Cairo University
Faculty of Engineering

Department of Computer
Engineering

Phase 1

ADB

Submitted to:

Eng / Abdelrahman Kaseb

Name	ID
Ahmed Sabry	9202119
Eman Mohamed	9202351
Zeinab Moawad Fayez Hassan	9202611
Menatalh Hossamalden	9203998

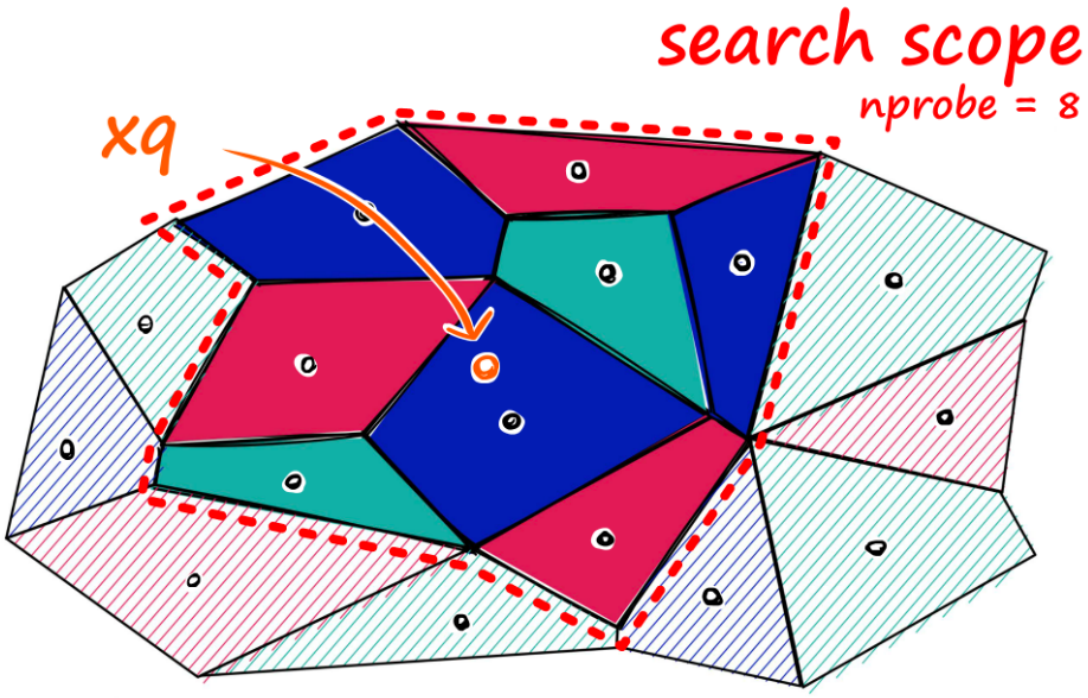
Indexer at size 10K - 1M: IVF

Description:

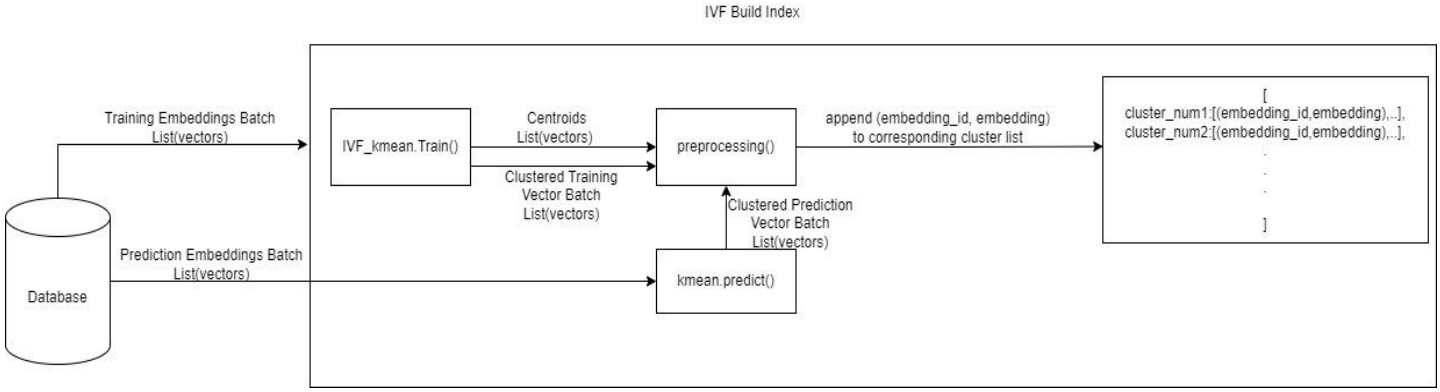
Inverted File Indexer:

- Training: convert space into subspaces using clustering methods (kmeans) And assign each subspace representative “centroids”
- Searching: get nearest nprobe centroids to query vector and apply flat index in each subspace to get top_k nearest vectors to query submitted.

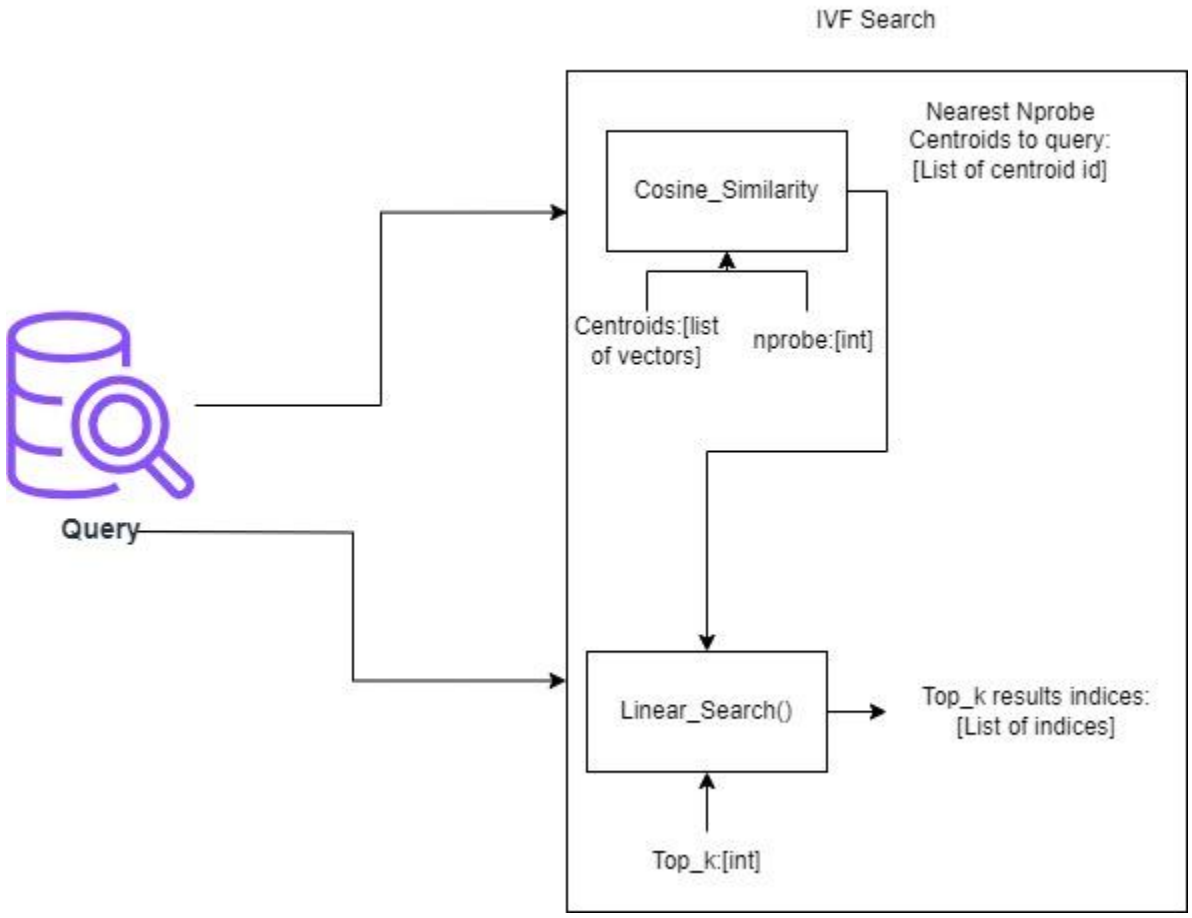
Architecture:



Indexing System:



Search System:



Reason:

1. High retrieval efficiency: The IVF index is designed to efficiently retrieve nearest neighbors without searching in whole space
2. Memory efficiency: It loads only vectors assigned to centroids
3. Query time control: It allows for controlling the query time by adjusting the number of inverted cells used in the index and # of centroids to search. By tuning this parameters, we can balance the search accuracy and the query speed according to specific requirements.

Results of our implementation:

algorithm	Parameter 1	Size	Recall out of 100	Time in s
IVF	Nprobe = 2, centroids = 16, iter = 32	10k	59	**0.003**
IVF	**Nprobe = 4, centroids = 16, iter = 32**	**10k**	**100**	**0.006**
IVF	Nprobe = 1, centroids = 16, iter = 32	100k	59	**0.049**
IVF	Nprobe = 4, centroids = 16, iter = 32	100k	100	**0.18**
IVF	Nprobe = 2, centroids = 32, iter = 32	100k	65	**0.049**
IVF	**Nprobe = 8, centroids = 32, iter = 32**	**100k**	**100**	**0.19**
IVF	Nprobe = 4, centroids = 64, iter = 32	100k	85	**0.049**
IVF	Nprobe = 16, centroids = 64, iter = 32	100k	100	**0.195**
IVF	Nprobe = 8, centroids =128, iter = 32	100k	90	**0.050**
IVF	**Nprobe = 32, centroids = 128, iter = 32**	**100k**	**100**	**0.19**
IVF	Nprobe = 16, centroids =256, iter = 32	500k	90	**0.308**
IVF	Nprobe =64, centroids = 256, iter = 32	500k	100	**1.1**
IVF	**Nprobe =32 , centroids = 512, iter = 32**	**500k**	**99**	**0.25**
IVF	**Nprobe =64 , centroids = 1024, iter = 32**	**1M**	**100**	**0.8**

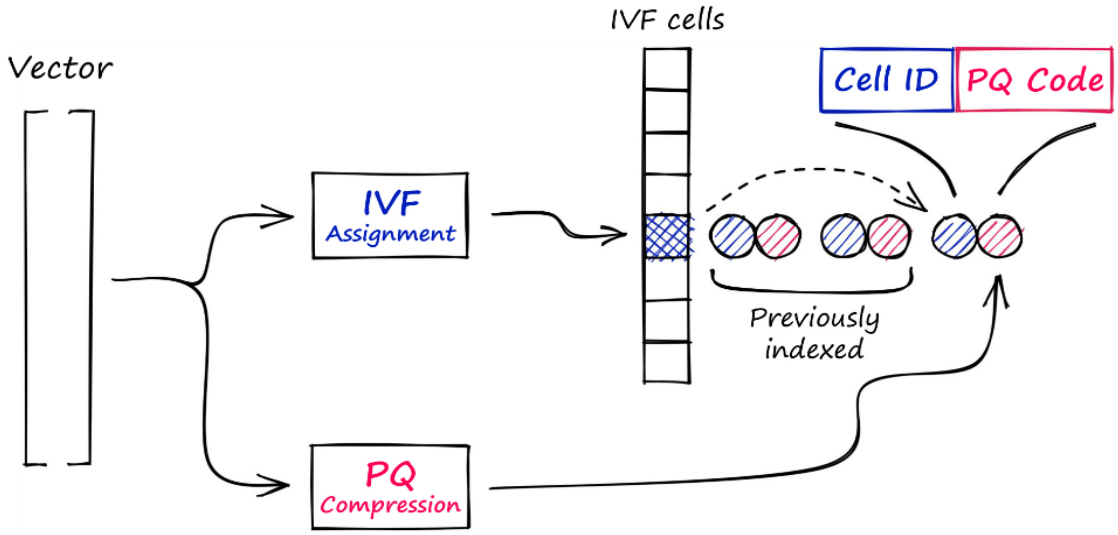
Indexer at size 1M - 20M : **IVF-ADC (possible solution #1)**

