Vektordatenbank

Modul: Software und Platformarchitektur

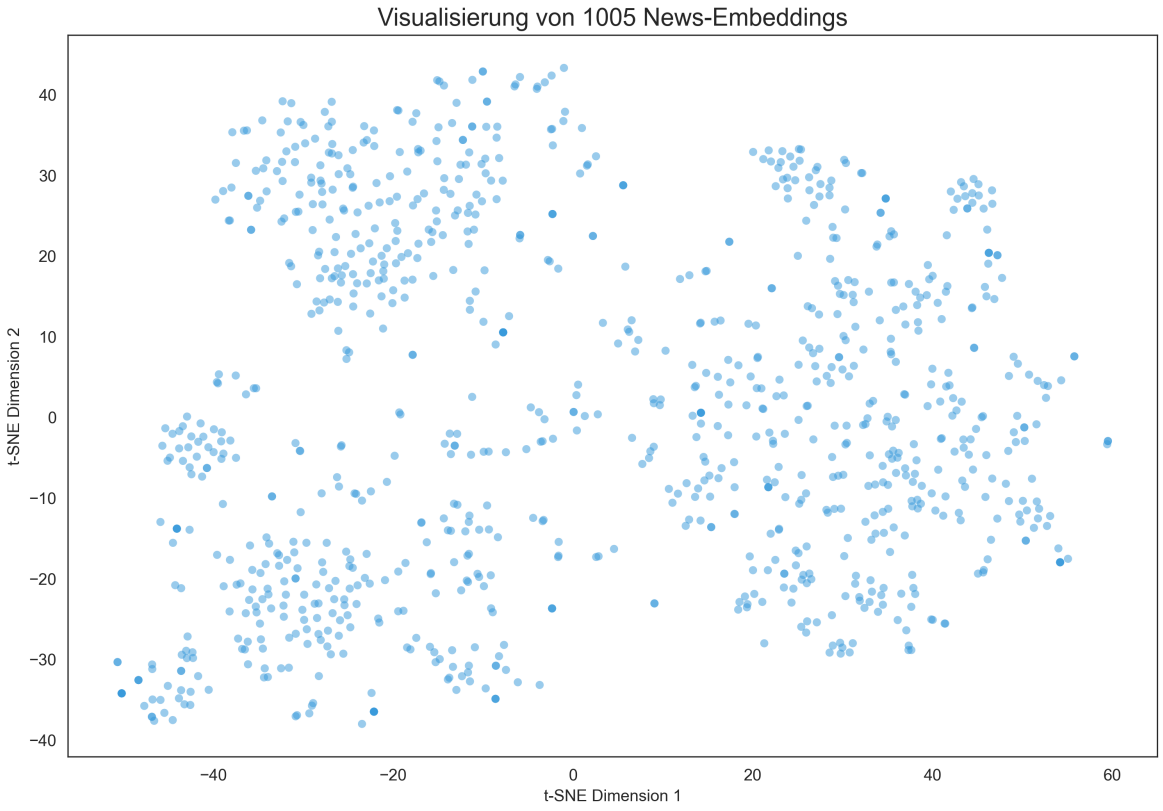
Cédric Caillet

17. Januar 2026

Lehrperson: Patrick Michel

Schule: TEKO Olten

Klasse: O-TIA-TIP-24-S-a



Inhaltsverzeichnis

1 Einleitung

Im Fach Software und Platformarchitektur haben wir den Auftrag erhalten uns zu einem bestimmten Datenbanktyp zu informieren welchen wir selbständig auswählen durften damit wir ein tiefegehendes Verstädnis aufbauen können um so dann der Klasse dies Vorzutragen zu dürfen. Datenbanken bilden das Fundament moderner Anwendungen, indem sie Daten strukturiert und performant speichern, verwalten und abrufbar machen.

