Weaviate

Multi-Vector embeddings revolution? or evolution?

Marcin Antas, Roberto Esposito

Who am I?

Weaviate Core Engineer

over 17 years of experience

almost 5 years in AI space

Preferred languages: Go, Python, Java, Scala, TS

working on Open Source AI-first Weaviate DB

Who am I?

Weaviate Research Engineer

Applied Research Team

Past experience:

Research on Approximate Nearest Neighbor and Compression

Agenda

1. Embeddings models
2. Vector Databases – How does it work?
3. MUVERA Multi-Vector encoding
4. Demo

Embeddings models

Embedding models

Embeddings are vector representations of data.

Model: Snowflake/snowflake-arctic-embed-m
Language: English   Dimensionality: 768

"Black cat sitting on the street on a rainy day at night"

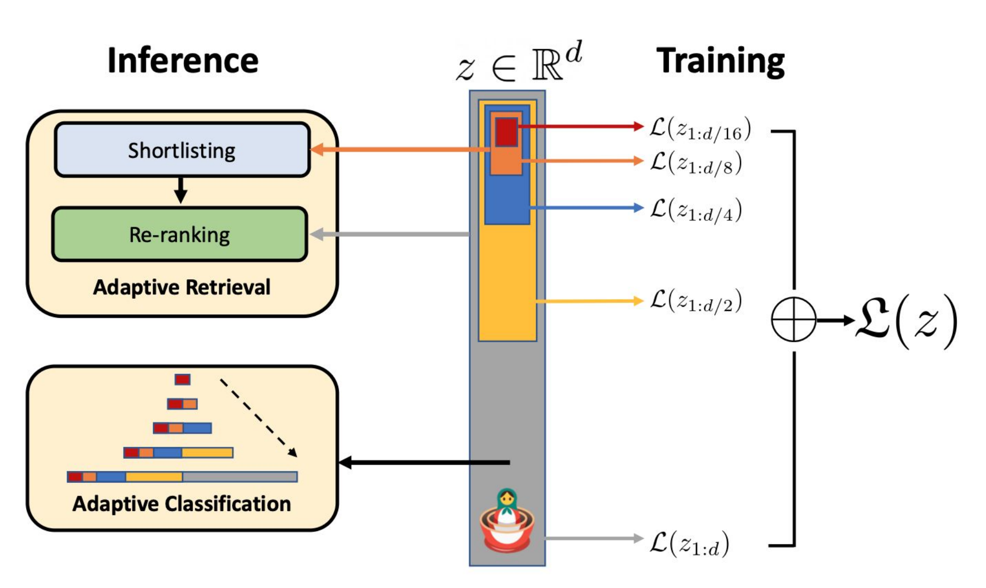
[0.02460668,-0.027135728,-0.0029105705, … ,-0.018872168]

Text embeddings models turn text into a vector representation.

Embedding models

Matroyshka Embeddings models offer multiple vector dimensions

Model: nomic-ai/nomic-embed-text-v1.5
Language: English **Dimensionality:** 64, 128, 256, 512, 768



Embedding models

AI Embeddings models:

- OpenAI V3 text embedding
- Google Embedding Gemma 300m
- Cohere Embed 4
- Snowflake Arctic Embed
- ModernVBERT Embed
- BAAI BGE-M3
- Jina AI Embeddings V4

Embedding models

Multi-Vector (ColBERT) embeddings

ColBERT produces as many embeddings as there are tokens (words) in a sentence, instead of producing one embedding for sentence.

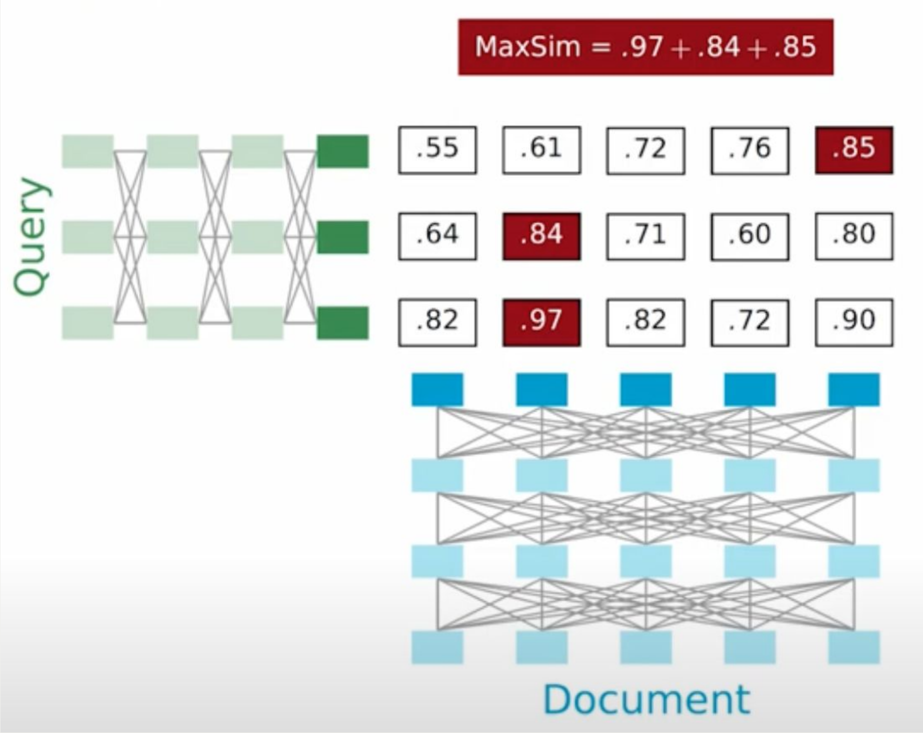
“Black cat sitting on the street on a rainy day at night”

v0 v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11

Sentence embedding: [v0,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11]

Embedding models

How to search data using ColBERT embeddings?