Description

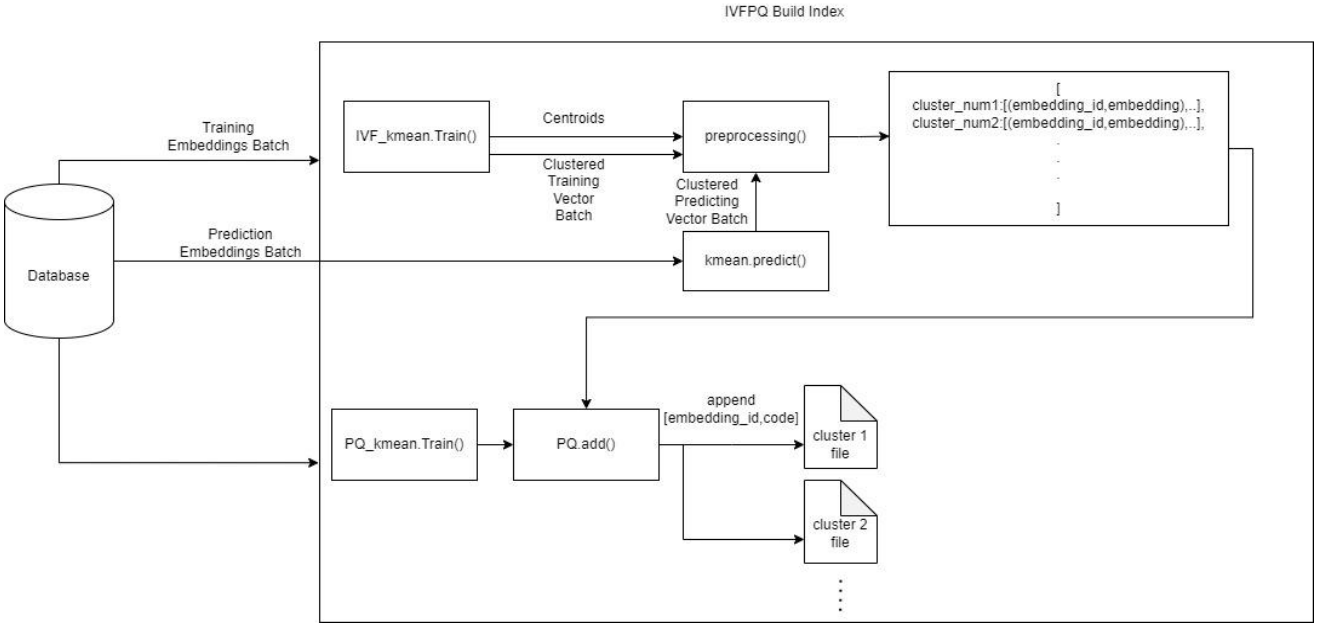
Inverted File Asymmetric Data Computation:

- Training
  - IVF: convert space into subspaces using clustering methods (kmeans) And assign each subspace representative “centroids”
  - PQ: divide vector to m subvector , for each subvector apply kmeans clustering to get 2**nbits centroids “IDs” that used to quantize vector to m-dimension vector
- Searching: get nearest nprobe centroids to query vector and apply flat index on quantized vectors using ADC in each subspace to get 100 top_k nearest vectors to query submitted, then apply flat index using original vectors on 100 top_k to get top_k.

