OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF

DBSE

FAKULTÄT FÜR INFORMATIK

Optimization of the Search Experience in Search Engines with Vector Databases and Transfer Learning

Under the supervision of:

Prof. Dr. rer. nat. habil. Gunter Saake
Dr.-Ing. David Broneske

6th September 2023

Ashish Soni (221453)

Data and Knowledge Engineering (M.Sc.)

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Agenda

- **Introduction**
- **Goal, Motivation and Main Contribution**
- **Background**
  - **Lexical vs Semantic Search**
  - **Transfer Learning and Vector Databases**
  - **Evaluation Measures and Similarity Metrics**
- **Methodology**
  - **Design**
  - **Experimental Setup**
- **Results**
- **Conclusion and Future work**

What is Search?

- **Search**, also known as information retrieval is the process of taking a user query and returning **ranked, relevant results**
- The **first** modern information retrieval system was built in the **1960s[1]** led by **Gerard Salton** with his research group at **Cornell**
- **Google** started as research project in the late **1990s[2]** becoming the world's dominant search engine due to two key innovations - **MapReduce** and **PageRank**
- **Currently,** Platforms such as **Quora**, **Reddit** and **Stack Overflow** have refined search, offering organized and user-specific content in the digital age

[1] Source: https://en.wikipedia.org/wiki/Information_retrieval
[2] Source: https://en.wikipedia.org/wiki/History_of_Google

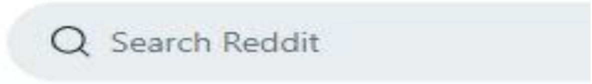
What is the significance of Search?

- According to **Internet Live Stats (May 2023)[1],** Google runs around **8.5 billion searches per day**

- **Quora[2], Reddit[3]** has **300+ million** monthly visitors

- Inaccurate or Irrelevant search results **can lead to misinformation**, **decreased user trust**, **lost productivity** and **bad decision making**

Source: https://www.quora.com



Source: https://www.reddit.com

[1] Source: https://fitsmallbusiness.com/google-search-statistics/#searches-on-google
[2] Source: https://www.demandsage.com/quora-statistics
[3] Source: https://foundationinc.co/lab/reddit-statistics

Goal

Enhancing the relevance and speed of results in search engines through the integration of Vector Databases and Transfer Learning

Motivation

- Evolving landscape of Q&A platforms
- Changing nature of search
- Advancements in neural network approaches

Main Contributions

- Comparison of vector databases - Milvus, Pinecone, Qdrant, Weaviate
- Performance benchmarking: Pinecone vs. PostgreSQL vs. PostgreSQL + pgvector
- Zero-shot evaluation of multilingual models
- Development of a multilingual semantic search prototype using quora dataset

Background: Lexical Search

- **also called Keyword Search[1]**

  Query: Where was the last world cup?

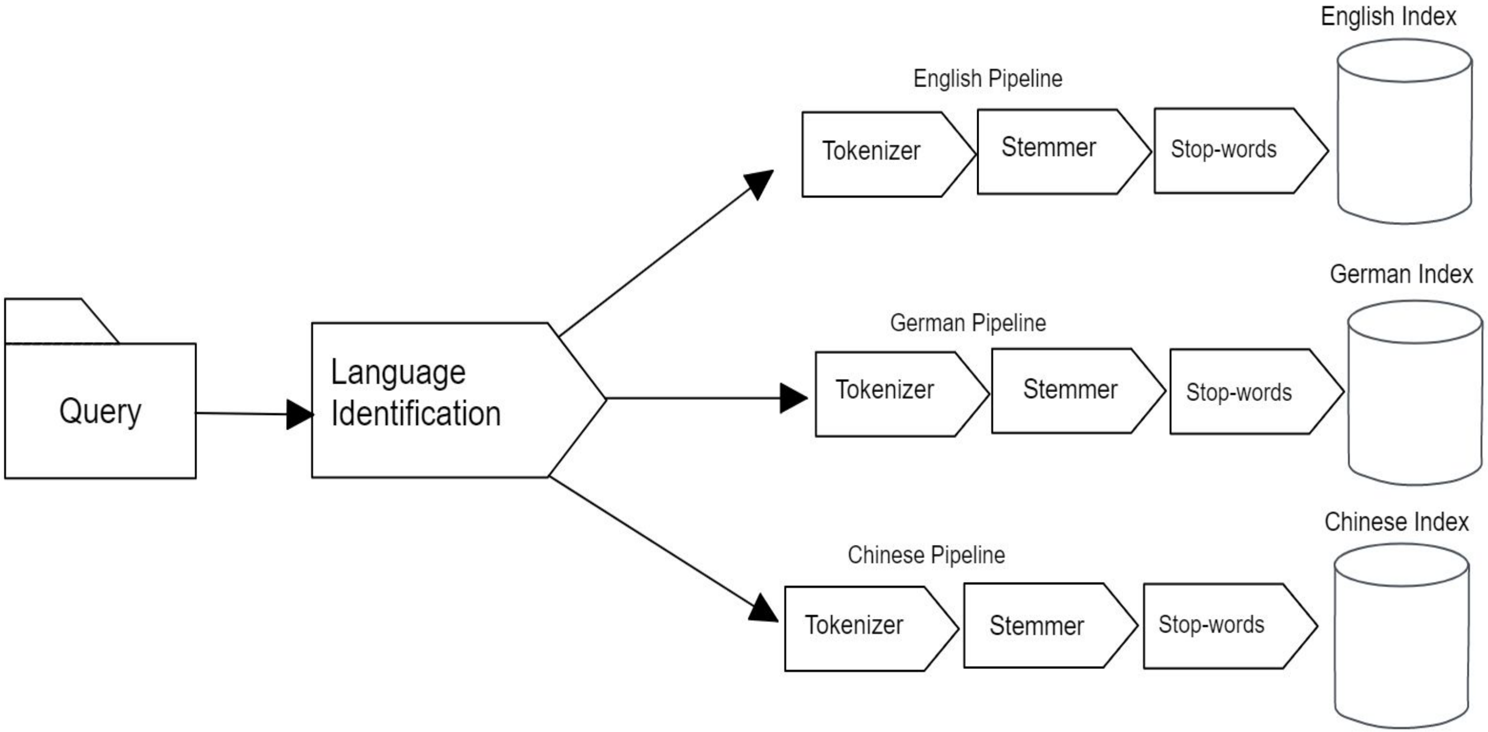
Sentences
The previous world cup was in Qatar
The sky is blue
The bear lives in the woods
An apple is a fruit

- **Lexical Search Problems…**

Sentences
The previous world cup was in Qatar
The cup is where you left it
Where in the world is my last cup of coffee?
An apple is a fruit

[1] Source: https://docs.cohere.com/docs/what-is-semantic-search

Multilingual Lexical Search



Multilingual Lexical Search

Challenges

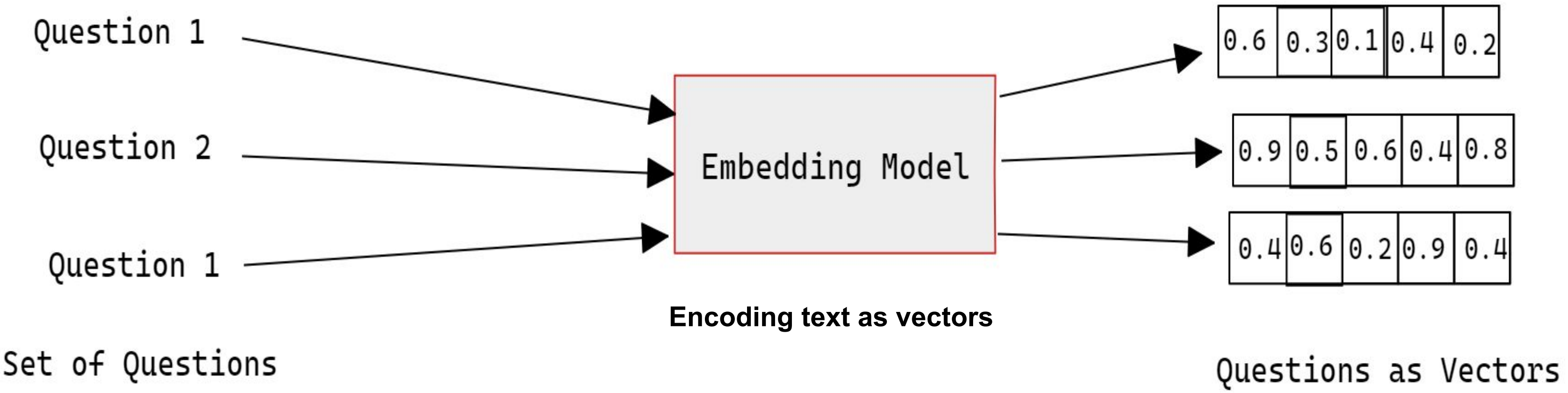
- Multilingual Support
- Storage
- Engineering
- Latency
- Maintenance

Semantic Search

What is an Embedding?