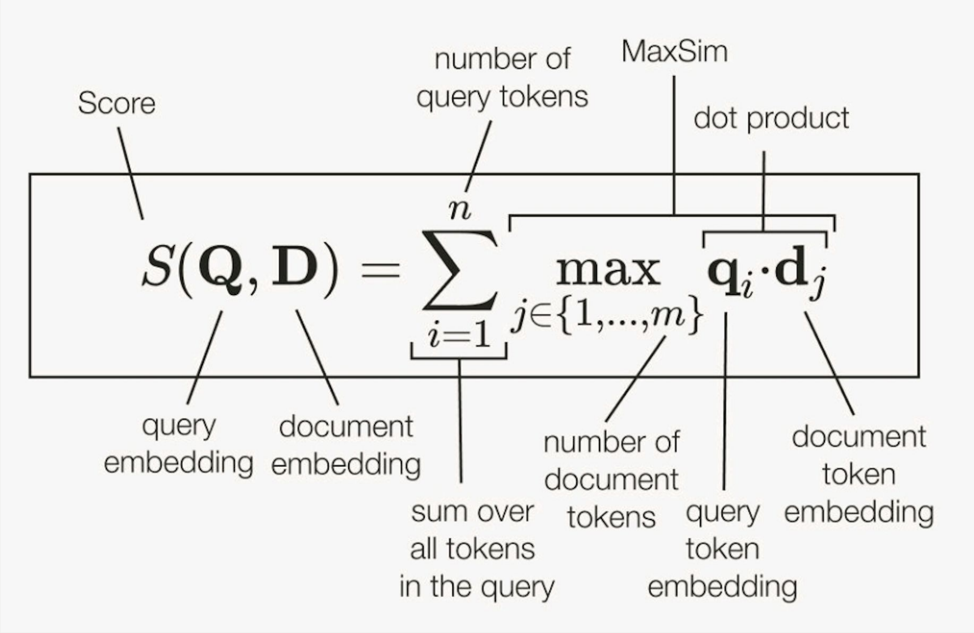


Source: Stanford University NLU online course

Embedding models

ColBERT embeddings
Late interaction

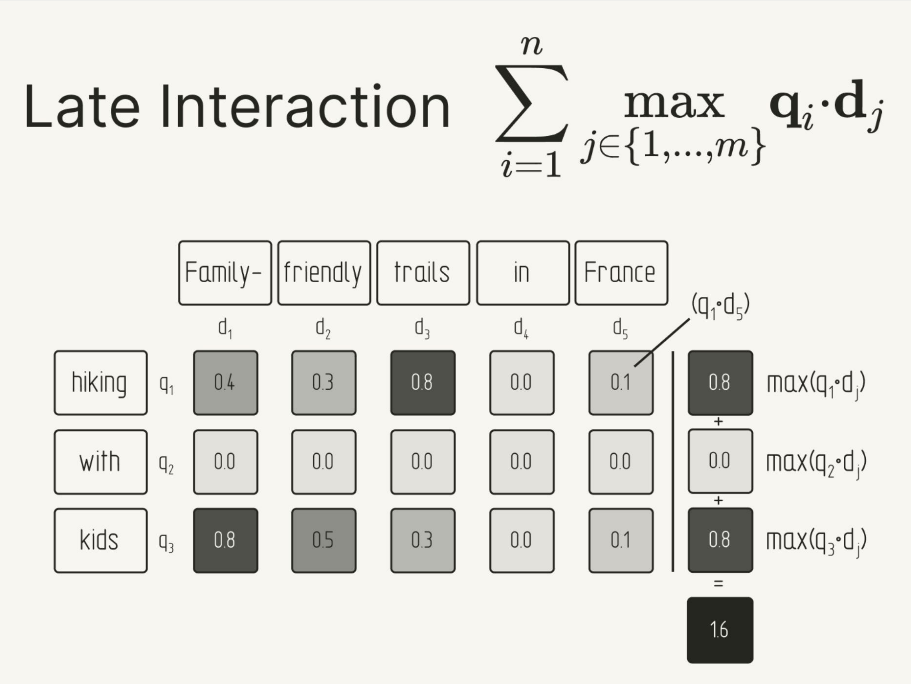
Contextualized Late Interaction over BERT