2 Was ist eine Vektordatenbank?

Die Vektordatenbank ist ein Datenbanktyp welcher Datenpunkte durch Vektoren mit einer festen Anzahl von Dimensionen dargestellt werden. Vektordatenbanken sind besser in der Lage, unstrukturierte Datensätze zu verarbeiten weil sie eine hochdimensionale Vektoreinbettung verwendet. Die gespeicherten "Vektoren"werden auf der Grundlage von Ähnlichkeiten gruppiert. Dieses Design ermöglicht Abfragen mit geringer Latenz und eignet sich daher ideal für KI-gesteurte Anwendungen.[1]

2.1 Wie funktioniert eine Vektordatenbank?

Wenn wir eine Vektordatenbank mit einer herkömlichen traditionellen Datenbank vergleichen, speichern traditionellen Datenbanken einfache Daten wie Wörter und Zahlen in Form einer Tabelle. Während normale Datenbanken nach exakten Datenübereinstimmungen suchen, versuchen Vektordatenbanken mithilfe spezifischer Ähnlichkeitsmasse nach der besten Übereinstimmung.[2]

Um unsere Datenpunkte in Vektoren umzuwandeln werden *Einbettungsmodelle* geschult. Es gibt verschiedene Einbettungsmodelle je nach Datentyp. Beispielweise gibt es ein Einbettungsmodell für Audio oder Texte oder Videos und Bilder. Vektordatenbanken speichern und erkennen die Ausgabe dieser Einbettungsmodelle. Inerhalb der Datenbank können praktisch jeder Datentyp auf Grundlagen der Bedeutungszusammenhänge oder Merkmale gruppiert oder als Gegensätze indentifiziert werden.[3]

[1] Holdsworth und Kosinski, 2024.
[2] Ali, 2025.
[3] Holdsworth und Kosinski, 2024.

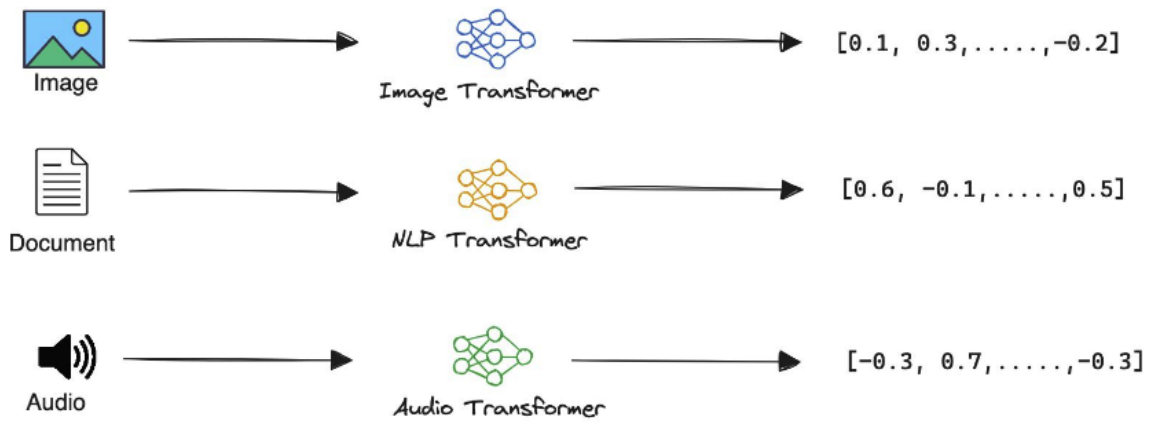


Abbildung 1: Einbettung von Datenpunkte (Quelle: Ogunjobi, 2023)

2.2 Was sind Vektoren?

Ein Vektor ist ein mathematisches Objekt wo in der Mathematik, Physik, Informatik, künstlichen Intelligenz und anderen Anwendungen verwendet werden. Vektoren sind einfach eine numerische Darstellung von Texten, Bildern, Dokumenten oder anderen Formaten in einem hochdimensionalen Raum. Diese Vektoren erhalten Informationen über die Merkmale der Originaldaten, wobei jede Dimension ein bestimmtes Merkmal darstellt.[4]

Hier ein vereinfachtes Beispiel für eine Worteinbettung für 2 Wörter bei dem jedes Wort als zweidimesionaler Vektor dargestellt wird:

- Hund [5,7] - Katze [8,6]

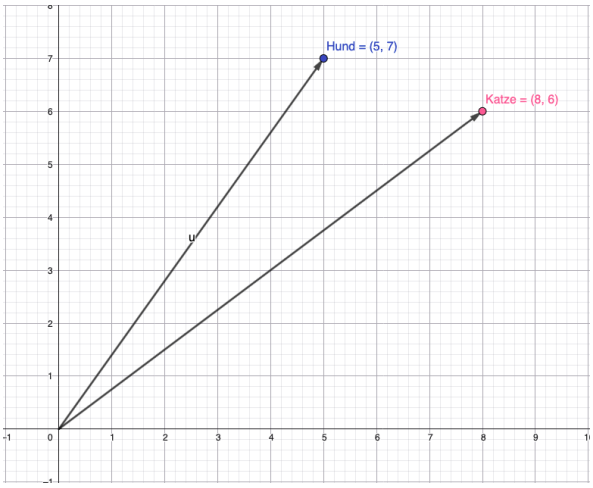


Abbildung 2: Vektor (Quelle: Eigene Darstellung)

[4] Arshad, 2024.

Es wird erwartet, dass Wörter mit ähnlichen Bedeutungen oder Kontexten ähnliche Vektordarstellungen haben. In unserem Beispiel sehen wir, dass die zwei Vekoren für „Hund“ und „Katze“ nahe beieinander liegen, was ihre semantische Beziehung (Bedeutungszusammenhänge) wiederspiegelt.

3 Vor- und Nachteile

Wie bei jeder Technolgie gibt es auch bei Vektordatenbanken nicht nur Vorteile sonder auch Nachteile.

3.1 Vorteile

- **Effiziente Ähnlichkeitssuche:** Vektordatenbanken zeichnen sich durch die Identifizierung ähnlicher Datenpunkte auf der Grundlage ihrer Vektordarstellung aus und ermöglichen Anwendungen, die eher auf Bedeutungszusammenhänge als exakten Keyword Übereinstimmungen basieren.
- **Verwaltung hochdimensionaler Daten:** Vektordatenbanken sind darauf ausgelegt, grosse Mengen hochdimensionalen Daten effizient zu verarbeiten wodurch sie sich für moderne datenintensive Anwendungen eignen.
- **Skalierbarkeit:** Auch bei Millionen Datensätzen liefern Vektordatenbanken performante Ergebnisse und können horizontal skaliert werden um wachsende Datenmengen und Verarbeitungsanforderungen zu bewältigen.
- **Leistungsoptimierung:** Vektorisierte Algorithmen wie *Hierarchical Navigable Small World* HNSW, *Locality-Sensitive Hashing* LSH, *Product Quantization* PQ oder *Approximate Nearest Neighbors Oh Yeah* ANNOY beschleunigen Such- und Abrufvorgänge was zu schnelleren Antwortzeiten führt.

3.2 Nachteile

- **Komplexität beim Verstehen:** Wenn man mit relationalen Datenbanken arbeitet kann der Umsteif auf Vektordatenbanken zunächst komplex wirken. Es braucht etwas Einarbeitung und erfodern je nachdem ein fundiertes Verständnis mehrdimensionaler Datenstrukturen und Algorithmen.
- **Datenintegrität und konsistenz:** Die Gewährleistung einer hohen Datenqualität und konsistenz über Vektoreinbettung hinweg kann kompliziert sein.

- **Ressourcenintensiv:** Die Verarbeitung hochdimensionaler Vektoren kann rechenintensiv sein.
- **Abfragegenauigkeit vs. Leistung:** Es ist notwendig, die Abfragegeschwindigkeit mit der Genauigkeit und Präzision der Vektorsuchergebnisse in Einklang zu bringen, was schwierig sein kann.

