Albon @ https://machinelearningflashcards.com/

BM25



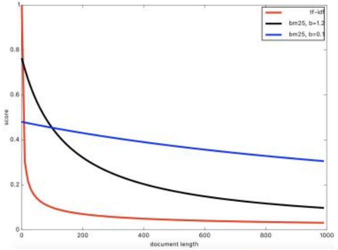
idf - how popular is the term in the corpus?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$



saturation curve - limit influence of tf on the score

length weighing - tweak influence of document length



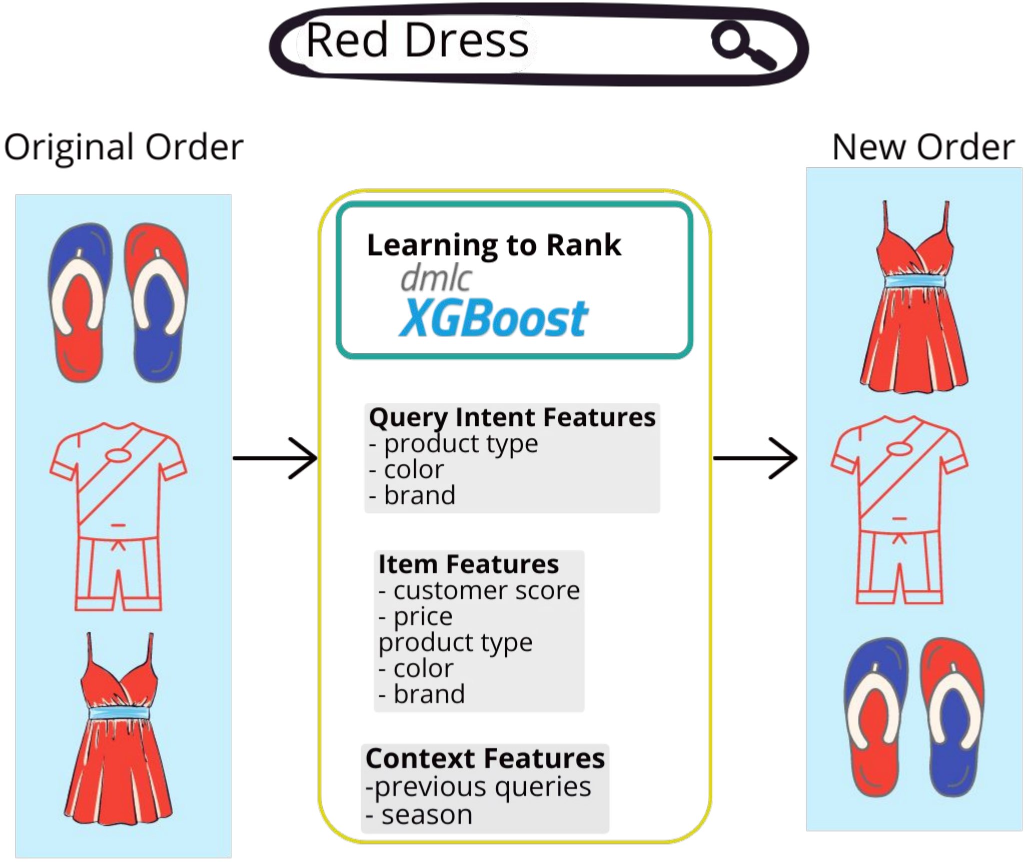
elastic{on}

73

elastic

Weber, Britta. "Improved Text Scoring with BM25." Conference Presentation at ElasticCon 2016, https://speakerdeck.com/elastic/improved-text-scoring-with-bm25, February 11, 2016.

Learning to Rank



Learning to Rank

- Documents and queries are represented as feature vectors (bag of words)
- Query, document and query-document features
  - Query Features: Number of words in query
  - Document Features: **Pagerank**, document length, inlink/outlink count…
  - Query-Document Features: BM25 score, TF*IDF score…
- RankNet, SVM-Rank, RankBoost, MART

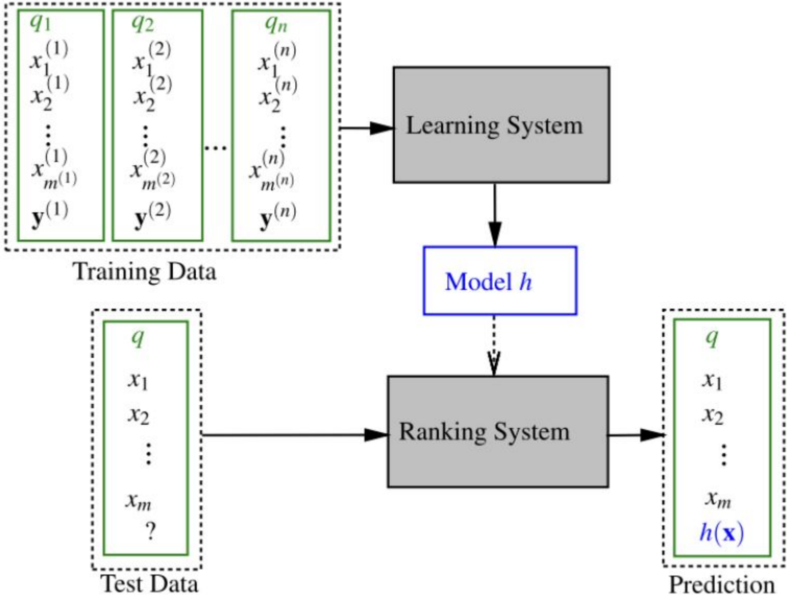


Image taken from https://www.cs.purdue.edu/homes/liu1740/report.pdf

Issues with Sparse Representation

- Lexical GAP: Covid vs Coronavirus vs Omicron variant
- Ambiguity: bank (institution) vs bank (geography)
- Position matters: “river bank” vs “bank river”
- Lack of Contextualized embeddings

  She will ***park*** the car so we can walk in the ***park***.

SCAN ME

Lab

Jupyter Hub: https://hub.np.training

Repo:https://bit.ly/search-workshop-2022

Lab 1 Goals

- Explore tokenization and some preprocessing
- Building a simple in-memory retrieval system using BM-25

Embedding Based Retrieval

Dense Embeddings

<u>Word Representation</u>

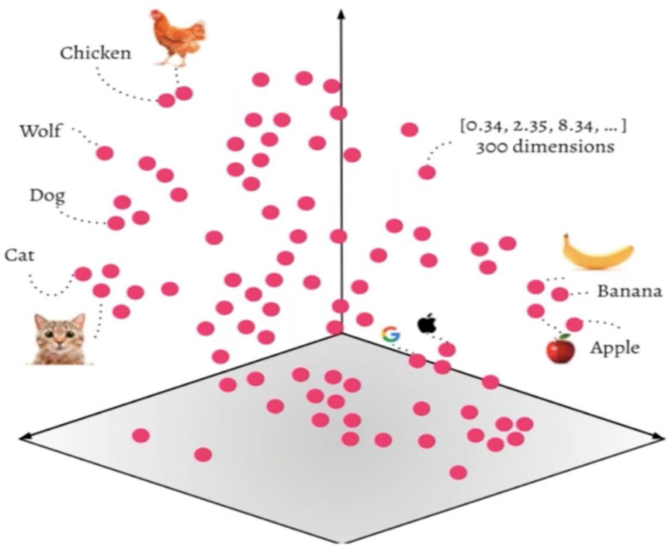
car: [0.2 , 0.3, 0.7]

automobile: [0.2 , 0.3, 0.7]

Similar concepts have similar embeddings

Regardless of content length, similar items should have similar embeddings

Size of embedding is independent of #tokens

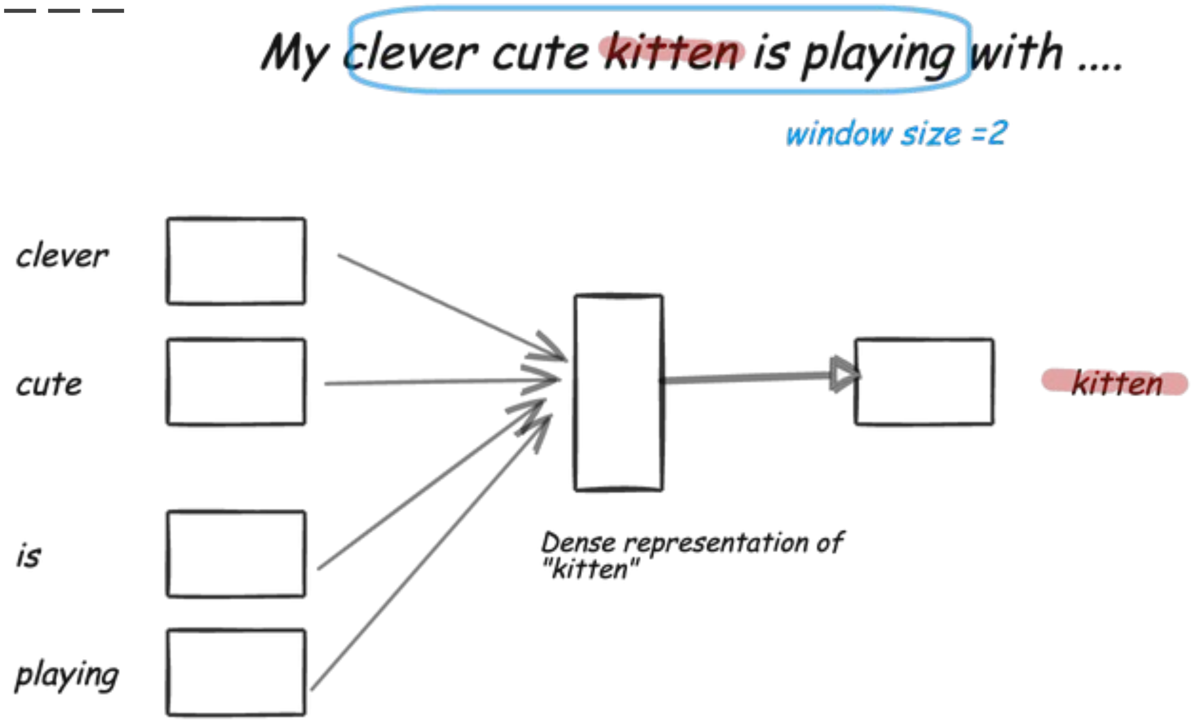


<u>Review Representation</u>

Review 1:  was great [0.5 , 0.1, … , 0.6]
Review 2: Chocolate ice cream was the best .. [0.5 , 0.1, … , 0.5]

Image from https://odsc.com/blog/getting-started-with-vector-based-search.