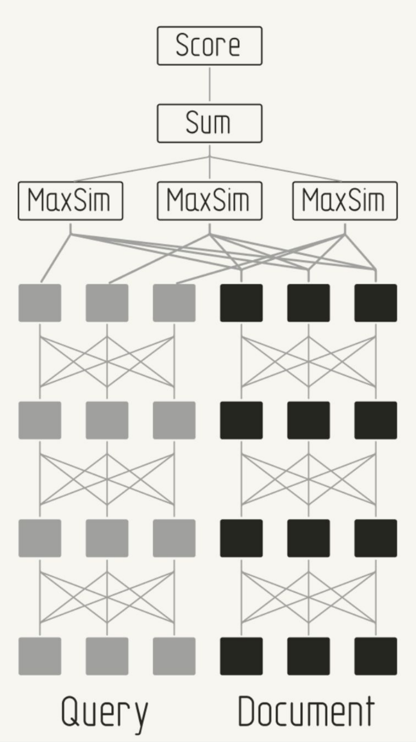


Embedding models

ColBERT embeddings

Late interaction





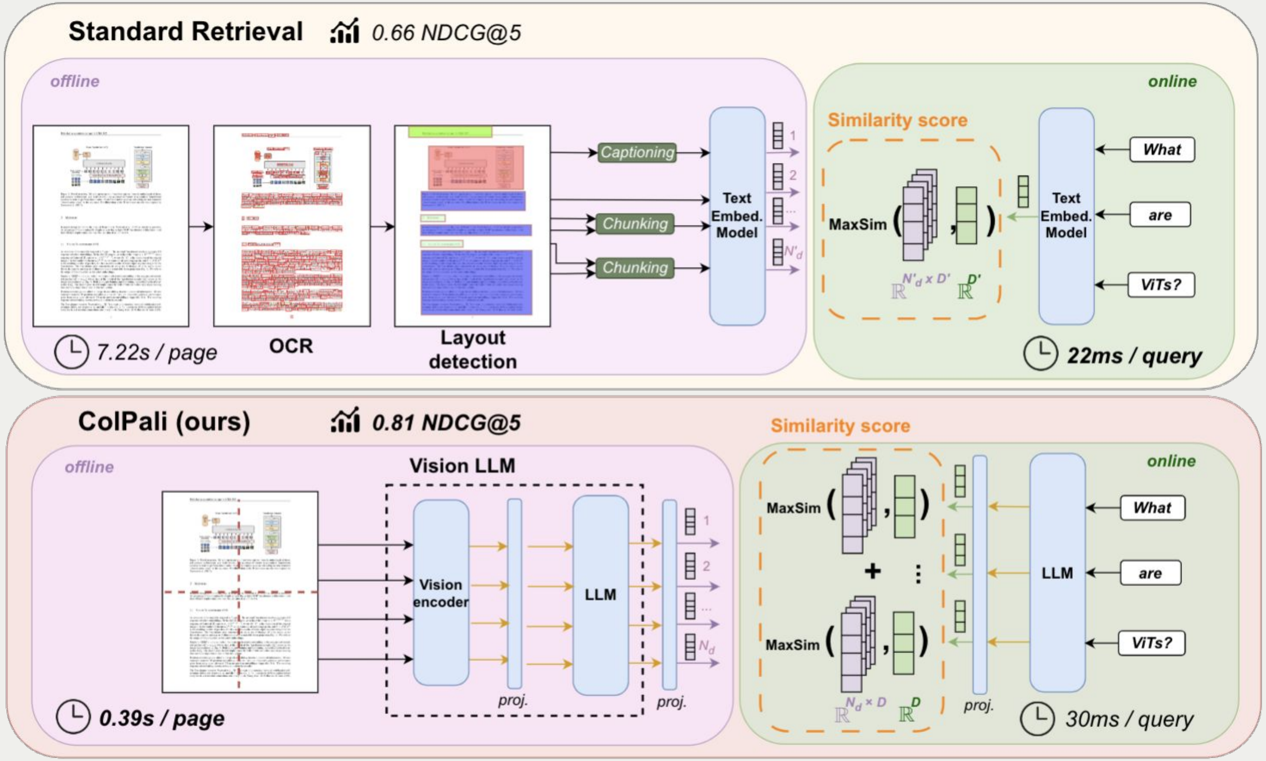
Embedding models

Multi-Vector vision embeddings models

ColPali
(PaliGemma)

ColQwen2
(Qwen2-VL)

ColNomic
(Fine tuned Qwen2.5-VL)