[5]

4 Einsatzgebiet / Beispiel für Vektordatenbanken

Vektordatenbanken wurden in den letzten Jahren immer bedeutender, vor allem durch den Aufstieg von *Large Language Models* LLMs wie GPT-4. Wie ich auch schon erwähnt ist der vorteil einer Vektordatenbank, dass sie Daten als hochdimensionale Vektoren (embeddings) abspeichern, was eine blitzschnelle Ähnlichkeitssuche ermöglicht.

4.1 Generative KI und LLMs (RAG)

Dies ist aktuell der bedeutendester Einsatzbereich. Vektordatenbanken dienen als Langzeitgedächnis für KI Modelle.

- **Retrieval-Augmented Generation (RAG):** Anstelle ein Modell ständig neu zu trainieren speichert man aktuelle Firmendaten oder Dokumente in einer Vektordatenbank. Wenn eine Anfrage kommt sucht die Datenbank nach relevantesten Textabschnitte und gibt diese als Kontext an das LLM weiter.

4.2 Semantische Suche (Natural Language Processing)

Herkömmliche Suchen basieren oft auf Schlüsselwörter (Keywords). Vektordatenbanken ermöglichen eine Suche nach der Bedeutung.

- **Kontextuelles Verständnis:** Wenn wir nach König suchen erhalten wir auch Ergebnise zu Monarch oder Herrscher. Aus dem Grund weil der Vektor für König nahe liegend ist wie Monarch und Herrscher.

4.3 Empfehlungssystem

Im lebhaften Einzelhandel verändern Vektordatenbanken die Art und Weise wie leute einkaufen. Es macht anhand des Verhalten des Nutzerprofil vorschläge für Produkte de-

[5] Für das Kapitel 3 wurden diverse Quellen genutzt.

ren Vektor nah um Nutzervektor liegen. Es werden auch ähnliche Artikel vorgeschlagen inerhalb millisekunden.

4.4 Medienanalyse

Vektordatenbanken sind ideal für unstrukturierte Daten wie Bilder und Videos. Von medizinischen Scans bis hin zu Überwachungsaufnahmen ist es echt wichtig, Bilder genau zu vergleichen und verstehen zu können. Es kann auch für Gesichtserkennung genutzt werden indem es Biometrische Merkmale als Vektor speichert und mit Live-Aufnahmen abgleicht

4.5 Anomalieerkennung und Cybersicherheit

In der IT-Sicherheit werden normale Verhaltensmuster (Netzwerkverkehr, Logins) als Vektoren gespeichert um Aussreiser zu erkennen oder Anomalien und somit potenzielle Sicherheitsverletzungen zu vermeiden.[6]

5 Recherche zu mindestens zwei Produkten

Bei Vektordatenbanken gibt es zwei verschiedene „arten“. Es gibt „Native Vektordatenbank“ und allgemeine Datenbanken welche die Vektor suche unterstützen. Wir werden uns das Produkt pgvector anschauen und ChromaDB.

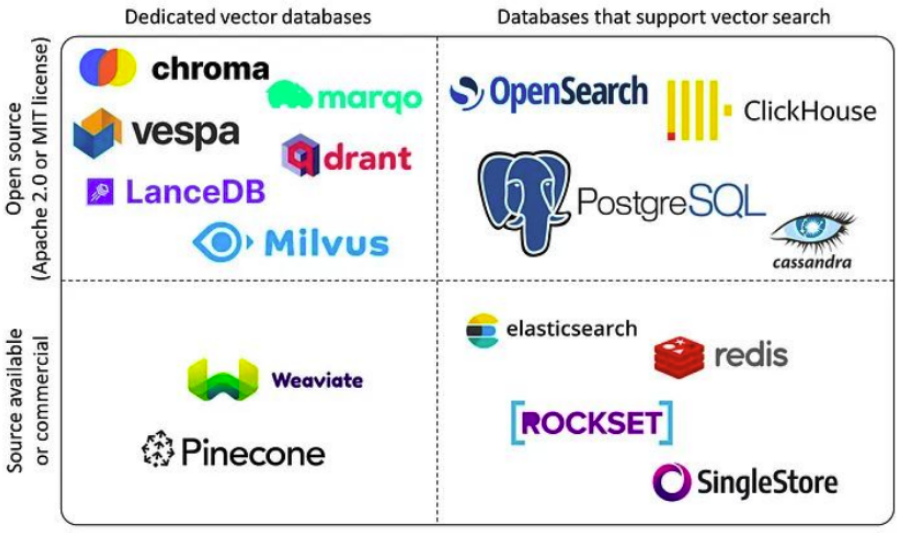


Abbildung 3: Braucht es eine dedizierte Vektordatenbank? (Quelle: Ogunjobi, 2023)

[6] Ali, 2025.