Word2Vec

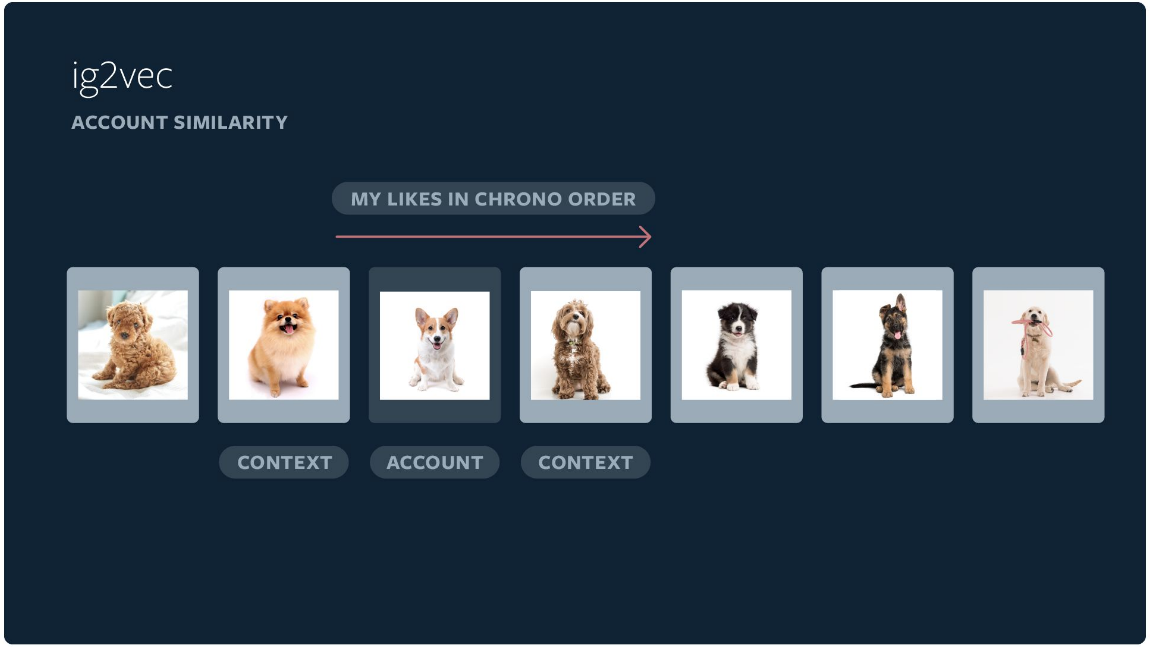


W2Vec Continuous Bag of Words

- Published in 2013
- Represent each word as a dense vector

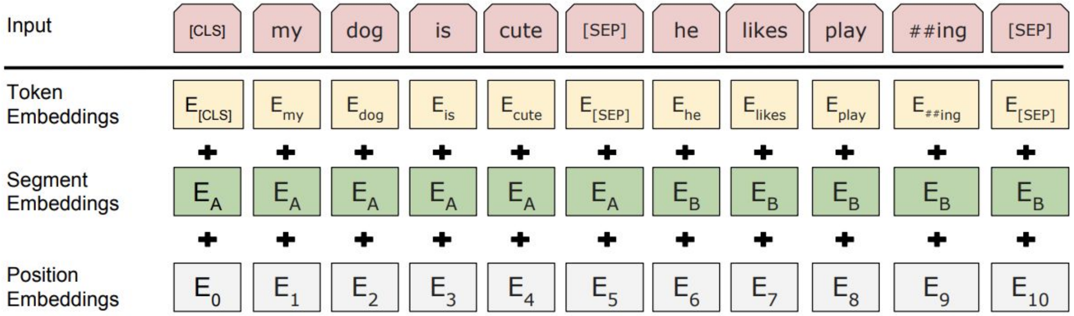
- Uses a neural network model to capture linguistic concept of words

Instagram: ig2Vec



Medvedev, Ivan, Haotian Wu, and Taylor Gordon. “Powered by AI: Instagram’s Explore Recommender System.” Powered by AI: Instagram’s Explore recommender system, November 2019. https://ai.facebook.com/blog/powered-by-ai-instagrams-explore-recommender-system/

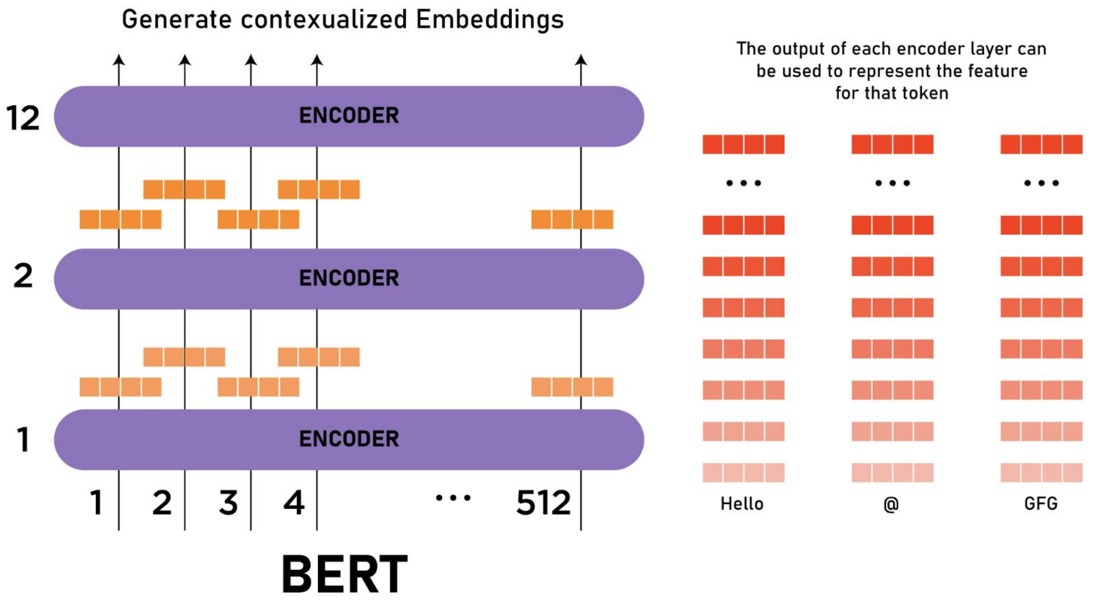
BERT



Input to the model, contains token , segment and position embedding

Image from BERT paper https://arxiv.org/pdf/1810.04805.pdf

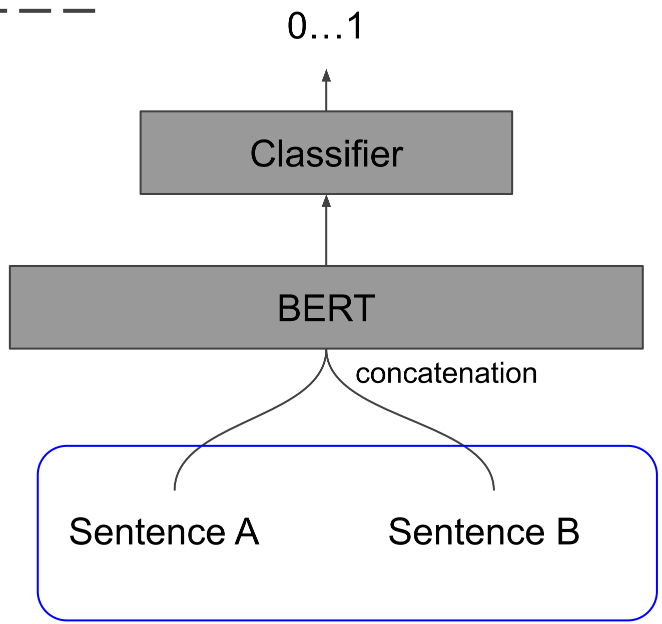
BERT



- BERT is composed of multiple encoders
- There is embedding for each token after each encoder
- Sentence embedding can be created from pooling
- Pooled embeddings are not best for similarity search