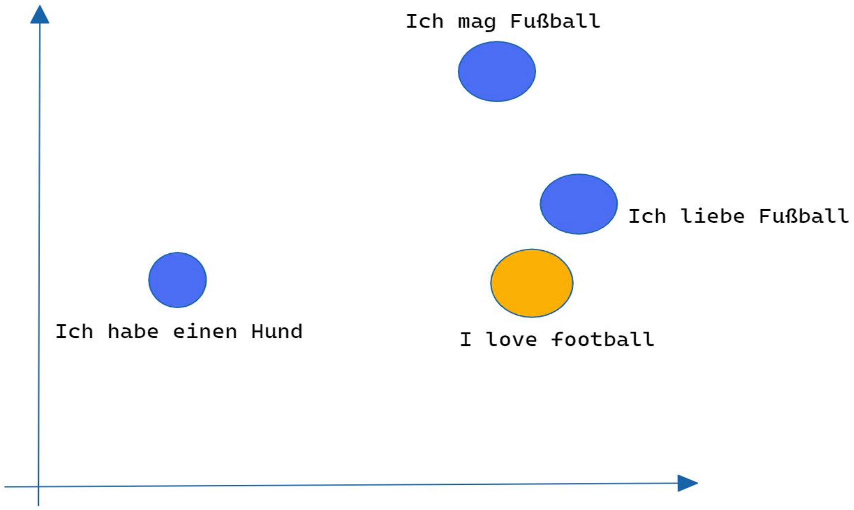
It is a way to assign each piece of text, a vector, which is a list of numbers[1].

- Word Embeddings
- Sentence Embeddings



[1] *Pinecone. Dense vector embeddings for nlp. URL: https://www.pinecone.io/learn/dense-vector-embeddings-nlp*

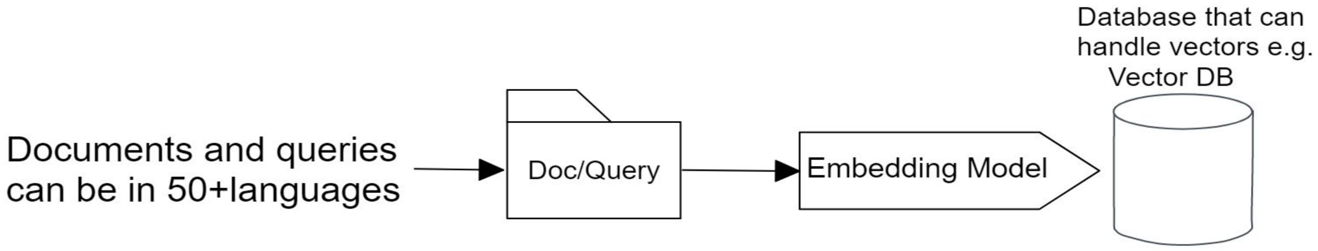
Multilingual Semantic Search



Similar sentences from different languages to similar vector spaces

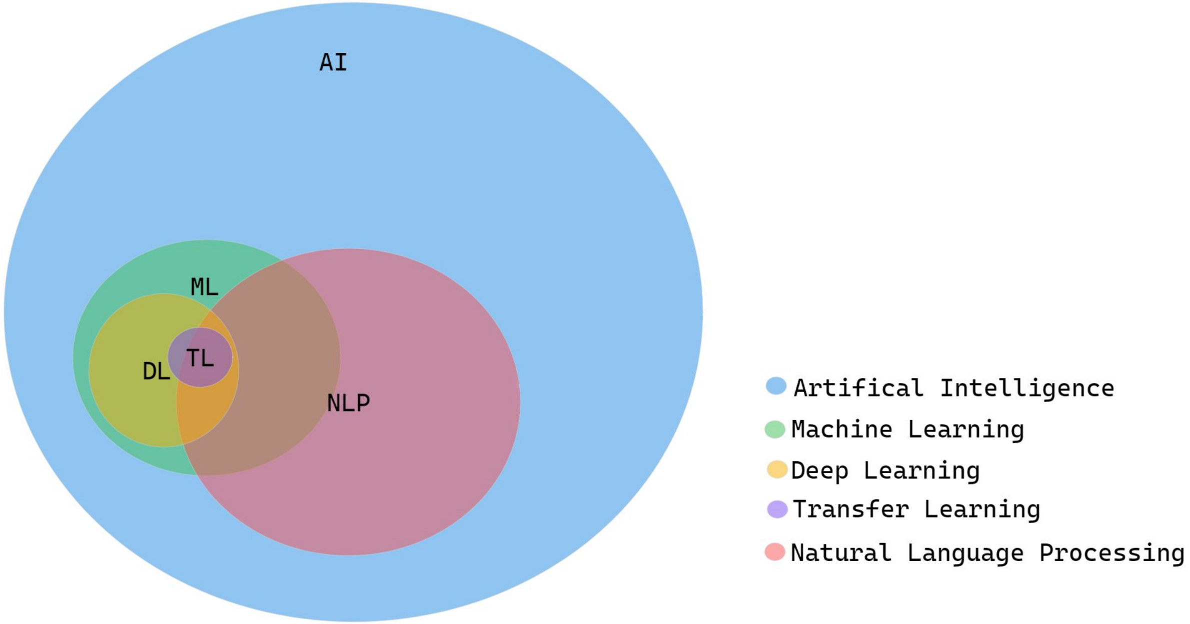
Challenges

- Computational resources
- Maintenance



Multilingual Semantic Search

Transfer Learning



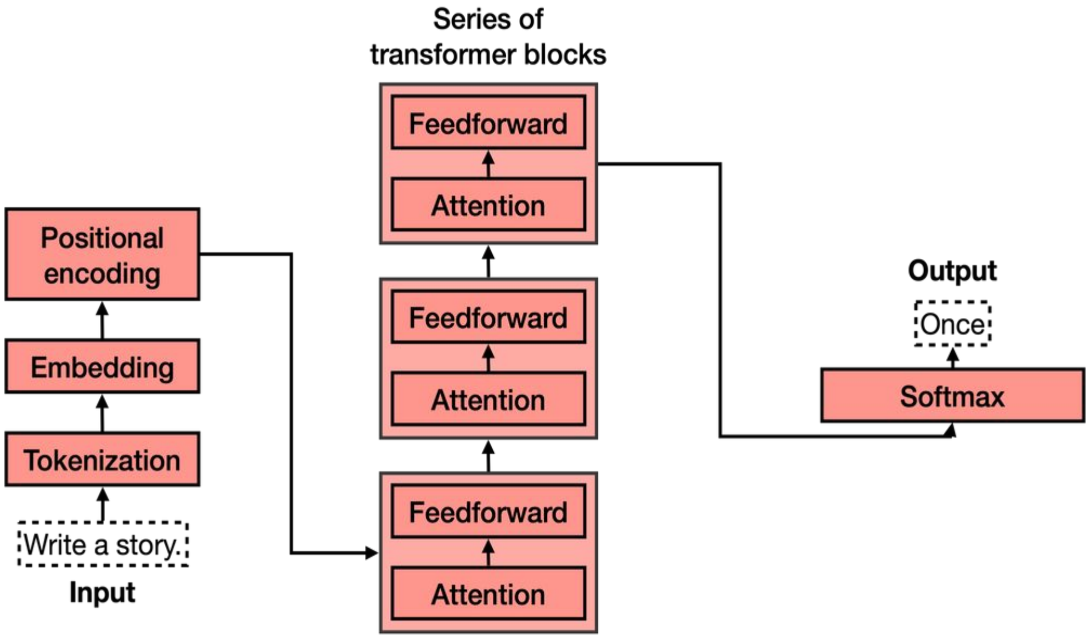
Depiction of Relationships between areas of AI

Advantages

- Leverages knowledge from one task to help learn a new related task
- Reduces the need for extensive data and training time for the new task