Architecture:



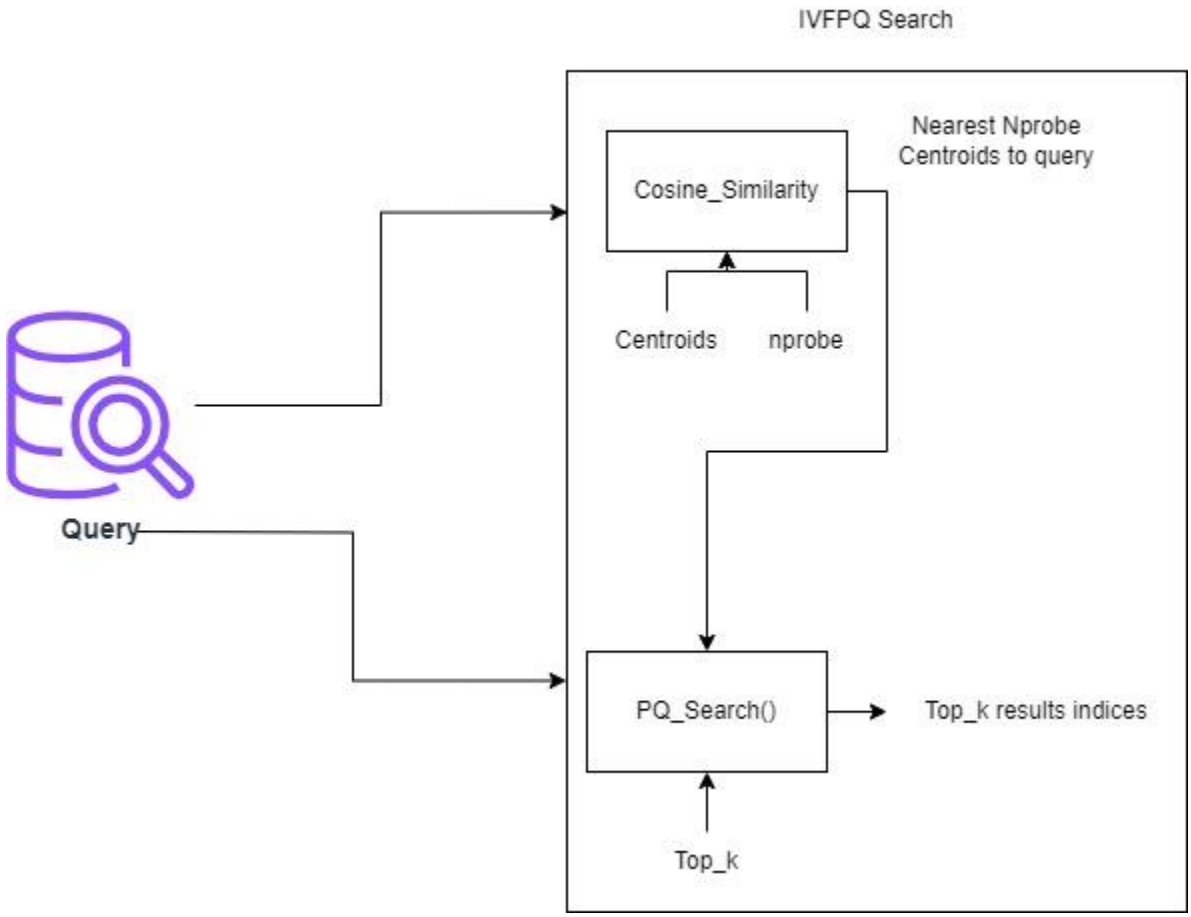
Indexing System:



PQ.add():

1. Designed to encode input data using a set of pre-trained KMeans estimators. It divides the input data into segments and applies Vector Quantization (VQ) using the KMeans estimators to each segment. The resulting output organizes the encoded data with associated identifiers.
2. Return data is added to the pickle file according to the centroid it belongs to(codes_0.pkl, codes_1.pkl, ...).