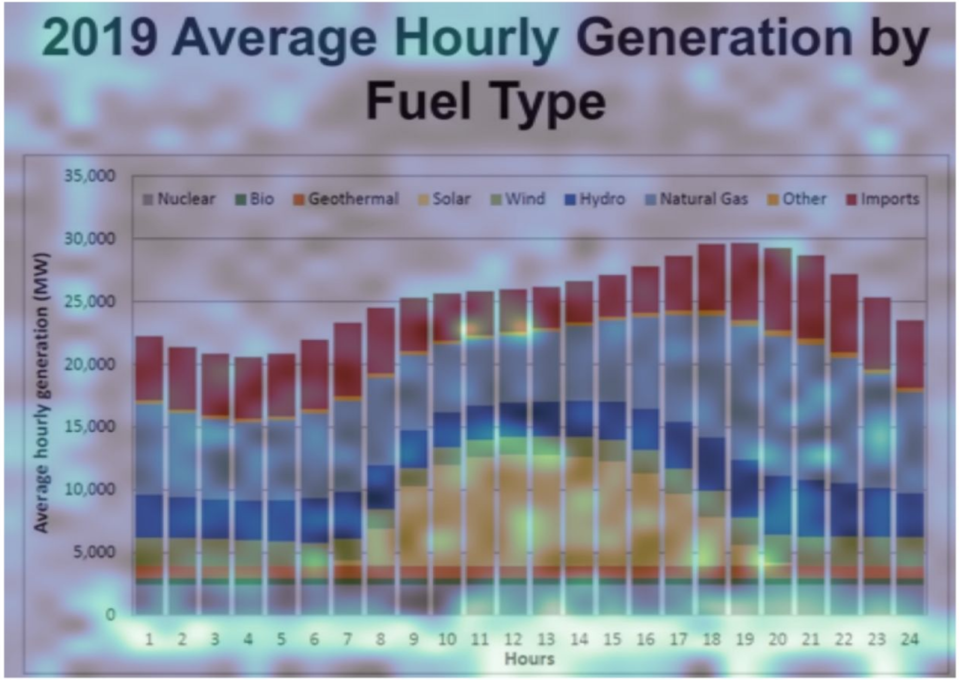
Embedding models

Multi-Vector vision embeddings models

ColPali
(PaliGemma)

ColQwen2
(Qwen2-VL)

ColNomic
(Fine tuned
Qwen2.5-VL)



Query: "Which hour of the day had the highest overall eletricity generation in 2019?"

Embedding models

Multi-Vector vision embeddings models

When to use Multi-Vector vision embedding models?

- PDF documents and research papers
- Screenshots of applications and websites
- Visually rich content where layout matters
- Multilingual documents where visual context is important

Vector databases – How does it work?

Vector representations of data

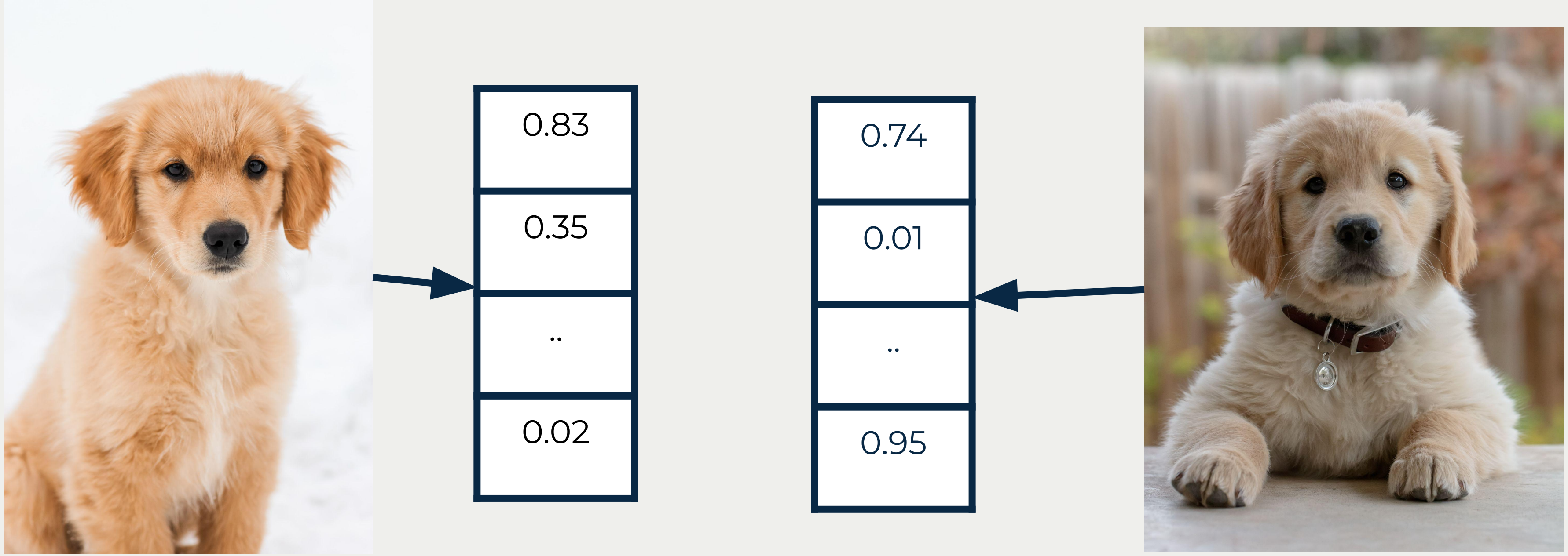
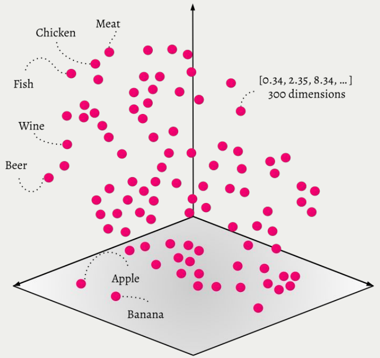


Photo by Shayna Douglas on Unsplash

Photo by Bill Stephan on Unsplash

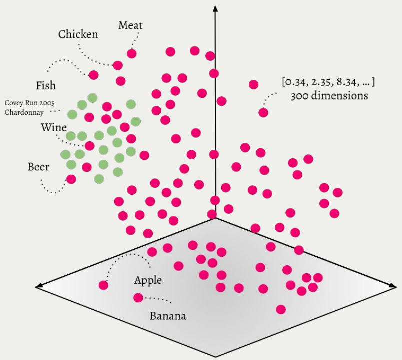
Vector databases – How does it work?