5.1 pgvector

pgvector ist eine Erweiterung für PostgreSQL, die Vektordaten-Typen und Funktionen für die Ähnlichkeitssuche in die weit verbreitete relationale Datenbank einführt. Durch die Integration der Vektorsuche in PostgreSQL bietet es eine nahtlose Lösung für Teams, die schon traditionelle Datenbanken nutzen aber Vektorsuchfunktionen hinzufügen wollen. Die wichtigsten Funktionen von pgvector sind:

- Erweitert ein bekanntes Datenbanksystem um vektorbasierte Funktionen sodass keine seperate Vektordatenbank mehr nötig ist.
- Kompatibel mit Tools und Ökosystemen die schon auf PostgreSQL setzen.
- Unterstützt die Suche nach dem ungefähren nächsten Nachbarn *Approximate Nearest Neighbor* ANN, für effiziente Abfragen von hochdimensionalen Vektoren.
- Macht die Einführung für Leute die sich mit SQL auskennen einfacher und ist damit für Entwickler und Dateningenieure gleichermassen zugänglich.[7]

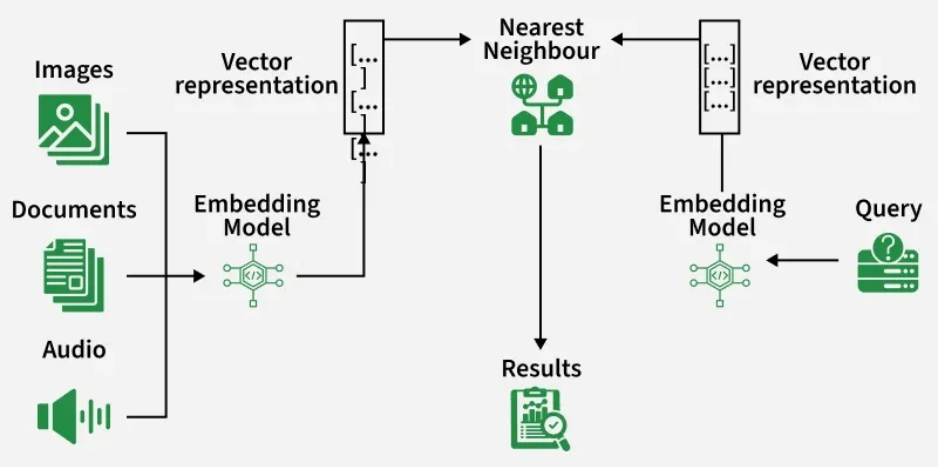


Abbildung 4: flowchart pgvector (Quelle: Azeem, 2025b)

5.2 ChromaDB

ChromaDB ist eine Open Source Native Vektordatenbank, die für effiziente Speicherung, Suche und Verwaltung von Vektoreinbettungen entwickelt wurde. Sie ermöglicht eine schnelle Ähnlichkeitssuche und bietet eine einfache API für Entwickler wodurch sie sich gut für die Erstellung von Bereitstellung KI gesteuerter Anwendungen eignet. Die Key Features von ChromaDB sind:

[7] Ali, 2025.