Search System:



PQ_Search():

1. For each of these closest centroids, it accesses the corresponding pickle file containing stored database codes.
2. It computes the distances between the query points and the stored codes from each pickle file, resulting in a list of distances
3. From each set of distances, the function selects the top k closest distances.
4. It aggregates these top k distances from various centroids into a list.
5. Finally, it sorts and selects the top k closest database codes from this aggregated list.

6. The function returns the IDs associated with the top k closest database codes to the query points.

Reason:

1. High retrieval efficiency: The IVF_PQ index is designed to efficiently retrieve nearest neighbors from large-scale databases.
2. Memory efficiency: It uses PQ quantization technique to reduce the dimensionality of the embeddings after clustering.
3. Query time control: It allows for controlling the query time by adjusting the number of quantization centroids or inverted cells used in the index. By tuning this parameter, we can balance the search accuracy and the query speed according to specific requirements.

Trade-offs:

Parameter 1	Parameter 2	Relation
IVF Cells	Index Building Time	**Inverse**
IVF Cells	Accuracy	**Direct**
IVF Cells	Memory	**Inverse**
IVF nprobes	Accuracy	**Direct**
IVF nprobes	Searching time	**Inverse**
IVF nprobes	Memory	**Inverse**
PQ Sub-quantizers	Index Building Time	**Inverse**
PQ Sub-quantizers	Accuracy	**Direct**
PQ Sub-quantizers	Memory Usage	**Direct**
PQ IDs	Index Building Time	**Inverse**
PQ IDs	Accuracy	**Direct**
PQ IDs	Memory Usage	**Inverse**
Query Time	Index Building Time	**Inverse**
Memory Usage(PQ)	**Recall**	**Inverse**

Indexer at size 1M - 20M: **IVF_HNSW_PQ (possible solution #2)**

Description:

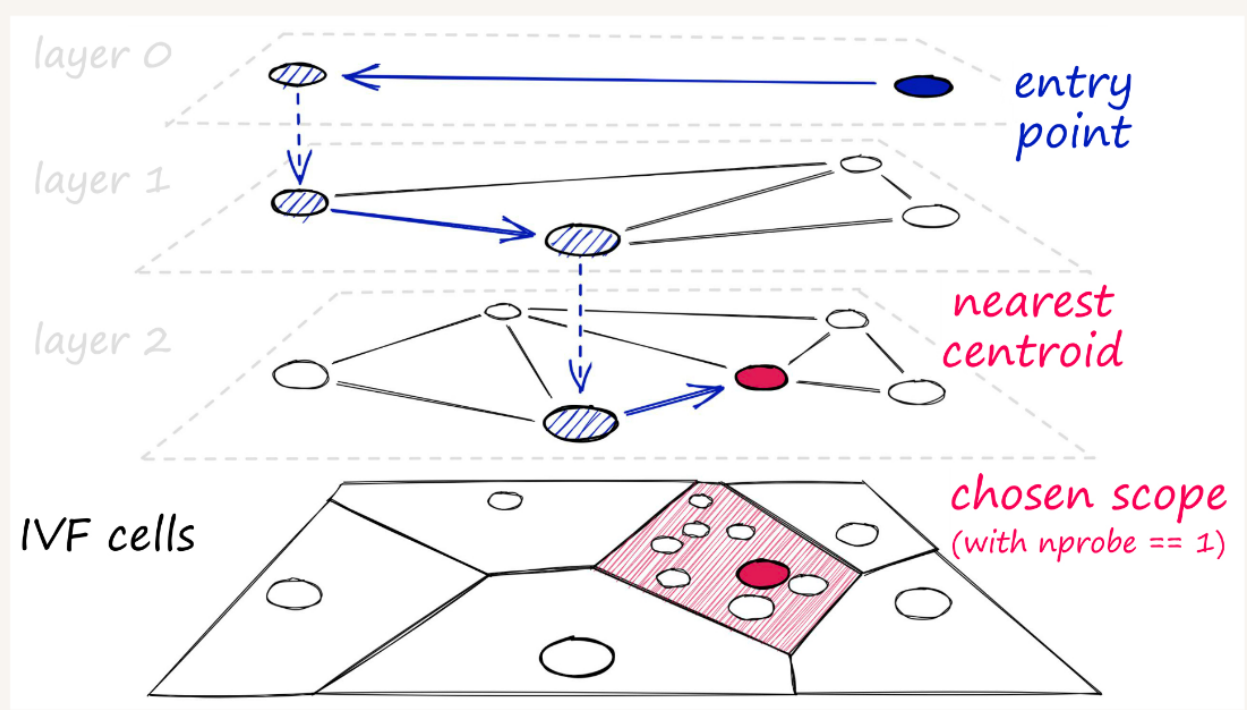
- Training
    - IVF: convert space into subspaces using clustering methods (kmeans) And assign each subspace a representative “centroid”.
    - HNSW: create Hierarchical Graph on IVF centroids.
    - PQ: divide vector to m subvector , for each subvector apply kmeans clustering to get 2**nbits centroids “IDs” that used to quantize vector to m-dimension vector
- Searching: starting with HNSW get nearest nprobe centroids to query vector and apply flat index on quantized vectors using ADC in each subspace to get 100 top_k nearest vectors to query submitted, then apply flat index using original vectors on 100 top_k to get top_k.

Architecture:

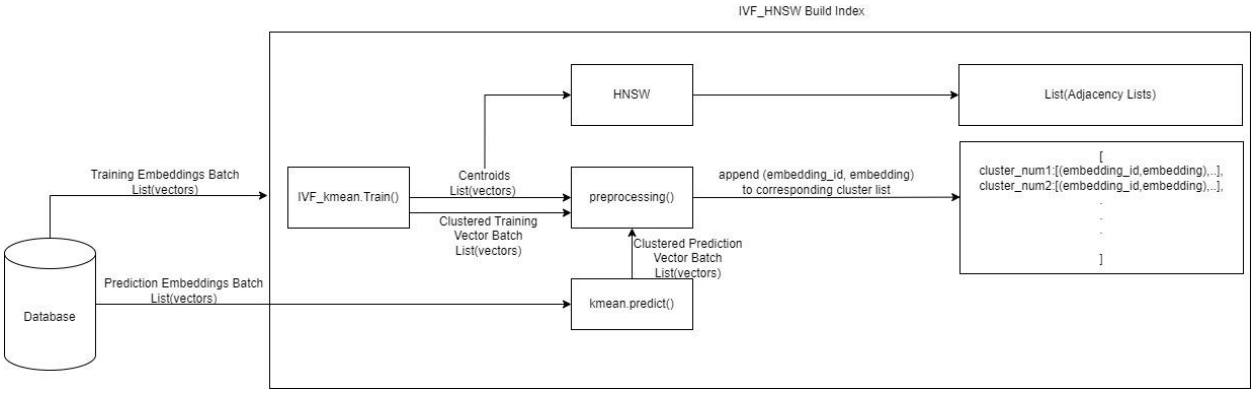
IVF centroids (subspaces)