Pretrained ML model



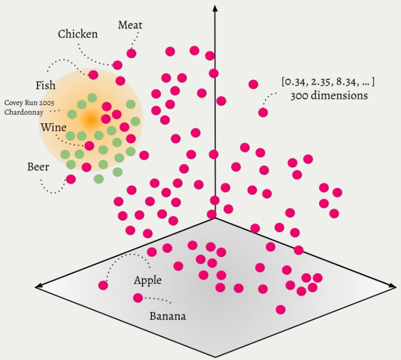
(e.g. NLP)

Your own data



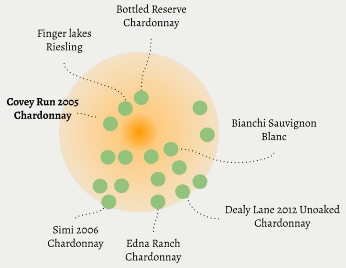
(e.g. Wines)

Your search query



(e.g. Wine for seafood)

Results

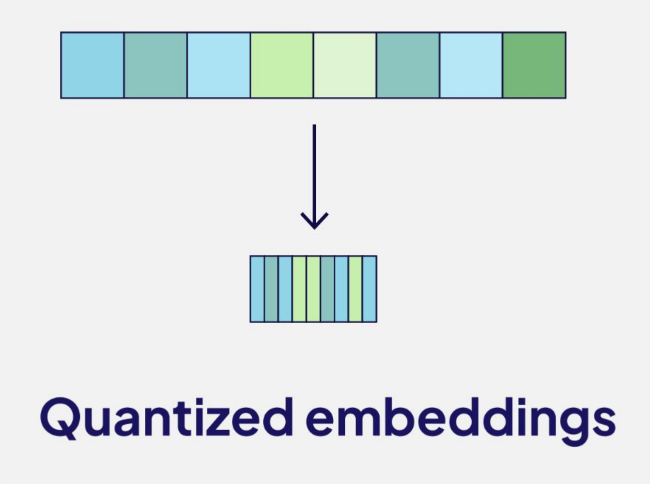


(e.g. Chardonnay)

Vector databases – How does it work?

Vector Index types:

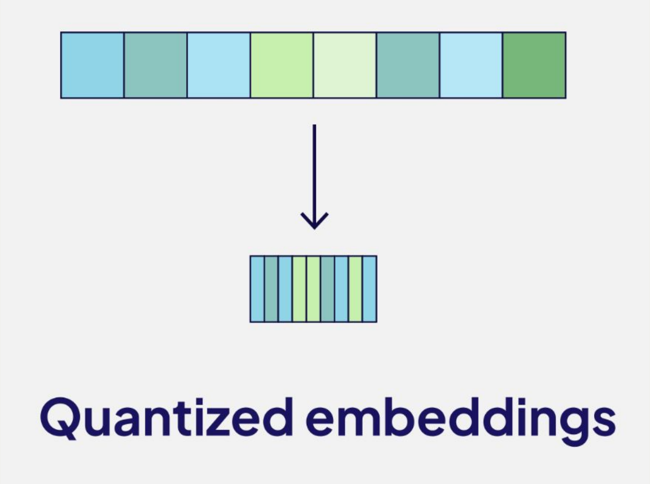
- HNSW / Flat (on disk)
  - PQ – Product Quantization
  - BQ – Binary Quantization
  - SQ – Scalar Quantization
  - RQ – Rotational Quantization
- HNSW Multi Vector
  - MUVERA



Vector databases – How does it work?

Vector Index types:

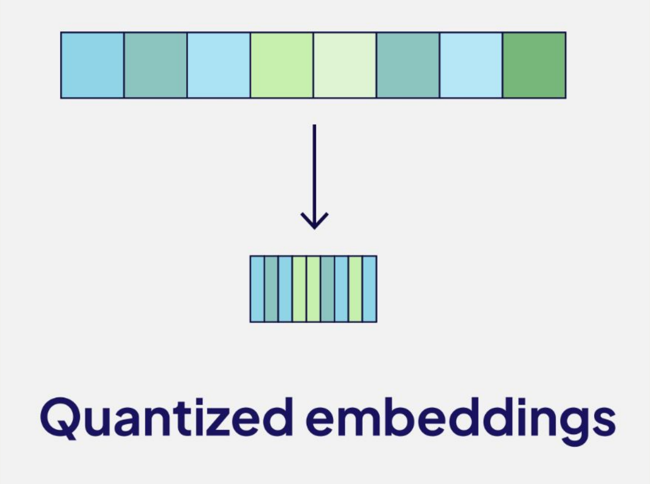
- HNSW / Flat (on disk)
  - PQ – Product Quantization
  - BQ – Binary Quantization
  - SQ – Scalar Quantization
  - RQ – Rotational Quantization
- HNSW Multi Vector
  - MUVERA



Vector databases – How does it work?

