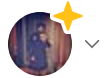




Search Medium

Write



# Faiss: Efficient Similarity Search and Clustering of Dense Vectors



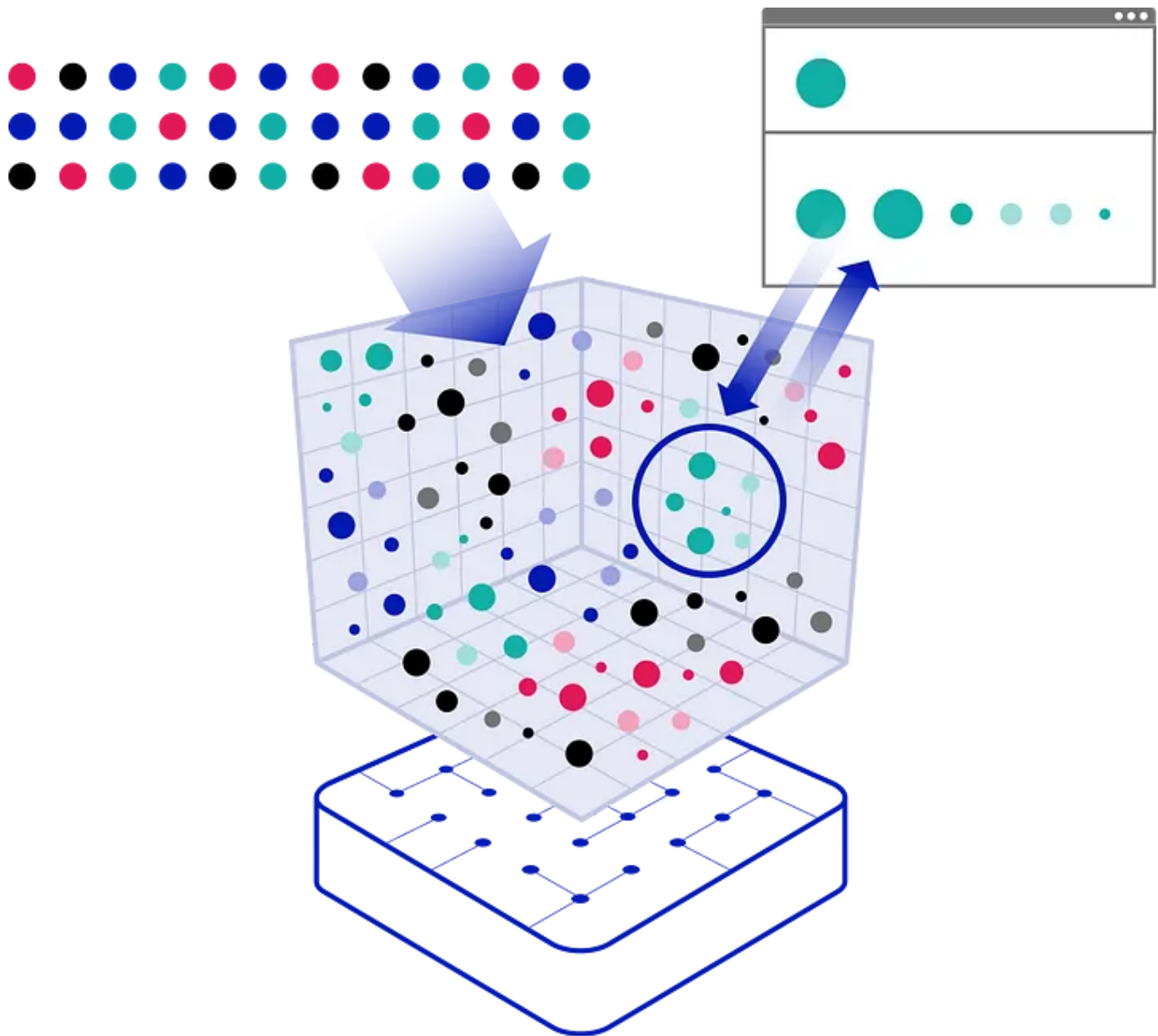
Pankaj Pandey · Following

3 min read · Jun 14



66





Faiss: Efficient Similarity Search and Clustering of Dense Vectors

Faiss is a powerful library designed for efficient similarity search and clustering of dense vectors. It offers various algorithms for searching in sets of vectors, even when the data size exceeds the available RAM. Developed primarily at Meta's Fundamental AI Research group, Faiss provides complete wrappers for Python/numpy and supports GPU implementations for faster performance. This blog post explores the key features of Faiss and demonstrates its usage with code examples.