Image from https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/

Sentence Pair Encoder - Cross-encoder



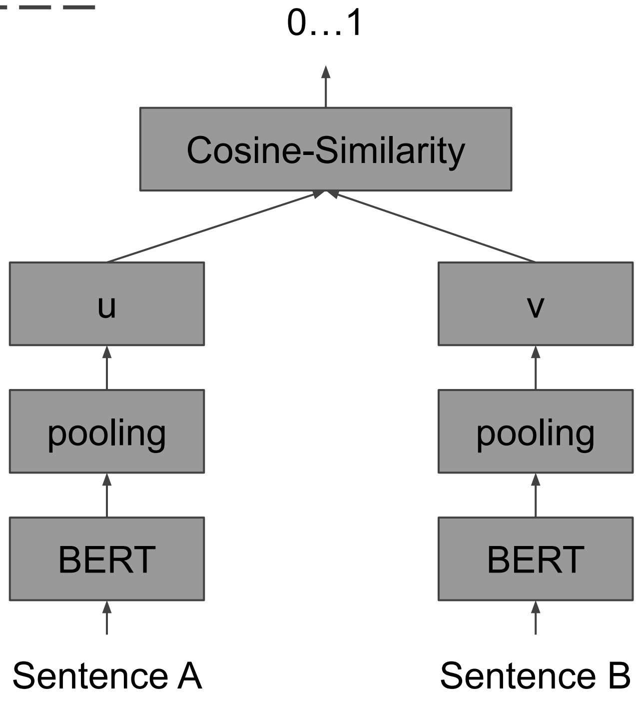
Pros

- Accurate classification

Cons

- Need new encoding for each pair
- Computationally inefficient for information retrieval.

Sentence Pair Encoder - Bi-Encoder



Pros

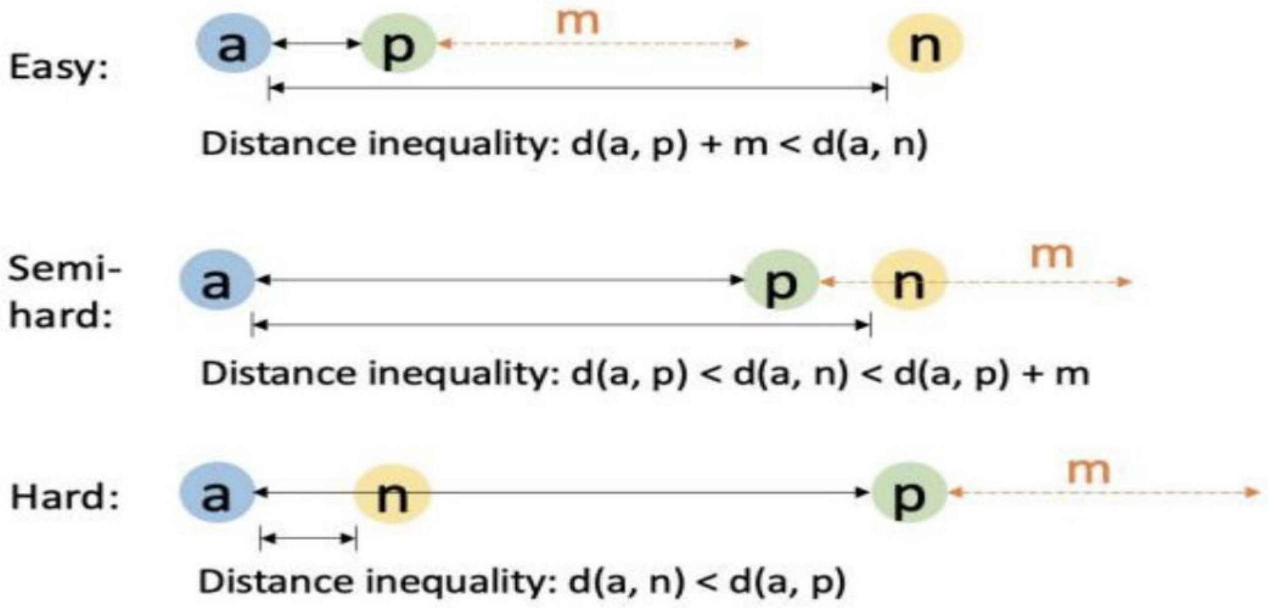
- Each sentence is encoded separately
- Distance between two sentence embeddings can be measured
- Faster and more computationally efficient

Cons

- Bi-encoders performance is poor compared to cross-encoders for supervised learning problems

Different types of negatives

-



Wang, Zeyu, Xiaoyu Liang, και Cheng Wang. ‘Controllable Text-to-Image Generation with Enhanced Text Encoder and Edge-Preserving Embedding’. Journal of Physics: Conference Series 1856 (04 2021): 012003. https://doi.org/10.1088/1742-6596/1856/1/012003.

Multiple Modality: Vision Transformers (ViT)

- Vision Transformer models like CLIP model uses two encoders (text and image).
- These two models are trained in parallel and optimized via a contrastive loss function
- End result is where you can search by text or image

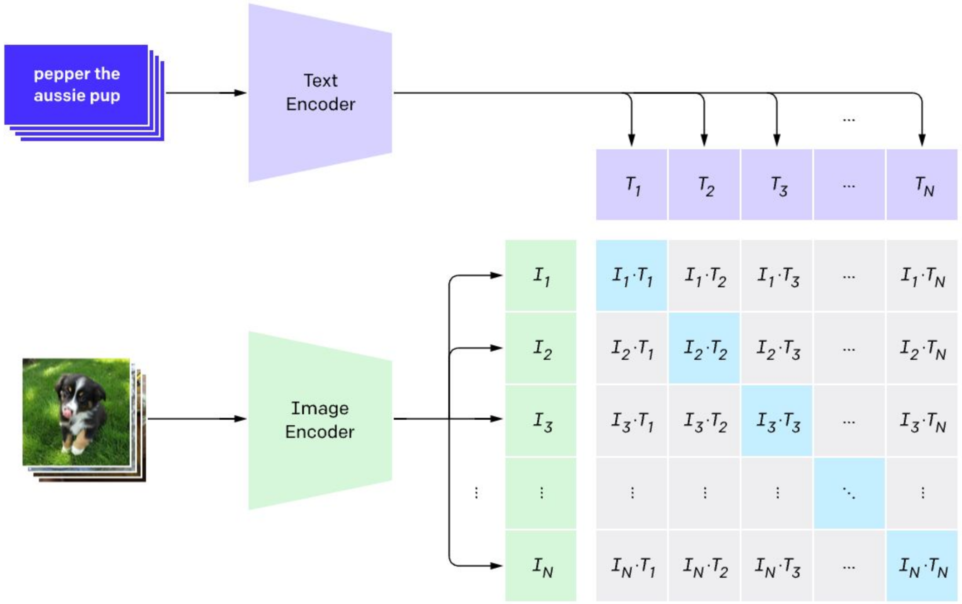


Image from OpenAi CLip blog
https://openai.com/blog/clip/

Encoder can include metadata from multiple fields

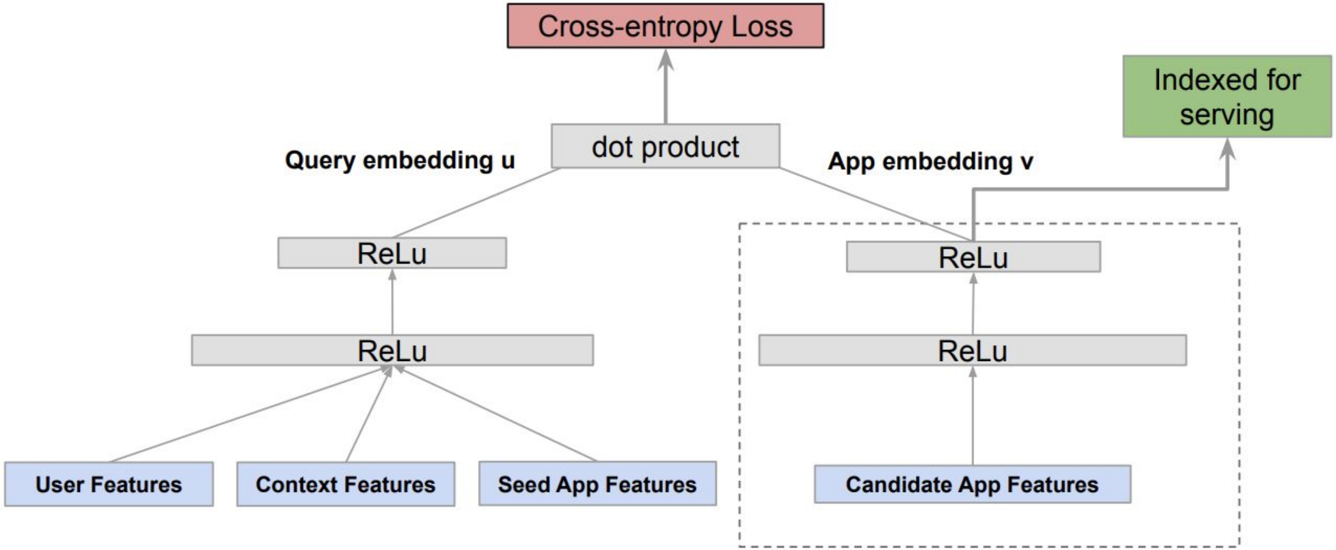


Figure 5: Two-tower model architecture for Google Play app recommendation.