[1] *Attention is all you need. 2017. URL:* http://arxiv.org/abs/1706.03762

Transformer Models



The architecture of the transformer model

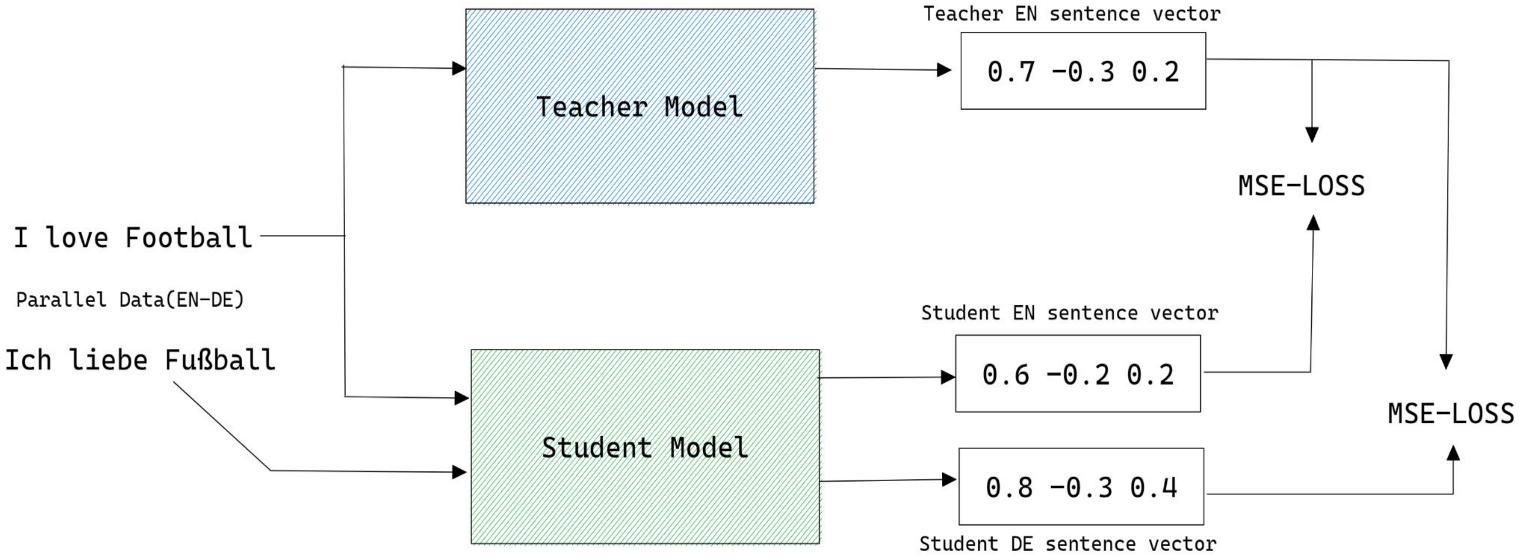
The transformer has the following main parts[1]:

- Tokenization
- Embedding
- Positional Encoding
- Transformer block
- Softmax

[1] Source: https://docs.cohere.com/docs/transformer-models
[2] *Attention is all you need. 2017. URL:* http://arxiv.org/abs/1706.03762

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Multilingual Sentence Transformers



Knowledge Distillation Training Approach

[1] *Sentence-bert: Sentence embeddings using siamese bert-networks. 2019. URL:* https://arxiv.org/abs/1908.10084
[2] *Making monolingual sentence embeddings multilingual using knowledge distillation. 2020. URL:* http://arxiv.org/abs/2004.09813

Vector Databases

- Designed to index and store vector embeddings
- CRUD capabilities along with metadata filtering

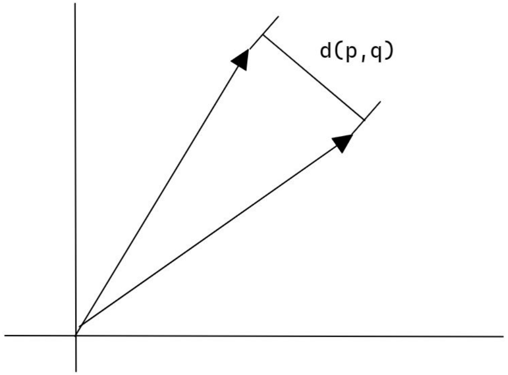
Feature	Qdrant	Milvus	Weaviate	Pinecone
Version	1.1.0	2.2.4	1.18.2	2.2.0
Written in	Rust	Go	Go	Rust
Consistency Model	Eventual	Strong	Eventual	Eventual
Max Vectors Dimension	N/A	32,768	N/A	20,000
Deployment	Managed/ Self Hosted	Self-Hosted	Managed/Se lf Hosted	Managed
Filtering with ANN	N/A	N/A	Pre-filtering	Single-stage filtering

[1] *Manu: a cloud native vector database management system*
[2] *Powering ai with vector databases: A benchmark(part i). https://www.farfetchtechblog.com/en/blog/post/powering-ai-with-vector-databases-a-benchmark-part-i*

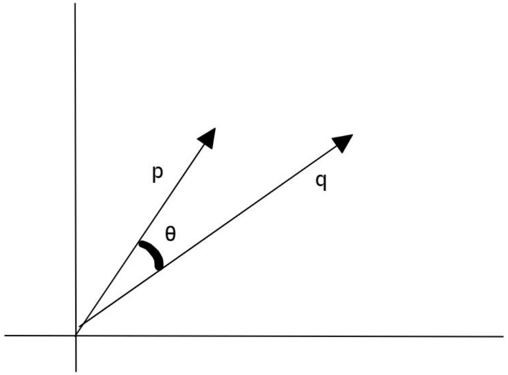
Similarity Metrics

Name	Vector Properties Considered
Euclidean Distance	Magnitudes and Direction
Cosine Similarity	Only Direction
Dot product Similarity	Magnitudes and Direction

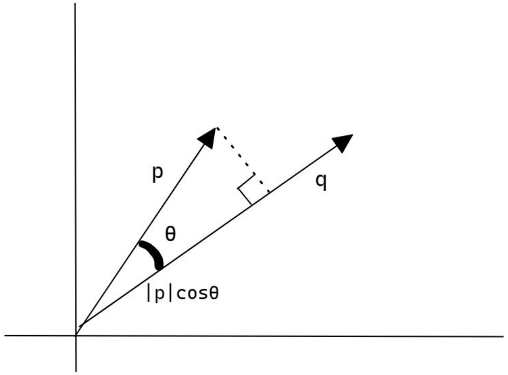
Metrics to compare similarity between vectors



Euclidean Distance



Cosine Similarity

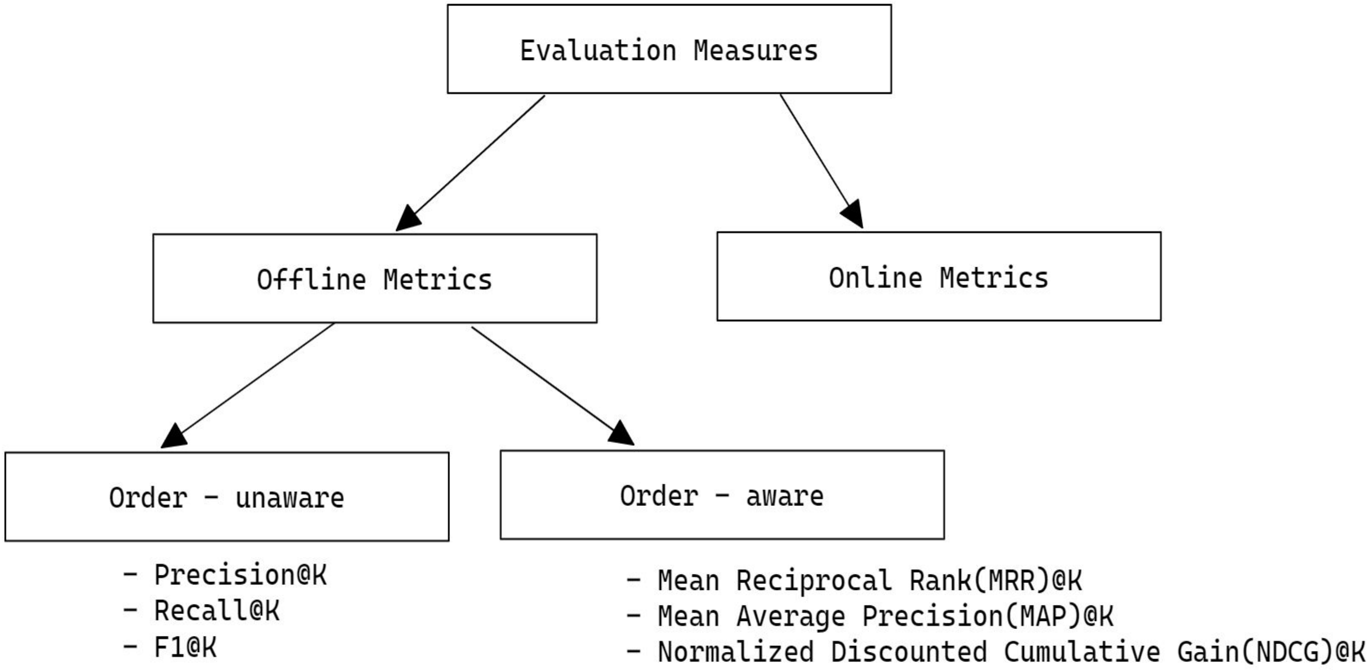


Dot Product

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

Evaluation Measures

Metrics[1]



Metrics to assess the performance of the IR system

[1] *A survey on performance evaluation measures for information retrieval system. 2015*
[2] *BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models*

Design: Research Questions

1. What are the currently available vector databases and their capabilities?

2. How does a vector database perform as compared to a PostgreSQL database(with and without pgvector extension)?

3. What are some of the similarity functions that help to measure the similarity of vectors?

4. What are the currently available multilingual models that can perform a semantic search?

5. How do these multilingual models differ in terms of zero-shot evaluation of their retrieval capabilities and comparison of their inference speed?