Vector Index types:

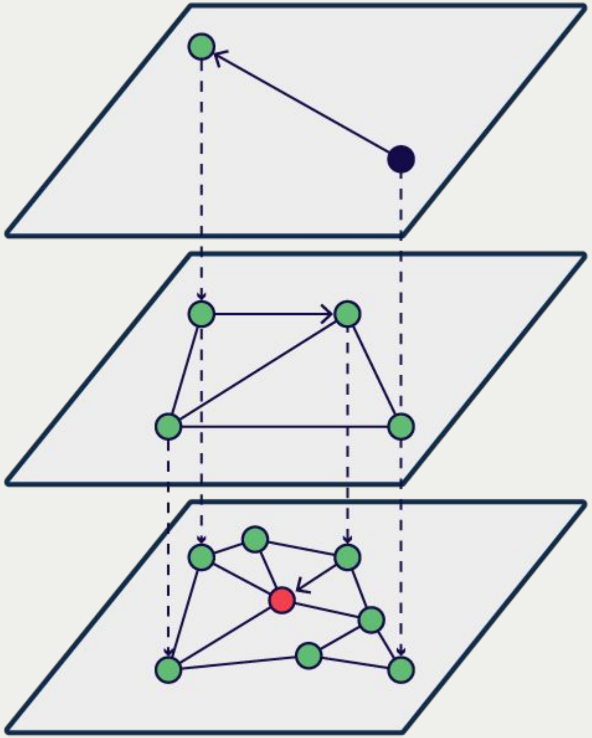
- HNSW / Flat (on disk)
  - PQ – Product Quantization
  - BQ – Binary Quantization
  - SQ – Scalar Quantization
  - RQ – Rotational Quantization
- HNSW Multi Vector
  - MUVERA



Vector databases – How does it work?

Vector Index types:

- HNSW / Flat (on disk)
  - PQ – Product Quantization
  - BQ – Binary Quantization
  - SQ – Scalar Quantization
  - RQ – Rotational Quantization
- HNSW Multi Vector
  - MUVERA



MUVERA Multi-Vector encoding

MUVERA Multi-Vector encoding

MUVERA:

- encodes multi vector into single vector called FDE
- each FDE product approximates MaxSim score

arXiv > cs > arXiv:2405.19504

Search

Help | A

Computer Science > Data Structures and Algorithms

[Submitted on 29 May 2024]

MUVERA: Multi-Vector Retrieval via Fixed Dimensional Encodings

Laxman Dhulipala, Majid Hadian, Rajesh Jayaram, Jason Lee, Vahab Mirrokni

Neural embedding models have become a fundamental component of modern information retrieval (IR) pipelines. These models produce a single embedding $x \in \mathbb{R}^d$ per data-point, allowing for fast retrieval via highly optimized maximum inner product search (MIPS) algorithms. Recently, beginning with the landmark ColBERT paper, multi-vector models, which produce a set of embedding per data point, have achieved markedly superior performance for IR tasks. Unfortunately, using these models for IR is computationally expensive due to the increased complexity of multi-vector retrieval and scoring.

In this paper, we introduce MUVERA (MUlti-VEctor Retrieval Algorithm), a retrieval mechanism which reduces multi-vector similarity search to single-vector similarity search. This enables the usage of off-the-shelf MIPS solvers for multi-vector retrieval. MUVERA asymmetrically generates Fixed Dimensional Encodings (FDEs) of queries and documents, which are vectors whose inner product approximates multi-vector similarity. We prove that FDEs give high-quality ϵ-approximations, thus providing the first single-vector proxy for multi-vector similarity with theoretical guarantees. Empirically, we find that FDEs achieve the same recall as prior state-of-the-art heuristics while retrieving 2-5× fewer candidates. Compared to prior state of the art implementations, MUVERA achieves consistently good end-to-end recall and latency across a diverse set of the BEIR retrieval datasets, achieving an average of 10% improved recall with 90% lower latency.