Yang, Ji, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, και Ed H. Chi. 'Mixed Negative Sampling for Learning Two-Tower Neural Networks in Recommendations'. Στο Companion Proceedings of the Web Conference 2020, 441–47. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020. https://doi.org/10.1145/3366424.3386195.

SCAN ME

Lab

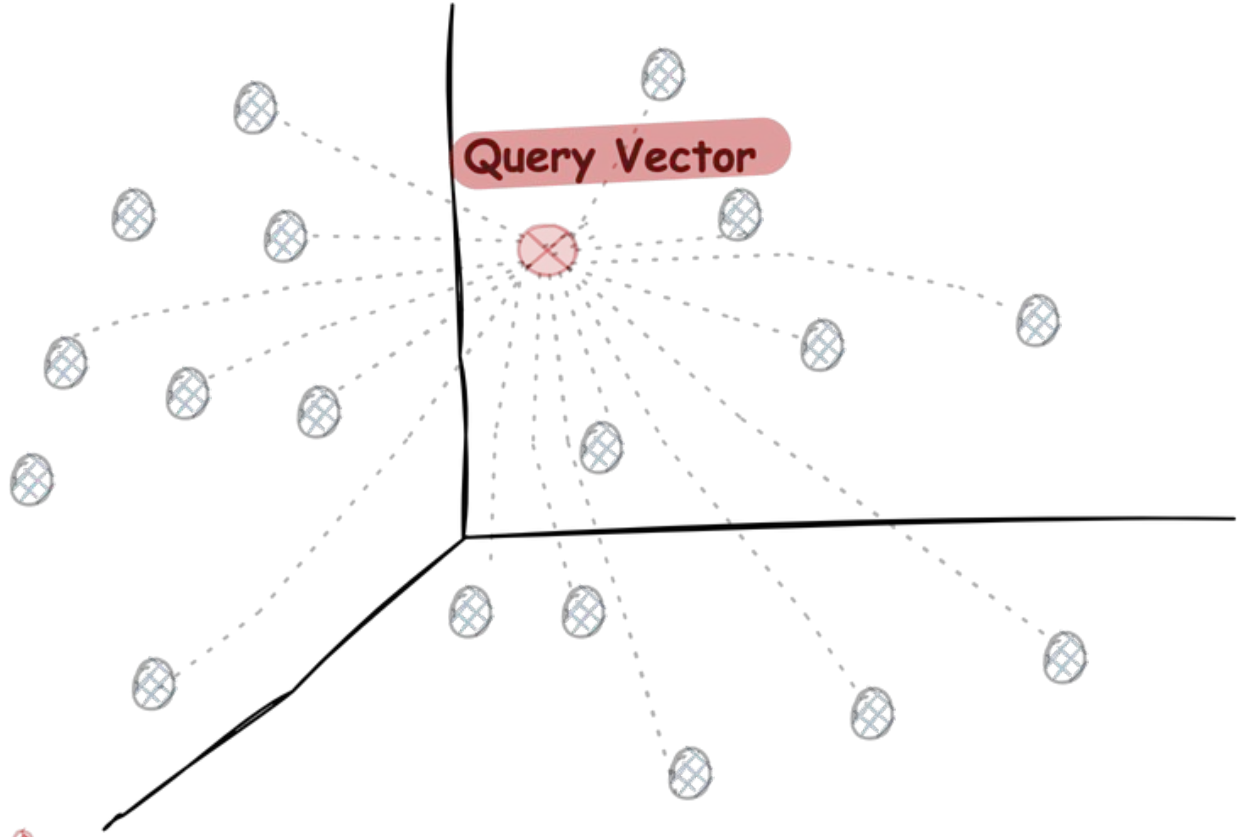
Jupyter Hub: https://hub.np.training

Repo:https://bit.ly/search-workshop-2022

Lab 2 Goals

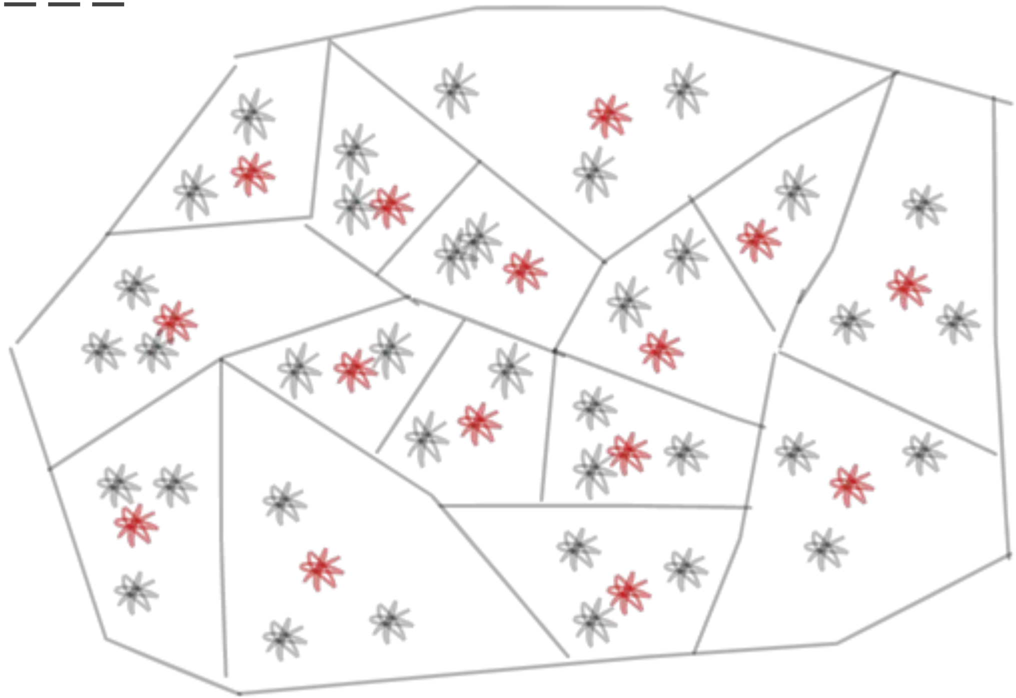
- Explore sub word tokenization
- Building a simple in-memory retrieval system using a language model
- Building a simple in-memory retrieval system using a multi-modal model like CLIP

Approximate Nearest Neighbors



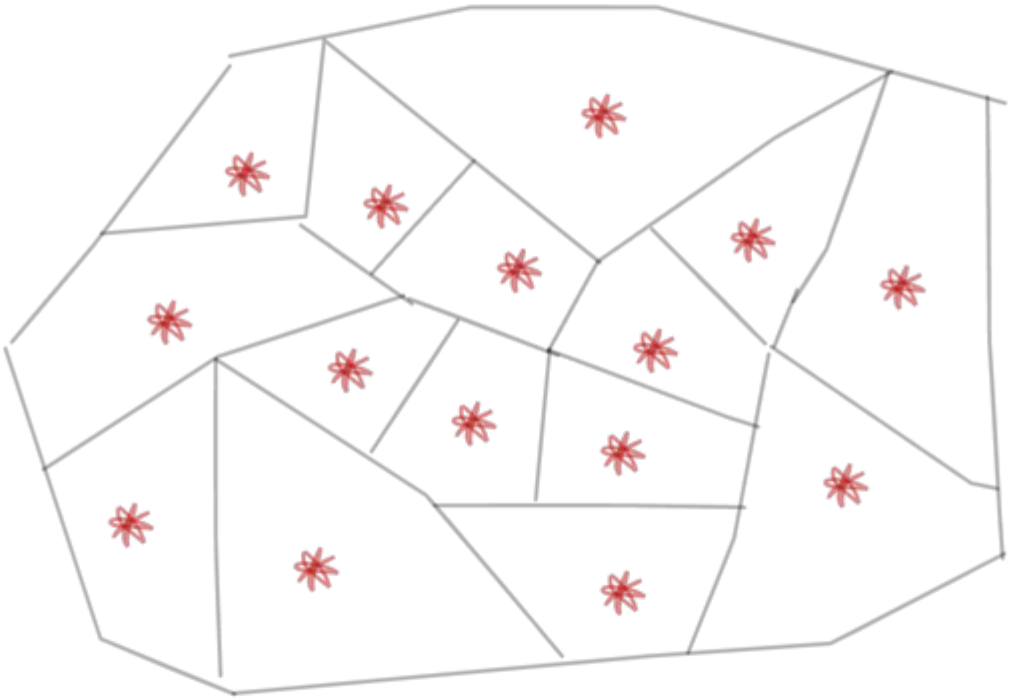
In a flat index / no index, our query vector is compared against all the vectors in the database.