Design and Experimental Setup for Database Performance

- Synthetic Data for Assessing Database Performance Benchmarks

Name	Datatype
Id	Integer
Sentence	Object
Embeddings	Object

Name	Id	Sentence	Embeddings
PostgreSQL	Integer()	Text()	ARRAY(Float)
Pinecone	String	String	Vector
PostgreSQL(pgvector)	Integer()	Text()	Vector(size)

- Core Tasks - Implementation of CRUD operations and handling of Batch Insertion

Row Size	Embedding Size
100	384
200	512
300	768
400	1024

K
3
5
10
100
250

Total Number of Rows
10000
30000
50000
70000
100000

Experimental Environment for Database Performance

Machine Configuration

- **Operating System:** Microsoft Windows 10 Home
- **Processor:** Intel(R) Core(TM) i5-7200U CPU @2.50GHz, 2701 Mhz, 2 Core(s), 4 Logical Processor(s)
- **Memory:** 8.0 GB RAM
- **Graphics:** NVIDIAⓇ GeForceⓇ 920MX (2 GB DDR3 dedicated)

Design and Experimental Setup for Multilingual Models

- Conducted zero-shot evaluation using two datasets Quora(English) and Germandpr-beir(German)
  - corpus
  - queries
  - qrels

Model Name	max_seq_length	embedding_dimension
paraphrase-multilingual-MiniLM-L12-v2	128	384
distiluse-base-multilingual-cased-v1	128	512
paraphrase-multilingual-mpnet-base-v2	128	768
quora-distilbert-multilingual	128	768

List of Multilingual Models

Recall@K & MRR@K
1
3
5
10
100

Top K values

- Inference Speed Comparison at K = 10

Experimental Environment for Multilingual Models

Machine Configuration

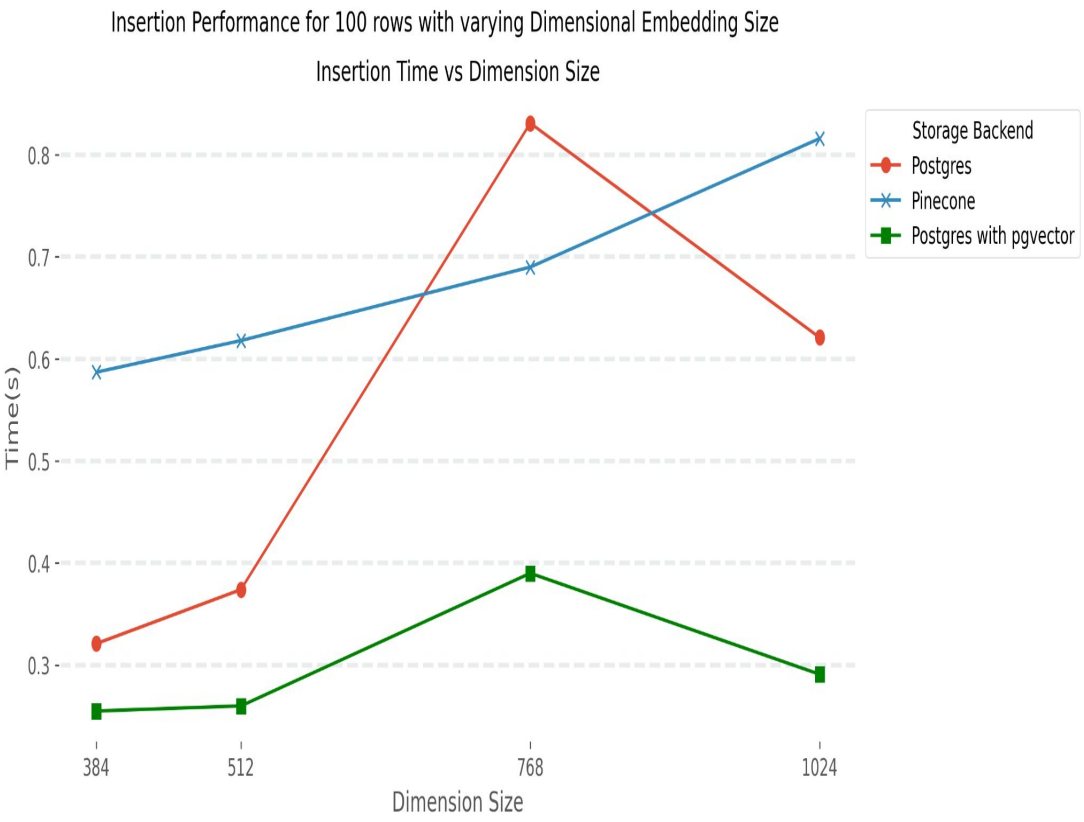
- **Operating System:** Ubuntu 20.04.5 LTS (Focal Fossa)
- **Processor:** Intel(R) Xeon(R) CPU @2.30GHz, 2300 Mhz, 2 Core(s), 4 Logical Processor(s)
- **Memory:** 11.0 GB RAM
- **Graphics:** NVIDIA A100-SXM (40 GB)
  - **Driver Version:** 525.85.12
  - **CUDA Version:** 12.0

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Results: Comparison of Insertion time for rows=100

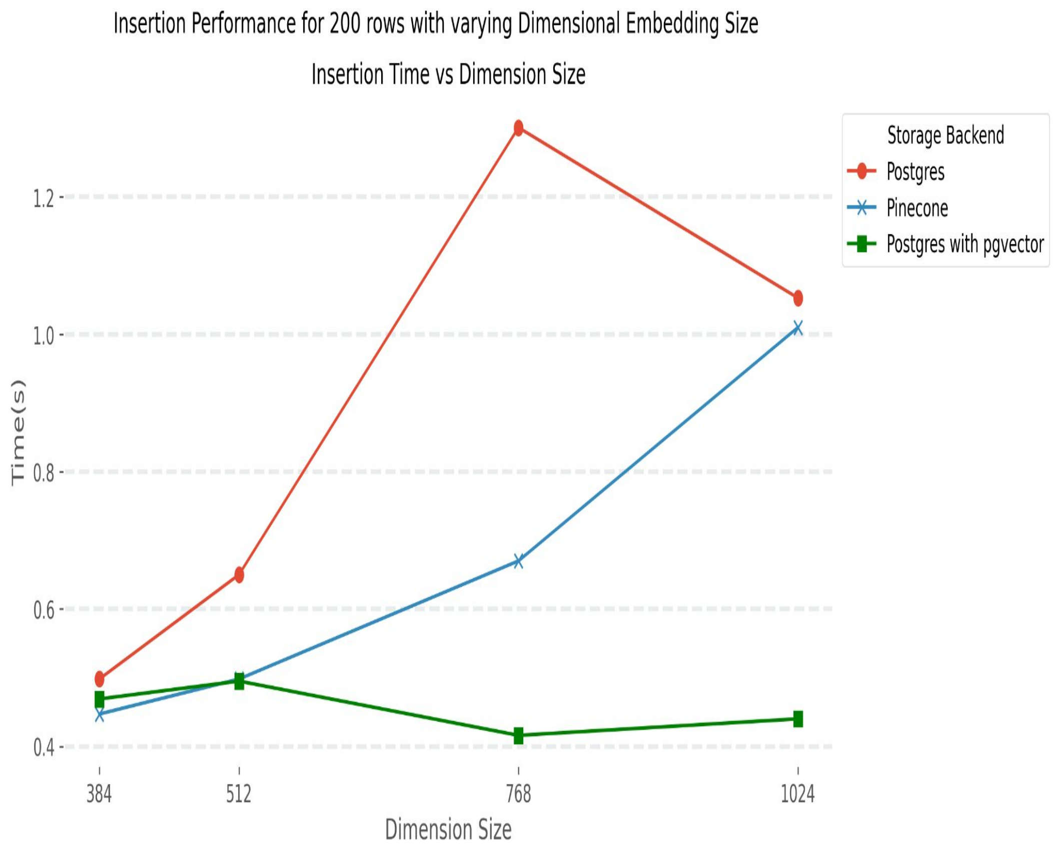
- **Pgvector** is consistently fastest across all embedding sizes
- **PostgreSQL** took longer for an embedding size of **768** than **1024** dimensions
- **Pinecone** outperformed **PostgreSQL** at **768** dimensions
- Both **PostgreSQL** and **Pgvector** had increased insertion times specifically for embedding size of **768**