MUVERA Multi-Vector encoding

Main Steps

1. Space partitioning
2. Dimensionality reduction
3. Repeat 1 & 2 multiple times

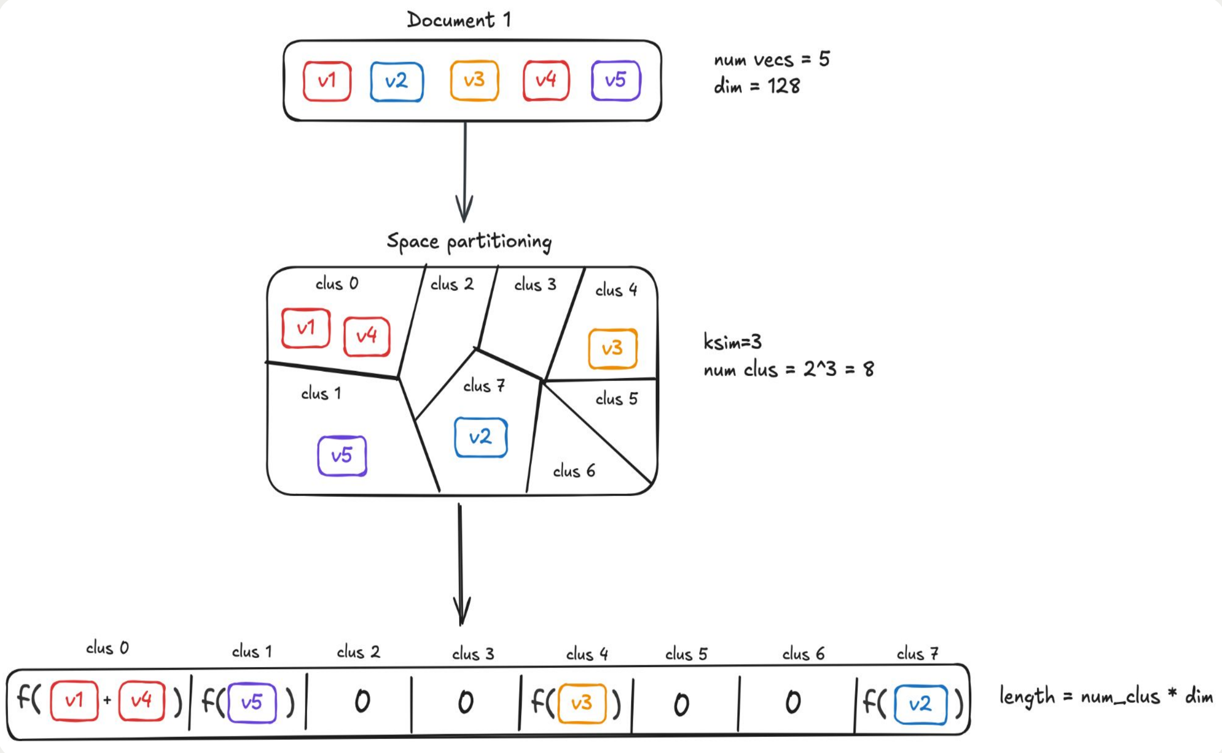
Parameters

- kSim: 4
- dProj: 16
- nReps: 10

MUVERA Multi-Vector encoding

MUVERA:

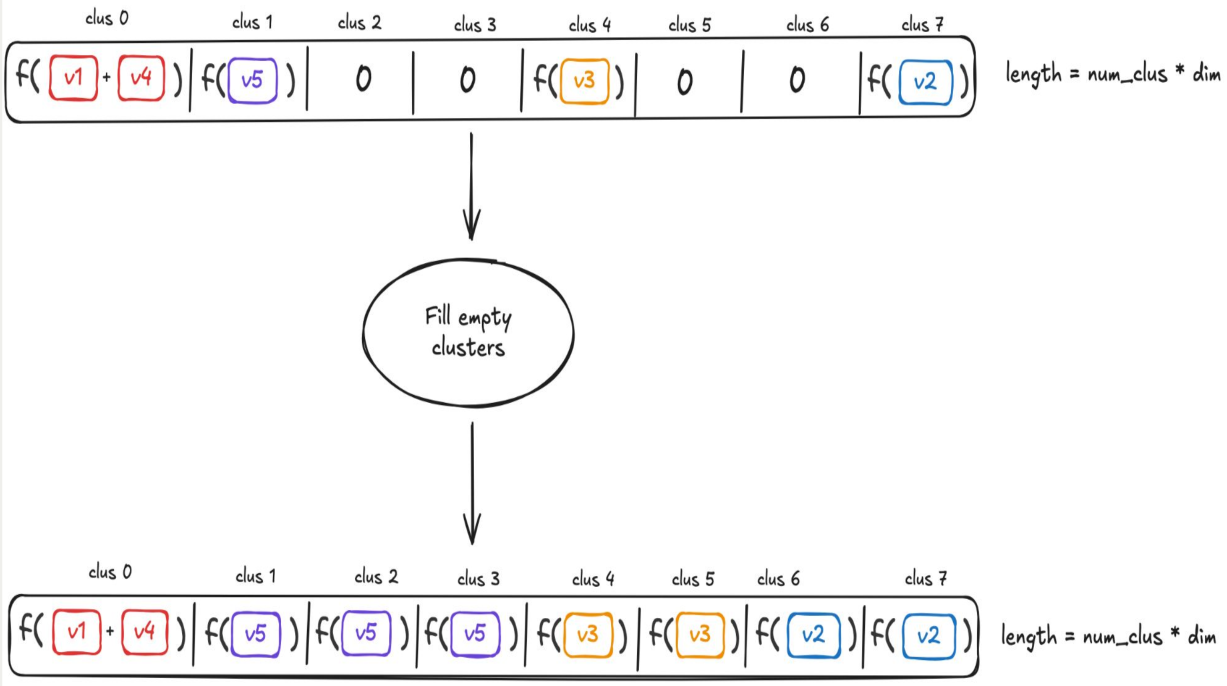
space partitioning uses SimHash based on Locality Sensitive Hashing