Inverted File Index: Building



- Find centroid and create Voronoi Cells
- Number of centroids is determined by **nlist** parameter

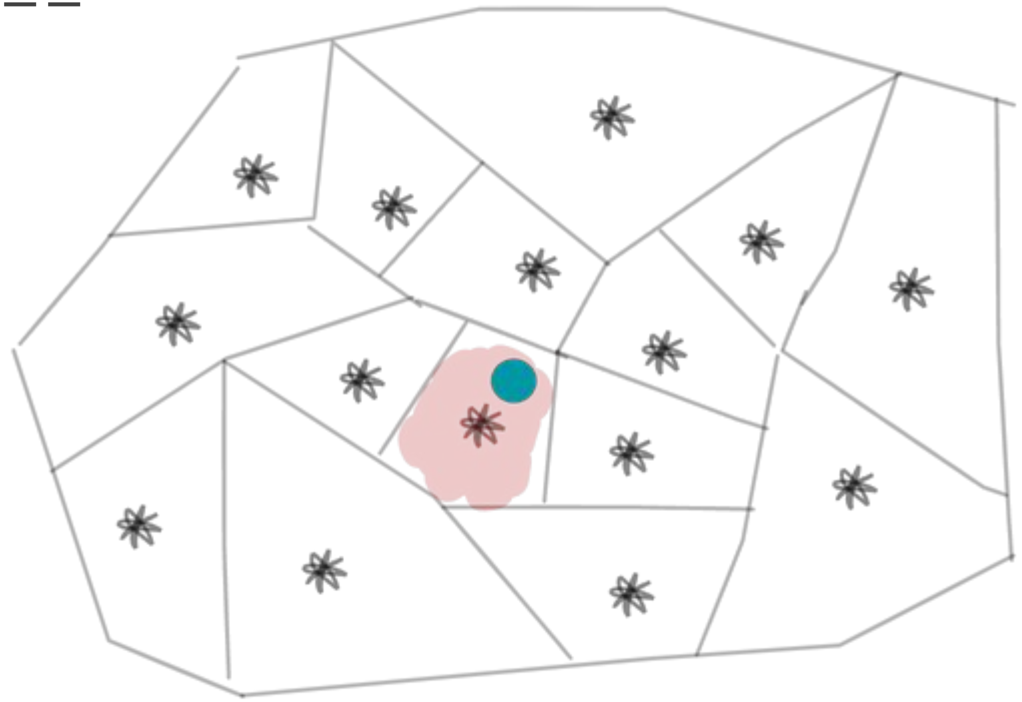
Inverted File Index: Building



- Built Voronoi Index with 15 centroids centroids

- Memory usage is not reduced
- But retrieval is faster

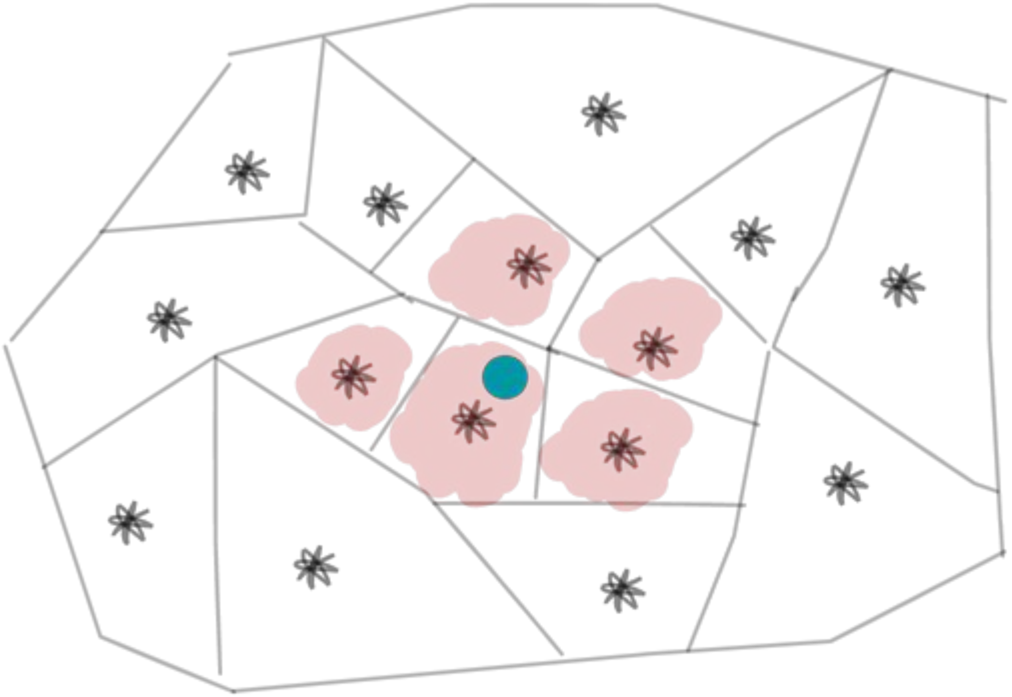
Inverted File Index: Searching



nprobe = 1

- Find the distance between query vector and all the centroids
- Query all the elements inside the n closest cluster determined by **nprobe** parameter

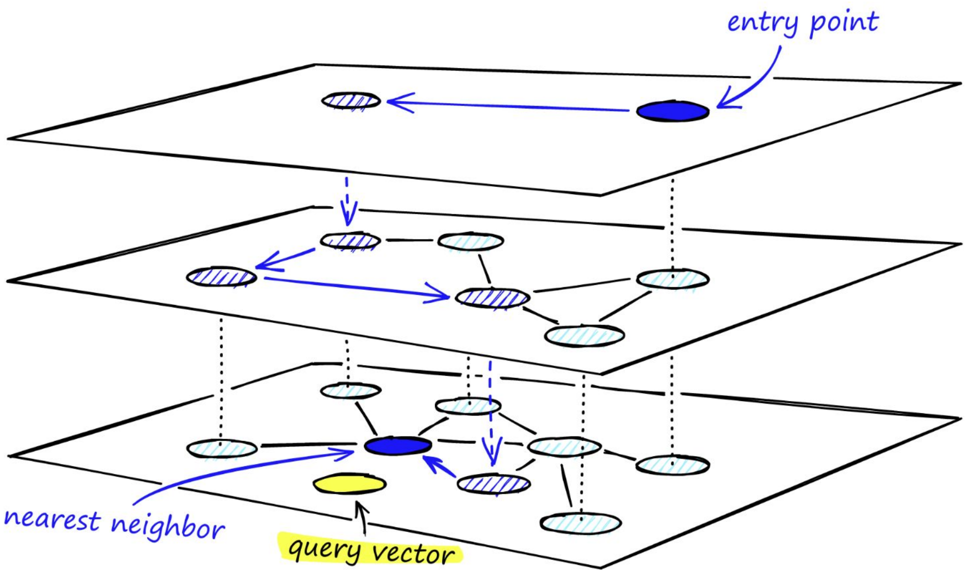
Inverted File Index: Searching



nprobe = 5

- Increasing the value of **nprobe**, improves recall but increases latency
- If nprobe=nlist, similar to flat index

Hierarchical Navigable Small Worlds (HNSW)



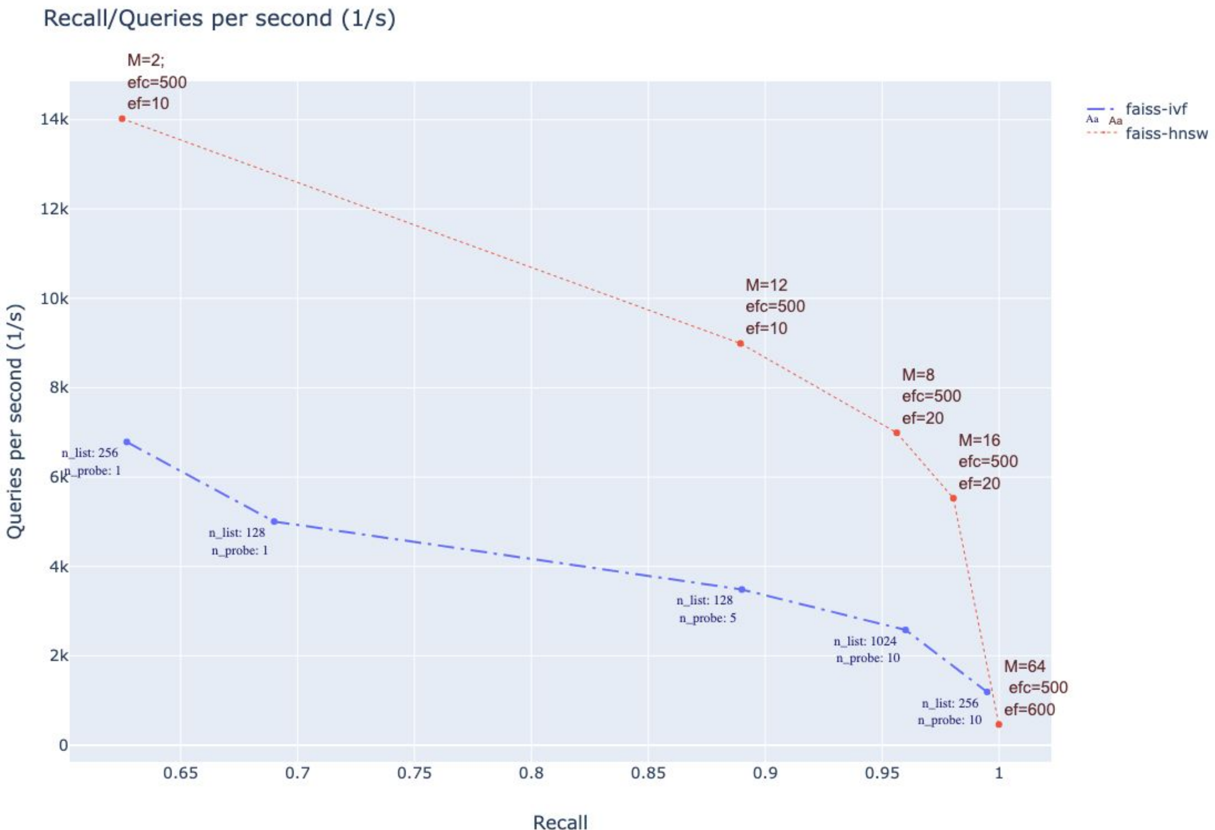
The search process through the multi-layer structure of an HNSW graph.