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

Results: Comparison of Insertion time for rows=200

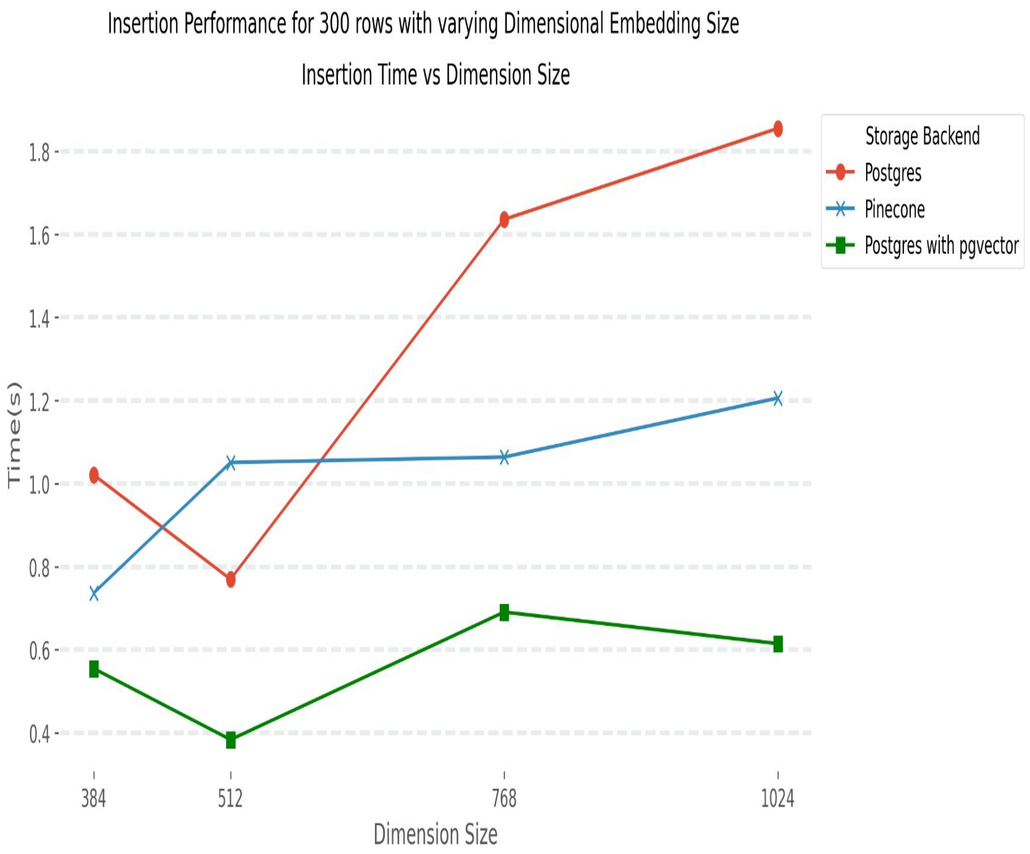
- Performance dip for **PostgreSQL**, especially at **768** dimensions
- **Pinecone** offered best performance for **384** dimensions and on par with **Pgvector** at **512** dimensions
- **Pgvector** outperformed **Pinecone** at **768** and **1024** dimensions
- **Pinecone's** time increases with growing embedding size while **Pgvector** excels at larger dimensions



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

Results: Comparison of Insertion time for rows=300

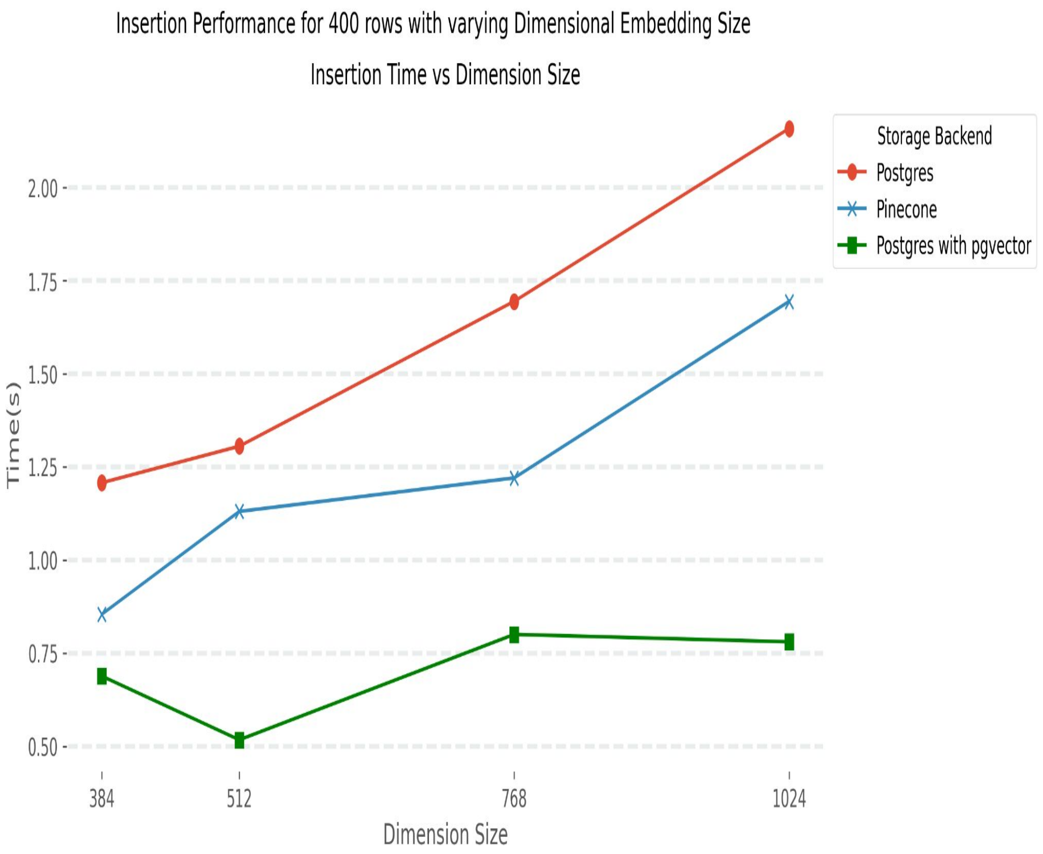
- **Pgvector** outperforms both **PostgreSQL** and **Pinecone** across all embedding sizes
- **Pinecone** excels over **PostgreSQL** in three embedding sizes but lags behind at **512** dimensions
- **PostgreSQL** takes more time for **1024** dimensions compared to **768**



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Results: Comparison of Insertion time for rows=400

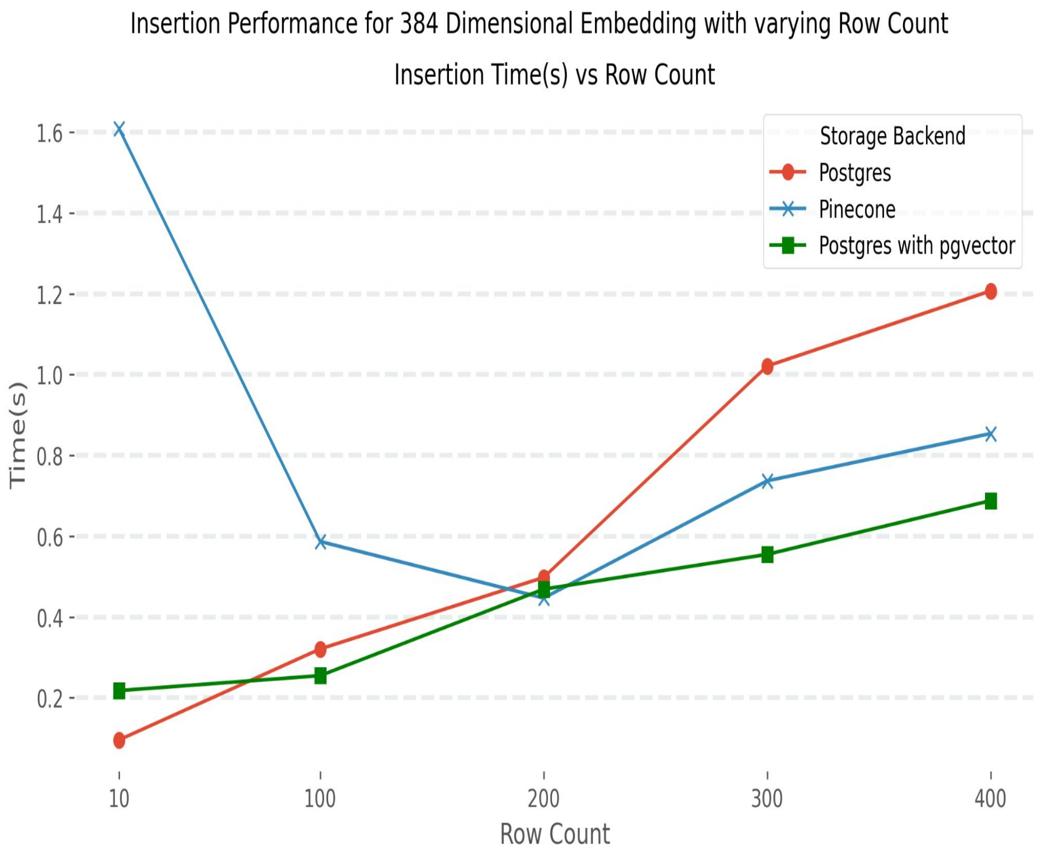
- **Pinecone** consistently outperforms **PostgreSQL** across all embedding sizes
- Both **Pinecone** and **PostgreSQL's** insertion time increase as embedding size grows
- **Pgvector** shows consistent performance utilizing half the time as compared to **PostgreSQL**