## Key Points:

1. Faiss is a library for efficient similarity search and clustering of dense vectors.
2. It supports various algorithms for searching in sets of vectors.
3. Faiss can handle data sizes that do not fit in RAM.
4. It provides complete Python/numpy wrappers and GPU implementations.
5. The library is written in C++ with a focus on performance and scalability.

## Understanding Faiss Methods for Similarity Search

Faiss assumes that instances are represented as vectors and can be compared using L2 (Euclidean) distances or dot products. Similarity is determined by the vectors with the lowest L2 distance or the highest dot product with a query vector. Faiss also supports cosine similarity for normalized vectors. Some methods in Faiss use compressed representations of vectors, while others employ indexing structures like HNSW and NSG to improve search efficiency.

## Example

### Performing Similarity Search with Faiss

Here's a code example demonstrating how to perform a similarity search using Faiss:

```
import faiss
import numpy as np

# Generate random data
data = np.random.rand(1000, 128).astype('float32')
```

```
# Create an index
index = faiss.IndexFlatL2(128)

# Add data to the index
index.add(data)

# Perform a similarity search
query_vector = np.random.rand(1, 128).astype('float32')
k = 5 # Number of nearest neighbors to retrieve
distances, indices = index.search(query_vector, k)

# Print the results
print("Nearest neighbors:")
for i in range(k):
    print(f"Neighbor {i+1}: Index {indices[0][i]}, Distance {distances[0][i]}")
```

## Installing Faiss

Faiss can be easily installed using precompiled libraries for Anaconda in Python or PIP. The library has minimal dependencies and requires only a BLAS implementation. CUDA can be used for optional GPU support. Installation instructions and details can be found in the project's INSTALL.md file.

## Understanding How Faiss Works

Faiss revolves around index types that store sets of vectors and provide search functions based on L2 and/or dot product vector comparison. Different index types offer trade-offs in terms of search time, search quality, memory usage per index vector, training time and the need for external data for unsupervised training. Faiss's GPU implementation is known for its fast exact and approximate nearest neighbor search, Lloyd's k-means and small k-selection algorithms.

## Exploring the Full Documentation of Faiss

For comprehensive documentation on Faiss, including tutorials, FAQs and troubleshooting tips, refer to the project's wiki page. The doxygen documentation provides per-class information extracted from code comments. If you're interested in reproducing research results using Faiss, the benchmarks README and link\_and\_code README offer valuable insights.

## Conclusion

Faiss is an indispensable library for efficient similarity search and clustering of dense vectors. Its wide range of algorithms, support for large-scale data and integration with Python/numpy make it a go-to choice for various AI and machine learning tasks. Whether you're working with CPU or GPU, Faiss provides optimized solutions for high-performance vector operations. Explore the Faiss documentation, try out the code examples and join the vibrant Faiss community to leverage its full potential.

## Legal Notice

Faiss is MIT-licensed. Please refer to the LICENSE file in the top-level directory for more details. Meta Platforms, Inc. holds the copyright for Faiss. For Terms of Use and Privacy Policy specific to this project, kindly refer to the relevant documentation.

For More information, please visit [GitHub — facebookresearch/faiss: A library for efficient similarity search and clustering of dense vectors.](https://github.com/facebookresearch/faiss)

## More from the list: "NLP"

Curated by Himanshu Birla



Jon Gi... in Towards Data ...

### Characteristics of Word Embeddings

★ · 11 min read · Sep 4, 2021



Jon Gi... in Towards Data ...

### The Word2vec Hyperparameters

★ · 6 min read · Sep 3, 2021



Jon Gi... in

### The Word2vec

★ · 15 min read



[View list](#)



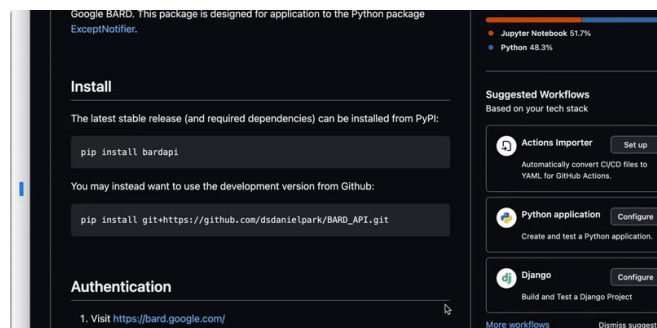
## Written by Pankaj Pandey

Following

66 Followers

Expert in software technologies with proficiency in multiple languages, experienced in Generative AI, NLP, Bigdata, and application development.

## More from Pankaj Pandey





Pankaj Pandey

## Google Bard API: A Powerful Tool for Developers

Google Bard API is a powerful tool that allows developers to access the power of Google's...