- Hierarchical Graph
- Every layer stores neighbors
- Earlier layers are sparse

- Common implementation in most production ANN databases

Image from https://www.pinecone.io/learn/hnsw/

ANN Benchmarks



Source:
ANN Benchmarks

Subset

Lot of ANN Option

FAISS

milvus

Weaviate

Pinecone

elastic

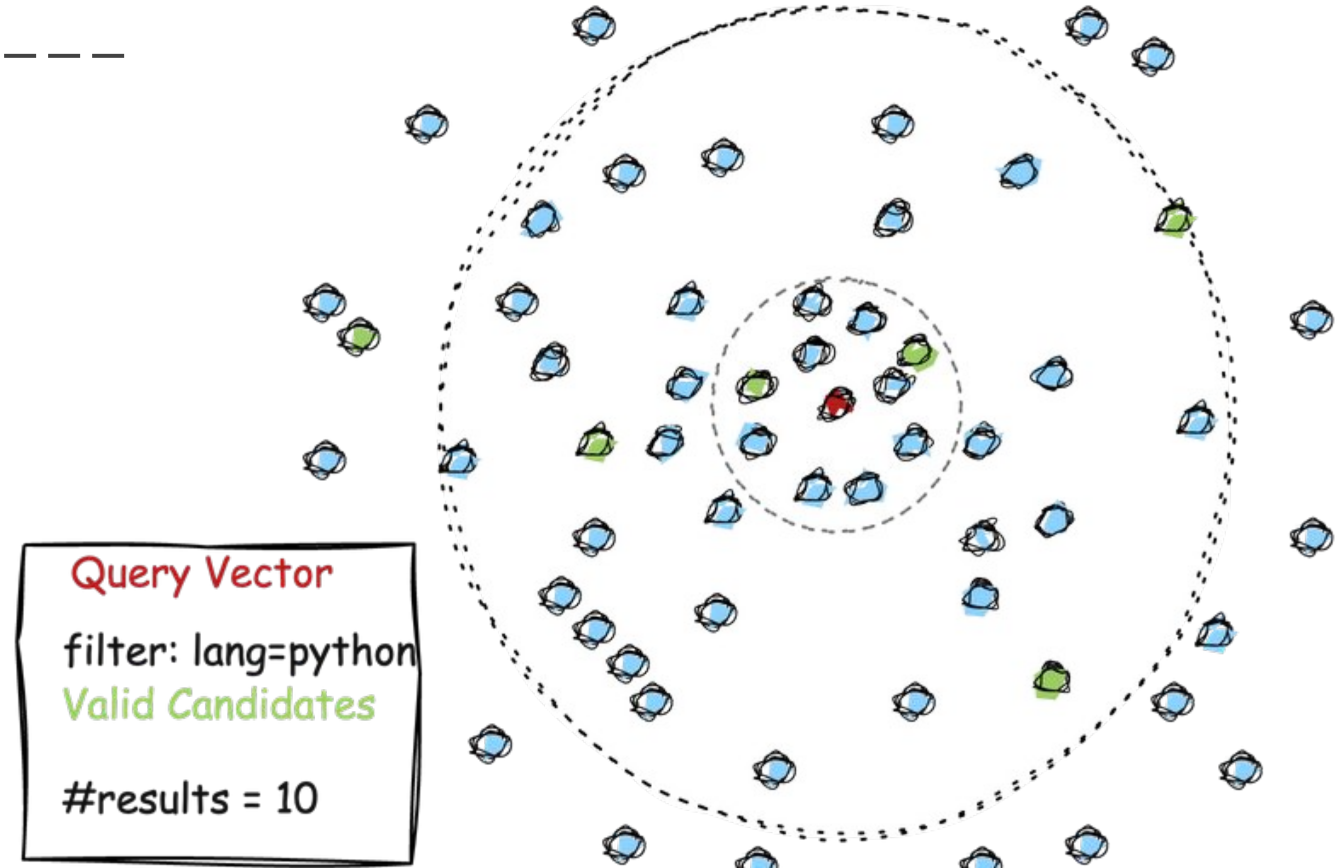
Matching Engine

vespa

Evaluation Considerations

- Managed vs Self-hosted
- Performance
- Update Embeddings / Partial Updates
- Metadata Filtering
- Filtering / Hybrid Retrieval
- Plugins

Filtering



Closest 10 candidates don’t meet our filter

Hybrid / Full Retrieval

Supports

elastic vespa

- In ES, **disjunction** of knn and bm25 match.
- The score of each hit is the sum of the knn and query scores. A boost can be specified

```
POST image-index/_search
{
  "query": {
    "match": {
      "title": {
        "query": "mountain lake",
        "boost": 0.9
      }
    }
  },
  "knn": {
    "field": "image-vector",
    "query_vector": [54, 10, -2],
    "k": 5,
    "num_candidates": 50,
    "boost": 0.1
  },
  "size": 10
}
```

ElasticSearch example of Hybrid Retrieval

Ex from Elastic Doc link

SCAN ME

Lab

Jupyter Hub: https://hub.np.training

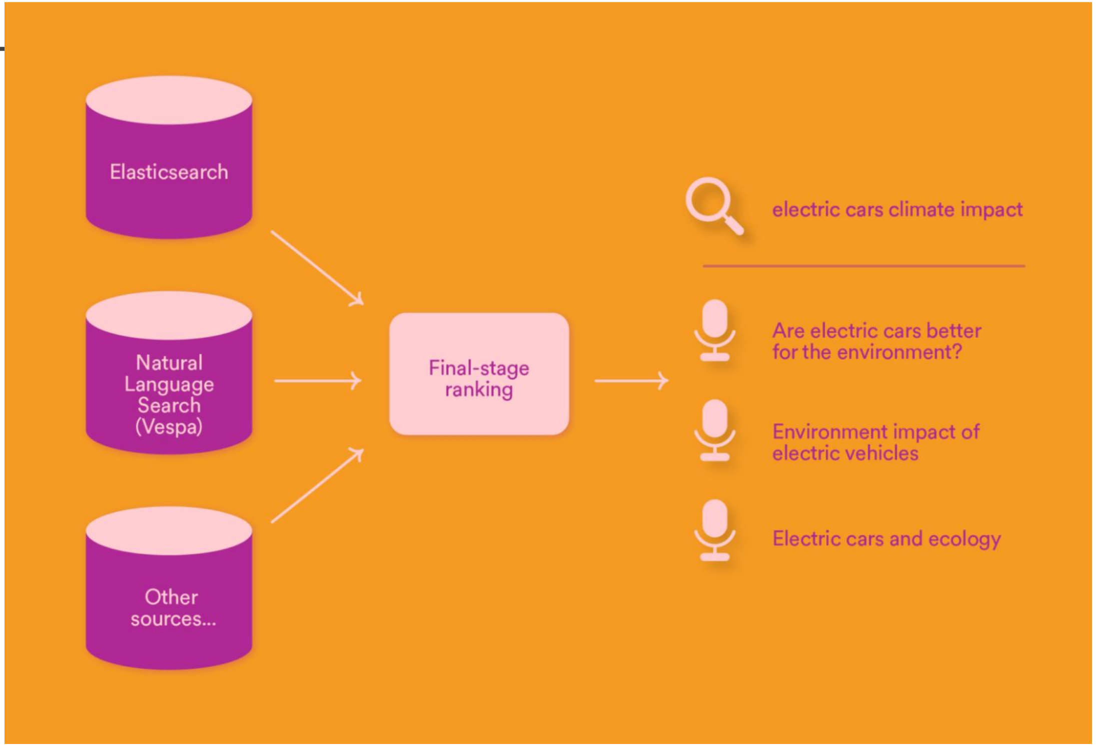
Repo:https://bit.ly/search-workshop-2022