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Results: Comparison of Insertion time for Embedding Size=384

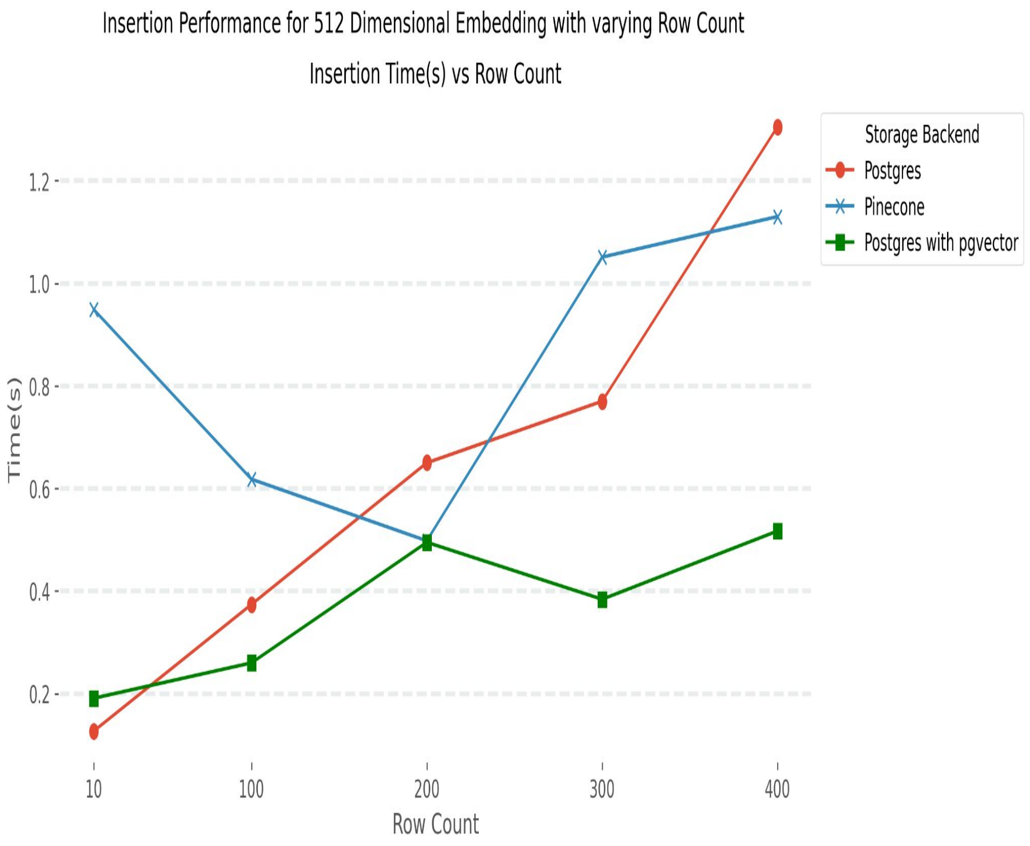
- **PostgreSQL:** time increases with 300 and 400 rows
- **Pinecone** takes more time initially, likely due to network overhead
- **Pgvector:** time for insertion grows as row count rises



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG | INF | FAKULTÄT FÜR INFORMATIK

Results: Comparison of Insertion time for Embedding Size=512

- **PostgreSQL:** constant increase in time with more rows
- **Pinecone** takes more time initially, but outperforms **PostgreSQL** by **0.2** seconds at **400** rows
- **Pgvector** dominates in performance across all row sizes except at **200** rows where it is matched by **Pinecone**