2 min read · May 30



58



Pankaj Pandey

## Understanding the ChatGPT API - Key Information and Frequently...

The ChatGPT API has revolutionized the way developers create applications. This blog po...

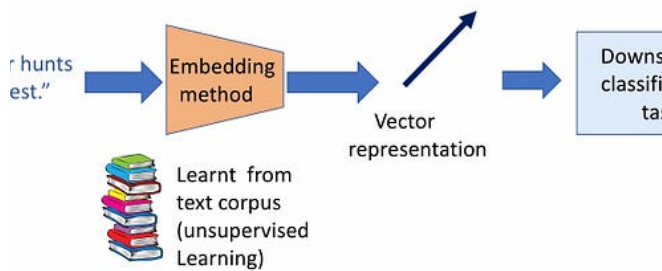
4 min read · Mar 8



76



### Typical pipeline for unsupervised text embedding



Pankaj Pandey

## Natural Language Understanding with Text Embeddings

Introduction

6 min read · Sep 1



53



Pankaj Pandey

## spacy-llm: Integrating LLMs into structured NLP pipelines.

It is crucial for everyone, especially machine learning developers, to be familiar with Spac...

2 min read · May 21



86



3



See all from Pankaj Pandey



## Recommended from Medium



Sirsh Amarteifio

### Cluster chatter: HDBSCAN + LLM

HDBSCAN is a density based (hierarchical) clustering algorithm. Clustering algorithms...

5 min read · Jun 13



3



Alyx

### Semantic Search with FAISS

HuggingFace get\_nearest\_example and Cosine Similarity Search

9 min read · Jul 15



71

1



## Lists



### Staff Picks

465 stories · 317 saves



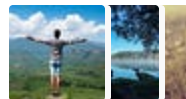
### Stories to Help You Level-Up at Work

19 stories · 235 saves



### Self-Improvement 101

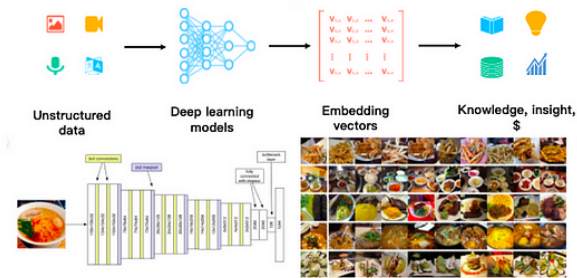
20 stories · 643 saves



### Productivity 101

20 stories · 597 saves





Core principle	Keyword based Search	Semantic Search
Applicable for	Fully /Semi Structured data	Unstructured data
Preferable with	Advance Search Filters	Natural language Interface
Best suited to capture	Explicit context in the query	Implicit context in the query
Frameworks	ElasticSearch, Solr, Pinecone etc	FAISS, Vespa, Solr, Pinecone etc
Hybrid Search		
Core principle	Semantic Search	
Applicable for	Fully / Semi Structured as well as Unstructured data	
Preferable with	Natural language interface alongside Advanced search filters	

Jayita Bhattacharyya in GoPenAI

# Primer on Vector Databases and Retrieval-Augmented Generation...

Vector Databases Generation (RAG)  
Langchain Pinecone HuggingFace Large...

9 min read · Aug 16

228 1

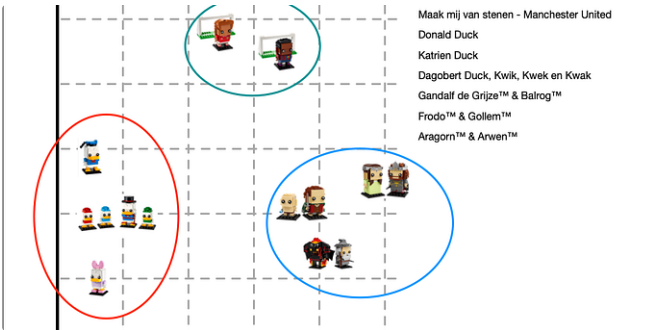
Maithri Vm

# Hybrid Search—Amalgamation of Sparse and Dense vector...

— Uniting meaning of data with metadata to leverage deeper context

13 min read · May 14

16



Jetro Coenradie

# Decoding Similarity Search with FAISS: A Practical Approach

I am preparing for a series of conference talks and workshops on Retrieval Augmented...

6 min read · Aug 29

3



Ariharasudhan

# FAISS : AI SIMILARITY SEARCH

FAISS is an open-source library developed by Facebook AI Research (FAIR) that provides...

6 min read · Jun 27

1

See more recommendations