Lab 3 Goals

- Build a flat ANN index
- Optimize retrieval by building an IVF Index

Conclusion

Multi Source Retrieval

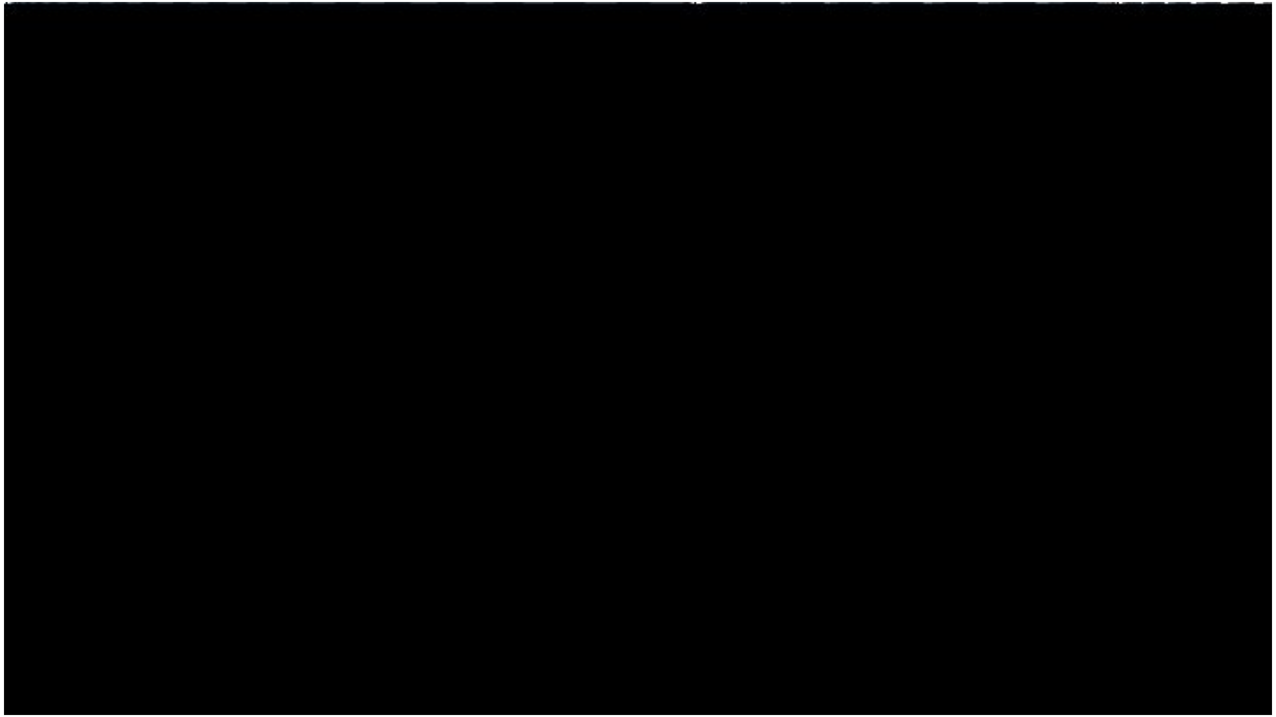


Spotify Podcast Episode Search

Retrieval from ES + ANN + other sources

Tamborrino, Alexandre . “Introducing Natural Language Search for Podcast Episodes - Spotify Engineering.” Introducing Natural Language Search for Podcast Episodes, March 17, 2022. https://engineering.atspotify.com/2022/03/introducing-natural-language-search-for-podcast-episodes/.

Multi Stage Retrieval and Ranking

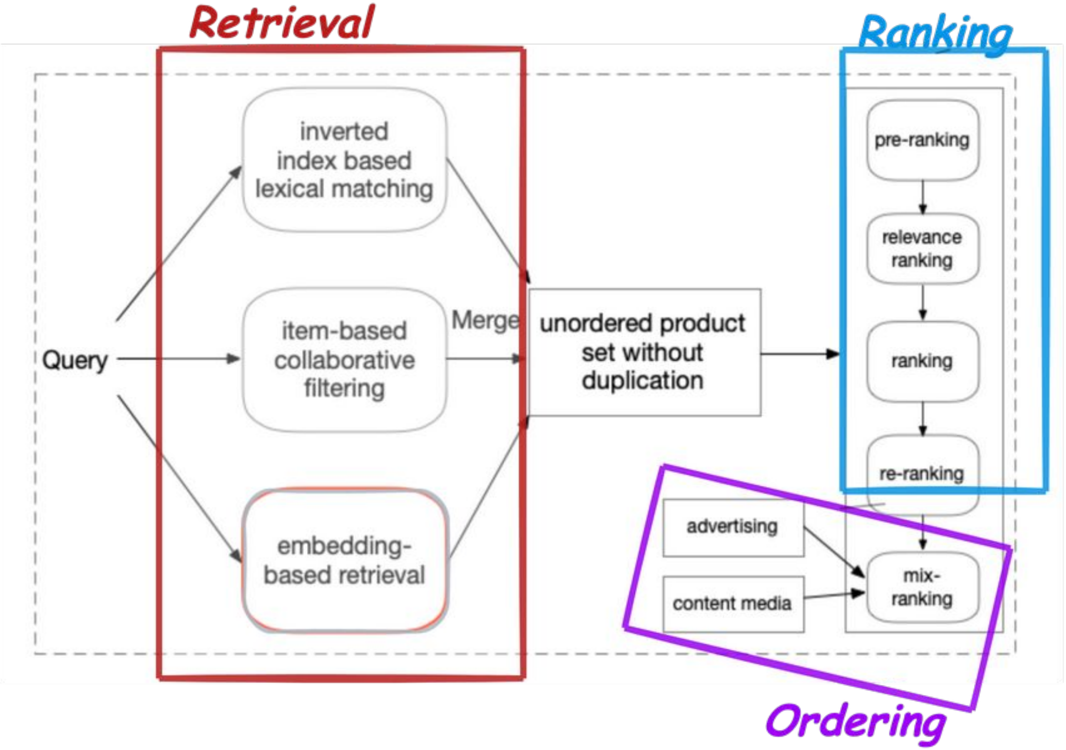


In 2019, Instagram Explore feed was made of

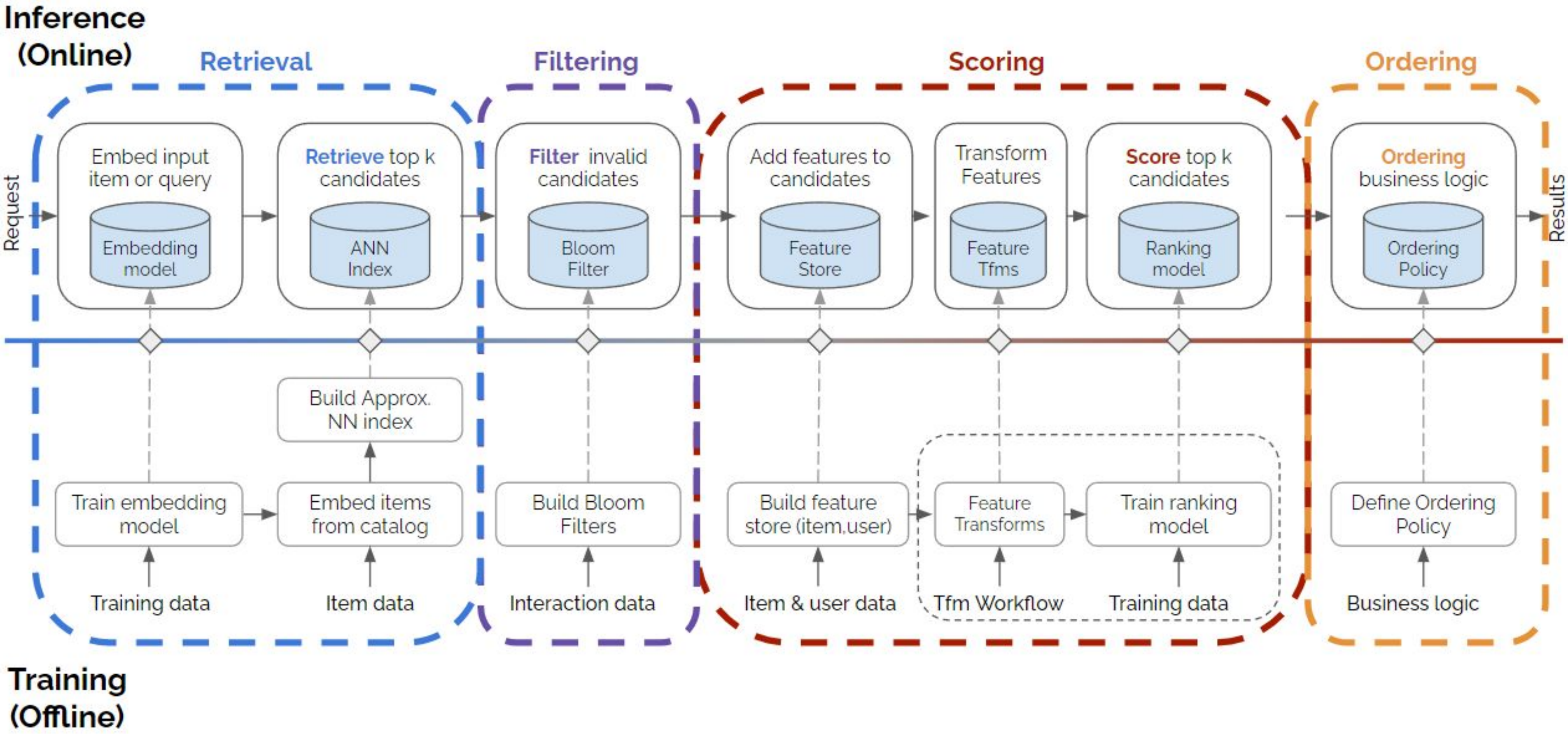
- Sparse Retriever: 500 candidates
- Lightweight NN: 150 ->50
- Deep Neural Network: 50 -> 25

Medvedev, Ivan, Haotian Wu, and Taylor Gordon. “Powered by AI: Instagram’s Explore Recommender System.” Powered by AI: Instagram’s Explore recommender system, November 2019. https://ai.facebook.com/blog/powered-by-ai-instagrams-explore-recommender-system/

Overview of Taobao Search



Li, Sen, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, και Qianli Ma. 'Embedding-based Product Retrieval in Taobao Search'. arXiv, 2021. https://doi.org/10.48550/ARXIV.2106.09297.