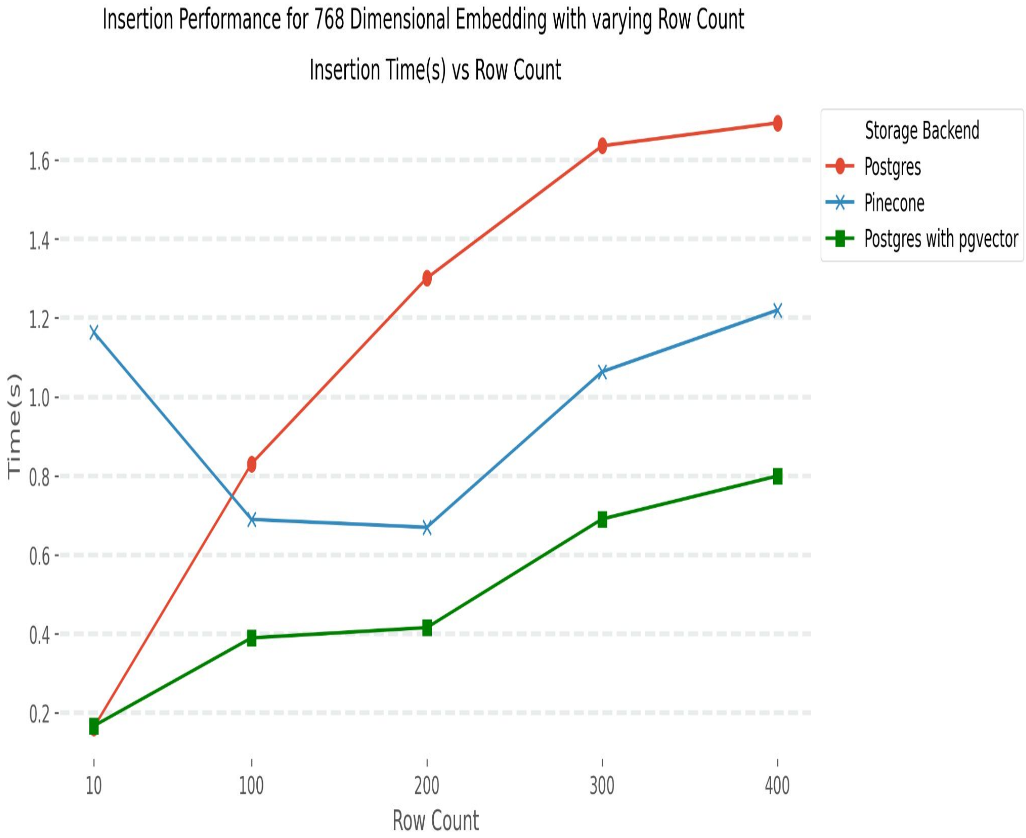


Results: Comparison of Insertion time for Embedding Size=768

- **PostgreSQL:** consistent degradation in performance

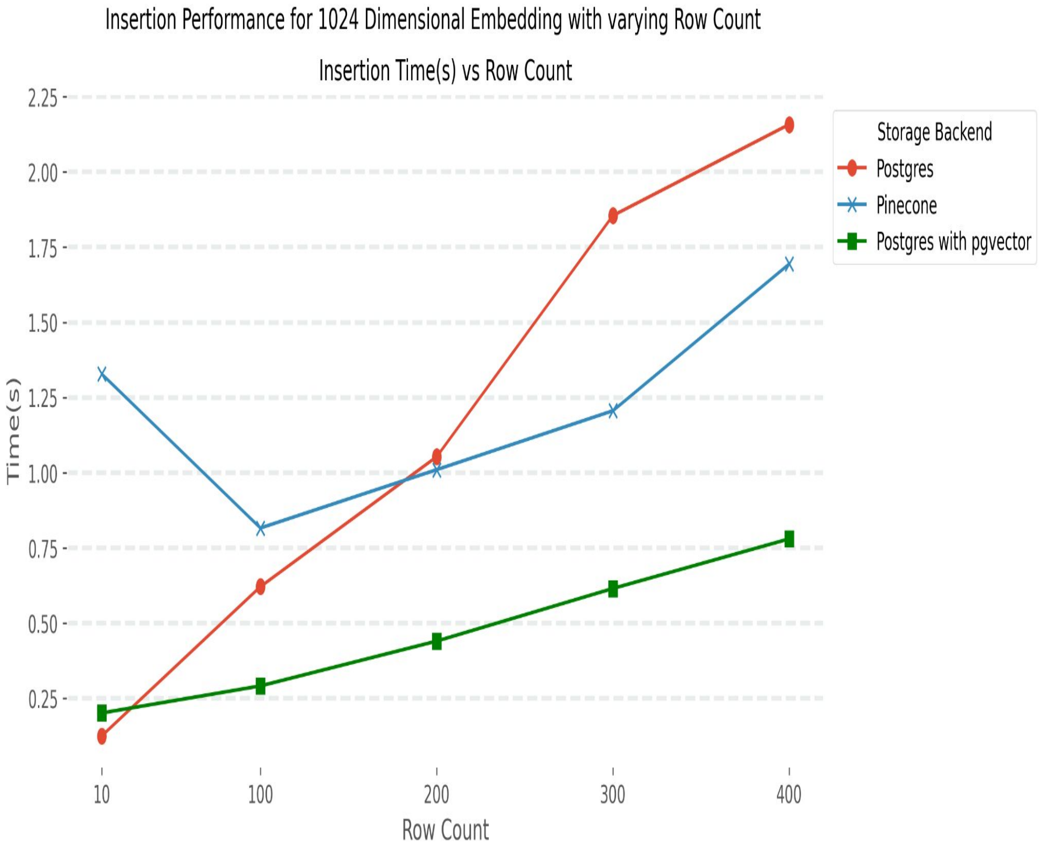
- **Pinecone** outperforms **PostgreSQL** across three row counts

- **Pgvector** leads in performance across all row sizes but also shows a rise in time



Results: Comparison of Insertion time for Embedding Size=1024

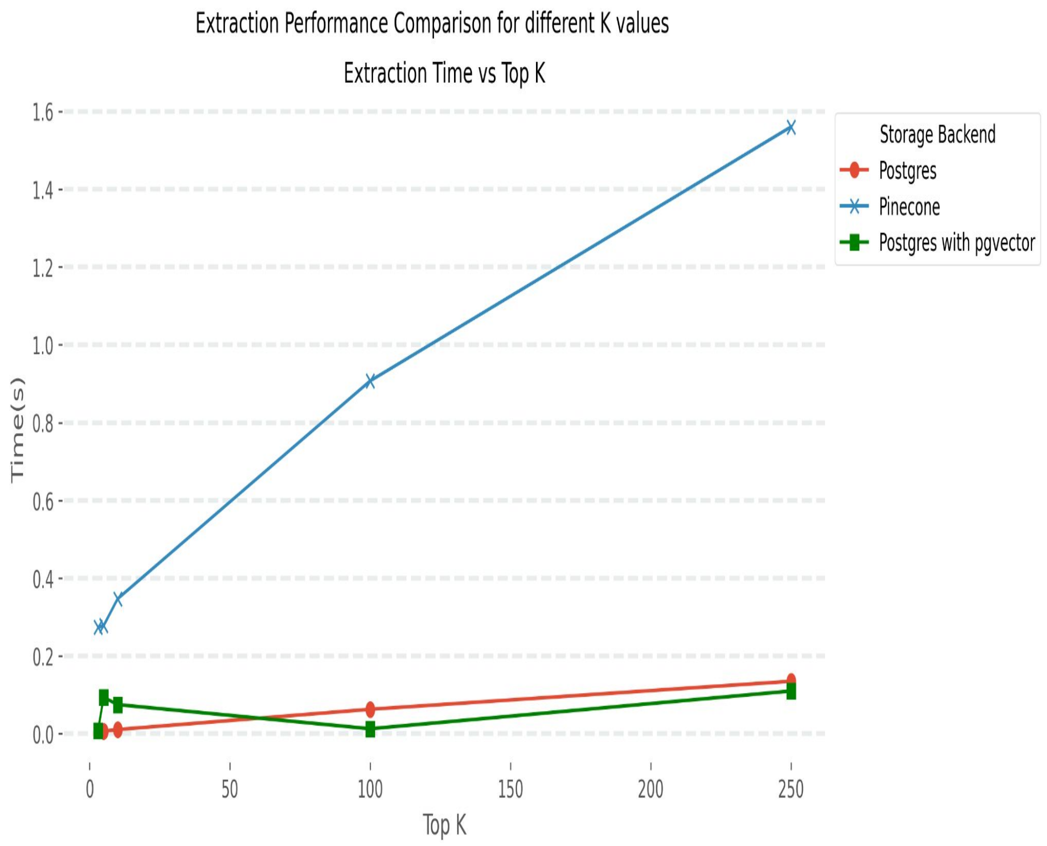
- **PostgreSQL:** leads for row count of **10**
- **Pinecone** suffers with network overhead, also takes longer with growing row count
- **Pgvector** maintains all insertion times under 1 second
- All storage backends show increased time for insertion



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Results: Time taken for Retrieval for different values of K

- **PostgreSQL** lacks vector operations; used standard **SELECT**
- **PostgreSQL** leads in performance of three smallest **K** values
- **Pinecone's** time increases with increasing **K**
- **Pgvector** excels at higher-K values



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG | INF | FAKULTÄT FÜR INFORMATIK

Results: Performance Evaluation for Update Task

- **Updating record by ID** was performed **20** times and the average time was recorded
- **PostgreSQL** stands out with the fastest update times
- **Pgvector** close second, taking only **0.004** seconds more than **PostgreSQL**
- **Pinecone** updated a record in under **0.15** seconds

Database Name	Updation time(s)
PostgreSQL	0.026
Pinecone	0.142
PostgreSQL with pgvector	0.030

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

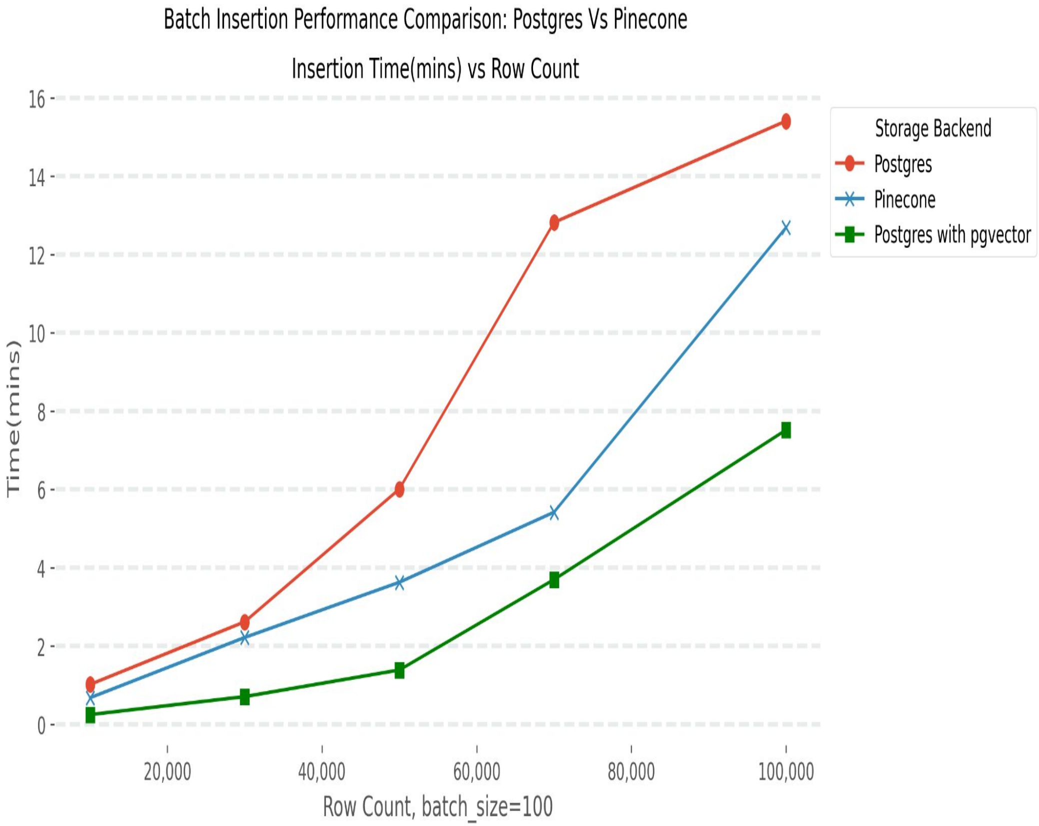
Results: Performance Evaluation for Delete Task

- **Deleting a single record by ID**
- **PostgreSQL and Pgvector showed comparable performance**
- **Pinecone** deleted a record in under **0.14** seconds

Database Name	Deletion time(s)
PostgreSQL	0.020
Pinecone	0.136
PostgreSQL with pgvector	0.023