- **Vektorspeicherung und -abfrage:** Das System sucht mithilfe fortschrittlicher Techniken wie *Hierarchical Navigable Small World* HNSW schnell nach ähnlichen Datenpunkten, da es für die effiziente Verarbeitung hochdimensionaler Daten ausgelegt ist.
- **Benutzerfreundlichkeit:** Es bietet eine einfache Python-basierte API, die es sowohl Anfängern als auch Experten leicht macht, mit Vektordaten zu arbeiten, ohne sich um die Komplexität der Vektorindizierung kümmern zu müssen.
- **Flexible Speicherung:** Das System bietet sowohl temporären Speicherplatz für Tests und Prototypen als auch permanenten Speicherplatz für die Produktion, wodurch unsere Daten sicher und zuverlässig aufbewahrt werden.
- **Integration von Machine Learning:** Es lässt sich problemlos in gängige Einbettungsmodelle von Plattformen wie Hugging Face und OpenAI oder sogar in benutzerdefinierte Modelle integrieren, was eine nahtlose Generierung und Speicherung von Einbettungen ermöglicht.[8]

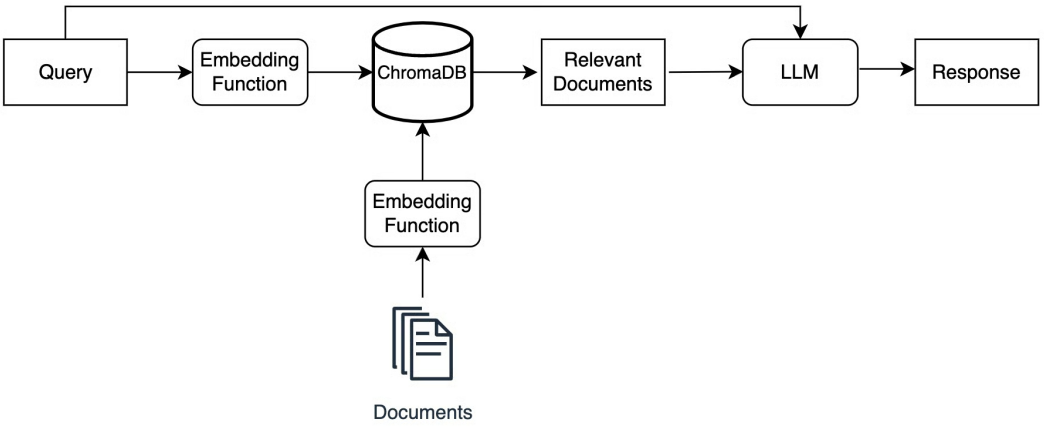


Abbildung 5: Retrieval-augmented generation flowchart ChromaDB (Quelle: Hoffman, 2025)

[8] Azeem, 2025a.

Abbildungsverzeichnis

Quellen- und Literaturverzeichnis

Internetquellen

Ali, M. (2025). *Wie funktioniert eine Vektordatenbank?* Verfügbar 7. Januar 2026 unter https://www.datacamp.com/de/blog/the-top-5-vector-databases

Arshad, S. (2024). *Vectors — a brief origin story.* Verfügbar 5. Januar 2026 unter https://generativeai.pub/everything-you-need-to-know-about-vector-databases-a-deep-dive-4903a40e67a9

Azeem, M. (2025a). *Introduction to ChromaDB.* Verfügbar 12. Januar 2026 unter https://www.geeksforgeeks.org/nlp/introduction-to-chromadb/

Azeem, M. (2025b). *pgvector.* Verfügbar 12. Januar 2026 unter https://www.geeksforgeeks.org/data-science/pgvector/

Hoffman, H. (2025). *Practical Example: Add Context for a Large Language Model (LLM).* Verfügbar 12. Januar 2026 unter https://realpython.com/cdn-cgi/image/width=2000,format=auto/https://files.realpython.com/media/Screenshot_2023-10-28_at_2.05.18_PM.92b839a5972b.png

Holdsworth, J., & Kosinski, M. (2024). *Was ist eine Vektordatenbank?* Verfügbar 5. Januar 2026 unter https://www.ibm.com/de-de/think/topics/vector-database

Ogunjobi, J. (2023). *Introduction to Embeddings.* Verfügbar 5. Januar 2026 unter https://tge-data-web.nyc3.digitaloceanspaces.com/docs/Vector%20Databases%20-%20A%20Technical%20Primer.pdf