HNSW Graph searching in centroids to find nearest nprobe centroids of ivf

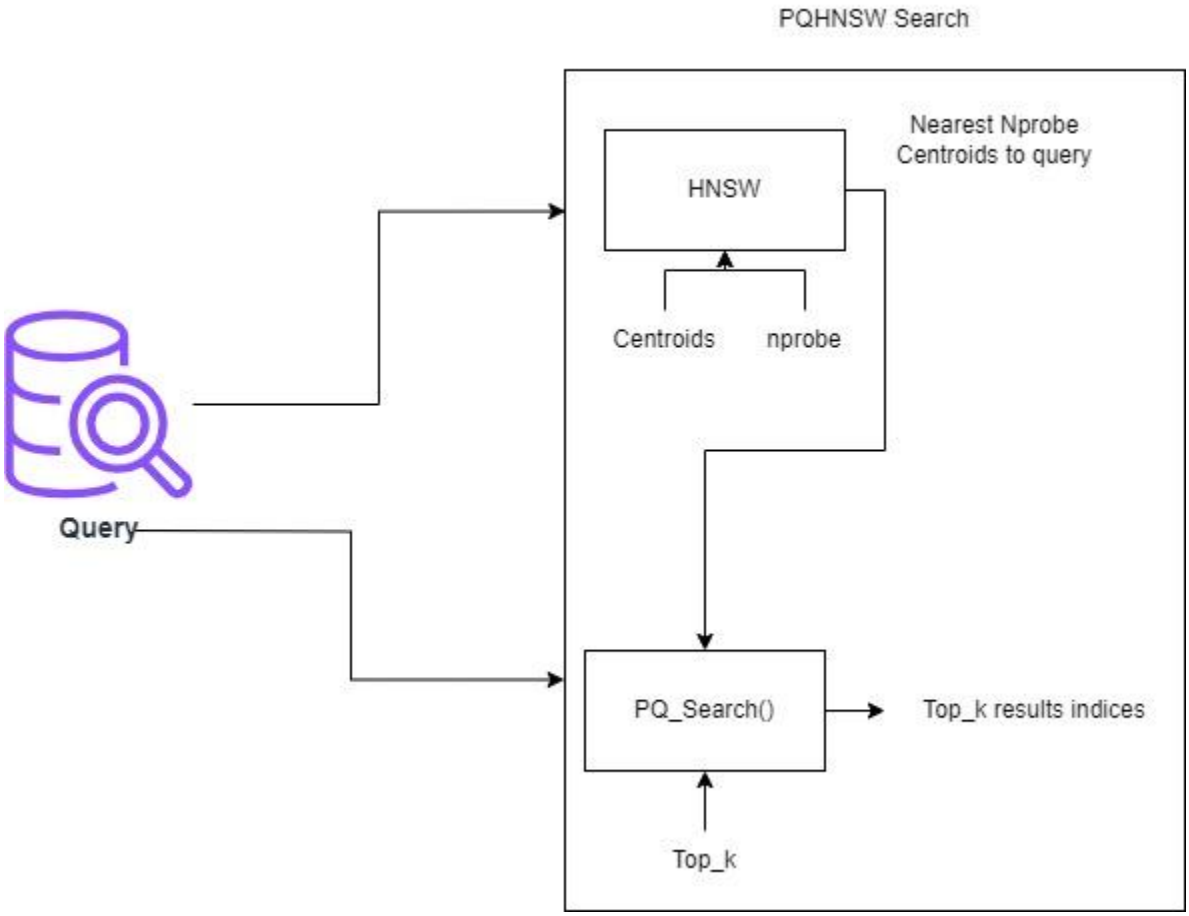
Using centroids searching quantized vectors of each centroid to get nearest top_k vectors to retrieve



Indexing System:



Search System:



Reason:

1. High retrieval efficiency: The IVF_HNSW_PQ index is designed to efficiently retrieve nearest neighbors from large-scale databases.
2. Memory efficiency: It uses PQ quantization technique to reduce the dimensionality of the embeddings after clustering.
3. Query time: It decreases searching time using HNSW in searching large centroids of ivf and achieving high recall.