Results: Performance Evaluation for Batch Insertion Task

- As rows increase, insertion time grows for all storage backends
- **PostgreSQL:** slowest; nearly **16** minutes for **100,000** rows
- **Pinecone** faster than **PostgreSQL** approx. **13** minutes for **100,000** rows and shows better performance at **50,000** and **70,000** rows compared to **PostgreSQL**
- **Pgvector:** best performance; under **8** minutes for **100,000** rows and consistently takes half the time than **PostgreSQL** across all row counts



Storage Backend Selection

- **Storage Efficiency: Pinecone** uses **4** bytes per dimension while **Pgvector** uses **4** bytes + additional **8** bytes for each vector

- **Pinecone** is purpose built for vector data, ensuring optimal performance, low latency and high relevance for large scale semantic search applications

- **Pinecone** uses a graph based index while **Pgvector** uses IVF

Feature	PostgreSQL	Pinecone	PostgreSQL with pgvector
Data Types	Standard + JSON/blob	Vectors	Standard + Vectors
Geometric Filters	No	Yes	Yes
Query Language	SQL	Model(Query, X) > threshold	SQL
Max Vector Dimensions	-	20,000	16,000
Distance Metric	-	Euclidean, Cosine, Dot Product	Euclidean, Cosine, Dot Product
ANN Based Algorithm	-	Graph Based	Inverted File Index(IVF)
Programming Language	C	Rust	C

[1] *Zilliz. Milvus. https://milvus.io/docs/index.md*

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Results: Evaluation of Performance on BEIR Quora Dataset

1. **Recall@K Evaluation:**
   - **Recall@3:** All models above 80%. Top: **MiniLM-L12-v2 (85.2%)**, Lowest: **quora-distilbert (81.7%)**
   - **Recall@5:** All improved, Top: **MiniLM-L12-v2 (90.4%)**, Lowest: **quora-distilbert (86.7%)**
   - **Recall@10:** All above 90%. Top: **mpnet-base-v2 (93.3%)**, Lowest: **quora-distilbert (91.1%)**
   - **Recall@100:** Approaching 100%. Top: **mpnet-base-v2 (99.4%)**, Lowest: **quora-distilbert (98.4%)**
2. **MRR@k Evaluation:**
   - **MRR@3:** Top: **MiniLM-L12-v2 (84.5%)**, Lowest: **quora-distilbert (80.8%)**
   - **MRR@5:** Minimal improvement. Top: **MiniLM-L12-v2 (85.3%)**, Lowest: **quora-distilbert (81.7%)**
   - **MRR@10 & MRR@100:** Marginal differences, same order for all models: **MiniLM-L12-v2, mpnet-base-v2, distiluse-cased-v1, and quora-distilbert.**

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Results: Evaluation of Performance on Germandpr-beir Dataset

1. **Recall@K Evaluation:**

   - **Recall@3:** Models above 70%, Top: **mpnet-base-v2(74.6%)**, Lowest: **quora-distilbert (59.5%)**
   - **Recall@5:** Significant improvements, Top: **mpnet-base-v2 & distiluse-cased-v1(both at 82%)**, Lowest: **quora-distilbert (70.3%)**
   - **Recall@10:** All above 80%. Top: **distiluse-cased-v1(88.4%),** Lowest: **quora-distilbert (79.6%)**
   - **Recall@100:** Approaching 97% for top 3 models, Top: **mpnet-base-v2 (97.6%)**, Lowest: **quora-distilbert (93.9%)**

2. **MRR@k Evaluation:**

   - **MRR@3:** Models start low (~60%), Top: **mpnet-base-v2(60.9%)**, Lowest: **quora-distilbert (46.2%)**
   - **MRR@5:** Marginal improvements. Top: **mpnet-base-v2(62.5%)**, Lowest: **quora-distilbert (48.7%)**
   - **MRR@10 & MRR@100:** Minimal improvements, Top performers: **mpnet-base-v2, distiluse-cased-v1, MiniLM-L12-v2 and quora-distilbert**

Results: Comparison of Inference Speed of Multilingual Models

Model Name	Output Dim.	Index Size(MB)	Avg Inference Time(s)
MiniLM-L12-v2	384	153.60	0.017
distiluse-cased-v1	512	204.80	0.009
mpnet-base-v2	768	307.20	0.017
quora-distilbert	768	307.20	0.009

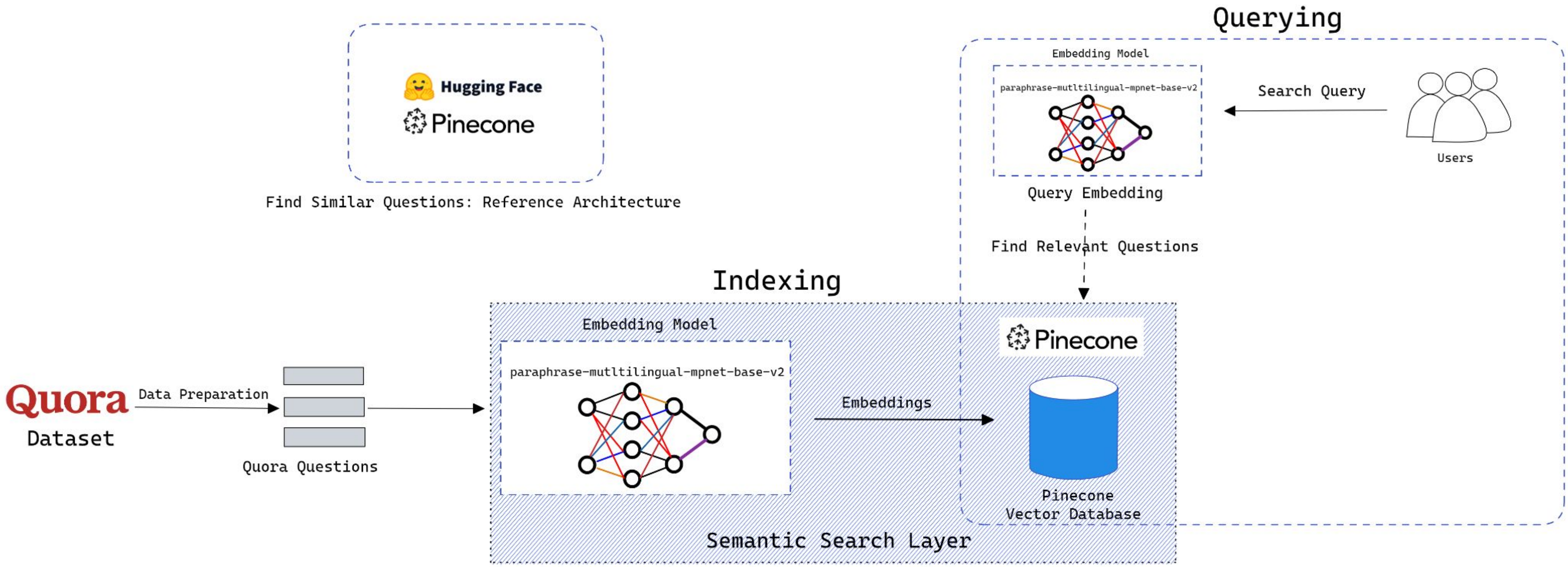
Selection of Best Model

- *Performance on Quora BEIR Dataset:* **MiniLM-L12-v2** dominates both in *Recall@k* and *MRR@k* **mpnet-base-v2** nearly matches **MiniLM-L12-v2**

- *Performance on German-DPR Dataset:* **mpnet-base-v2** tops both metrics. **distiluse-cased-v1** offers comparable performance

- **mpnet-base-v2** chosen for its consistent high performance across experiments, ensuring fast and relevant results to users

Prototype: Multilingual Semantic Search Application



Multilingual Semantic Search Application Architecture

Demo: https://huggingface.co/spaces/Ashish08/Multilingual-Search-Quora-Similar-Questions

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF
FAKULTÄT FÜR INFORMATIK

Conclusion

- **Storage Backend Evaluation**
  - PostgreSQL **pgvector** extension showed superior performance over Pinecone
  - Overheads such as network and authentication affect Pinecone's speed.
  - Pinecone version 2.0, purpose-built for vector storage, offers high search quality with speed.
- **Multilingual Model Performance**
  - **Paraphrase-multilingual-MiniLM-L12-v2** and **paraphrase-multilingual-mpnet-base-v2** were s in the BEIR-Quora dataset.
  - **Paraphrase-multilingual-mpnet-base-v2** excelled in the Germandpr-beir dataset.
  - Inference Speed: **distiluse-base-multilingual-cased-v1** and **quora-distilbert-multilingual**, with 9 milliseconds for fetching top-k results.

Future work

- Investigate Performance of other Vector Databases
- Comparison of Performance after Fine tuning models
- Explore the impact of varying vector indexes on search quality and speed

Thank You!