MUVERA Multi-Vector encoding

MUVERA:

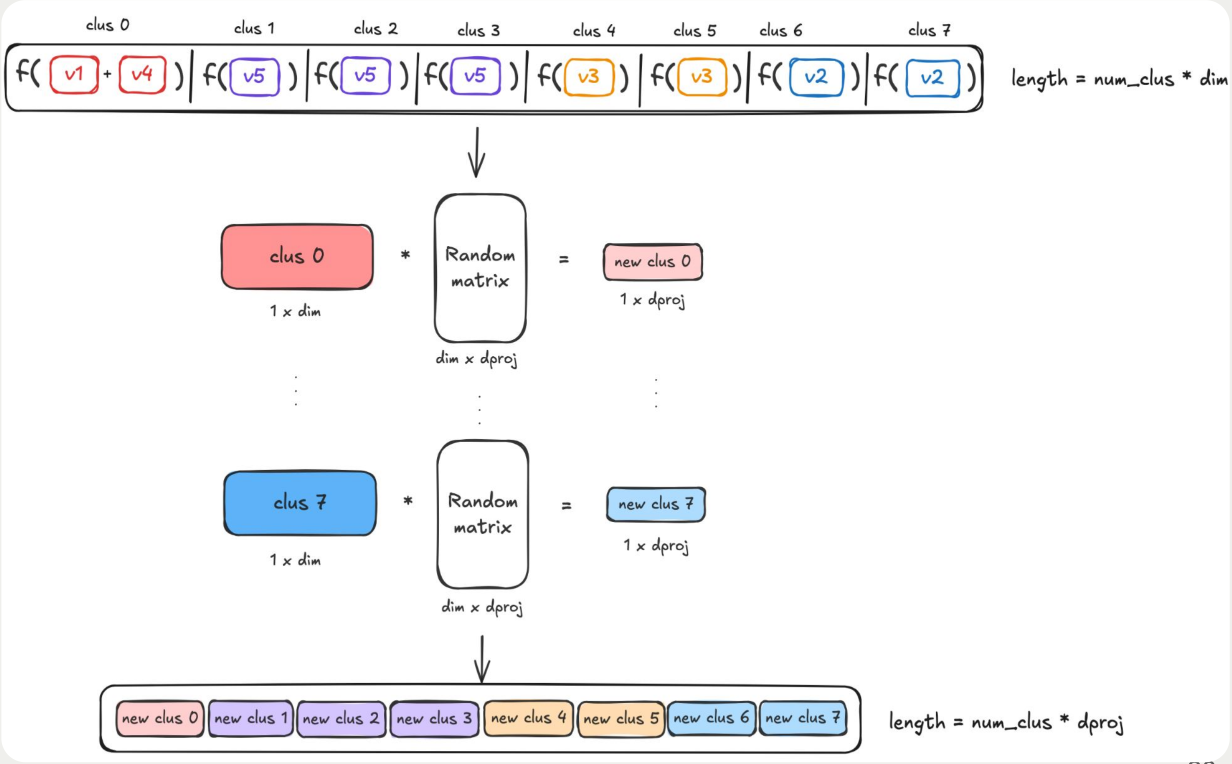
filling empty cluster during document encoding



MUVERA Multi-Vector encoding

MUVERA:

dimensionality
reduction uses
random matrices
to project dimensions



MUVERA Multi-Vector encoding

MUVERA:

pros:

- improved import times
- reduced memory requirements (smaller index)
- faster QPS

cons :

- worse recall (precision search)

MUVERA Multi-Vector encoding

MUVERA:

pros:

- improved import times
- reduced memory requirements (smaller index)
- faster QPS

cons :

- worse recall (precision search)
    - rescoring is the way of fixing the recall

Demo

Demo

Multi-Vector Vision models

1. Weaviate – v1.35
2. ColBERT vision model
   ColQwen2.5

Demo

Multi-Vector Vision models

AI powered OCR pipeline:

Demo

Multi-Vector Vision models

AI powered OCR pipeline:

1. Extract document page as an image

Demo

Multi-Vector Vision models

AI powered OCR pipeline:

1. Extract document page as an image
2. Vectorize image with Multi-Vector embeddings vision model

Demo

Multi-Vector Vision models

AI powered OCR pipeline:

1. Extract document page as an image
2. Vectorize image with Multi-Vector embeddings vision model
3. Store Multi-Vector embeddings in Vector DB using MUVERA encoding

Demo

Multi-Vector Vision models

AI powered OCR pipeline:

1. Extract document page as an image
2. Vectorize image with Multi-Vector embeddings vision model
3. Store Multi-Vector embeddings in Vector DB using MUVERA encoding
4. All set up!

Connect with us!

weaviate.io

weaviate/weaviate

weaviate_io

Thank you!

More efficient multi-vector embeddings with MUVERA

June 5, 2025 · 16 min read

Roberto Esposito
Research Engineer

Joon-Pil (JP) Hwang
Educator