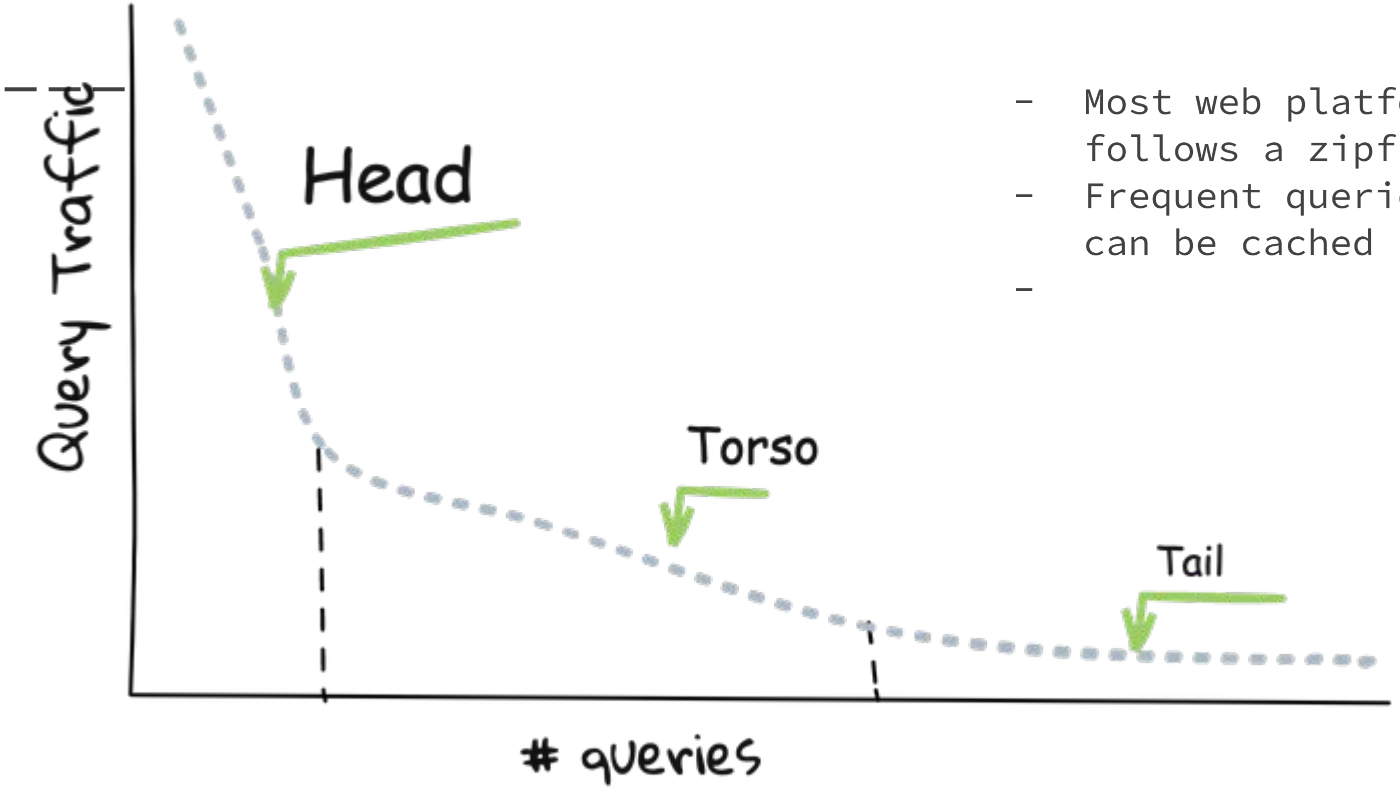


Oldridge, Even. “Recommender Systems, Not Just Recommender Models | by Even Oldridge | NVIDIA Merlin | Medium.” Medium, June 27, 2022. https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e.

	Retrieval	Filtering	Scoring	Ordering
Music Discovery	Find similar songs based on nearest neighbour search	Remove tracks users listened before	Predict likelihood that a user will listen to a song	Trade-Off between score, similarity, BPM, etc
Social Media	Find new posts in user's network	Remove posts from blocked and muted users	Predict likelihood that a user will interact with it	Change order that adjust posts are from different authors
Online Store	Find items which are usually co-purchased	Remove items which are out of stock	Predict likelihood that a user will purchase an item	Reorder items based on price points
Streaming Service	Find items based on different rows/shelves/topics	Remove items which are not available for user's country	Predict user's stream time per item	Organize recommendations to fit genre distributions

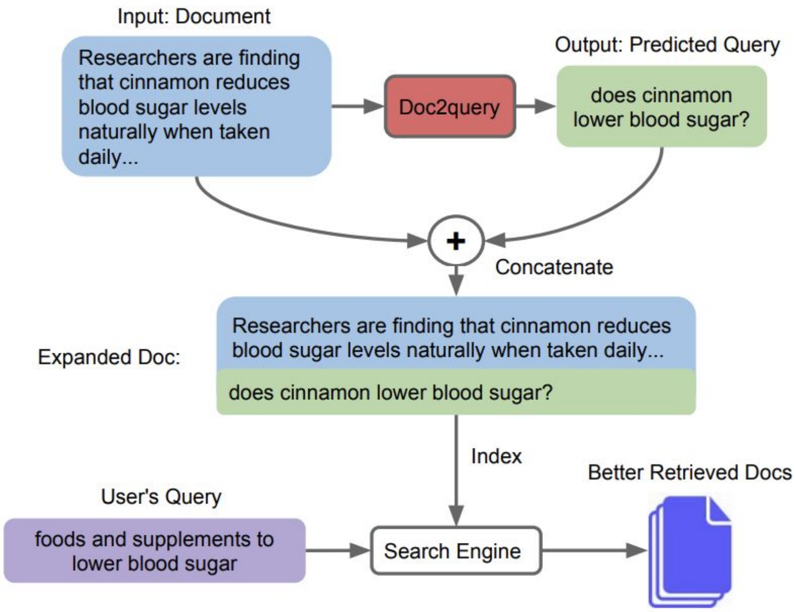
Oldridge, Even. “Recommender Systems, Not Just Recommender Models | by Even Oldridge | NVIDIA Merlin | Medium.” Medium, June 27, 2022. https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e.

Zipfian Distribution of Web Traffic



- Most web platforms traffic follows a zipfian distribution
- Frequent queries / head traffic can be cached
-

Enhancing Sparse Index: Doc2Query / Doc2T5Query



- Use a causal language model to generate additional text to add to documents when indexing.
- At retrieval time, use BM25

Nogueira, Rodrigo, Wei Yang, Jimmy Lin, και Kyunghyun Cho. 'Document Expansion by Query Prediction'. arXiv, 2019. https://doi.org/10.48550/ARXIV.1904.08375.

Benchmarking

Dataset	BM25	Dense (TAS-B)		BM25 on docT5query	BM25 + Cross Encoder
Baseline (MS MARCO)	0.228	0.408	-	0.338	0.413
Quora	0.789	0.835	-	0.802	0.825
DBPedia	0.313	0.384	-	0.331	0.409
-	-	-	-	-	-
TREC-COVID (Medical)	0.656	0.481	-	0.713	0.757
Signal-1M (Tweets)	0.330	0.289	-	0.325	0.338

Do models trained on MS MARCO work for different datasets ?

Dense Retrieval performs well on similar domain

COVID/Tweets are out of domain

BM25 + DR perform best

Cost to Serve

Model	Dimension	Latency (CPU)	Latency (GPU)	Index Size
BM-25	-	20 ms	-	0.4 GB
docT5Query	-	30 ms	-	0.4 GB
Dense Embedding Passage Retrieval	768	230 ms	19 ms	3 GB
BM25 + Cross Encoder	-	6100 ms	450 ms	0.4 GB

Estimated average retrieval latency and index sizes

Dataset: DBPedia (1 million docs)

Lower Latency and Index Size is preferred

CPU: 8 core Intel Xeon Platinum 8168 CPU @ 2.70GHz

GPU: 1 Nvidia Tesla V100

Thakur, Nandan, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, και Iryna Gurevych. 'BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models'. arXiv, 2021. https://doi.org/10.48550/ARXIV.2104.08663.

Resources

- Faiss Missing Manual (Pinecone)
- Natural Language Processing (NLP) for Semantic Search (Pinecone)

- CIKM 2021 Tutorial: IR From Bag-of-words to BERT

- Haystack US 2021 - Semantic Product Search - Vector Search for E-Commerce - Simon Hughes

Hard and Soft Negative Mining

- Start training with positive and negative data (soft negative) points.
- Best negative samples are the ones similar to anchor based on current embeddings. (Hard Negative)
- Use Hard Negative to further optimize the embeddings.
- Two strategies
  - Local for mini-batch
  - Global
- Carefully select the hard negatives
  - Anchor Query = cordless phone
  - False Positives = corded phone