# ****Step-by-Step Guide: Storing & Retrieving Word Embeddings in ChromaDB****

## ****Introduction****

This guide will walk you through the process of:

1. Installing and setting up ChromaDB.
2. Training a Word2Vec model.
3. Extracting word embeddings.
4. Storing embeddings in ChromaDB.
5. Querying ChromaDB for similar words.

## ****Step 1: Install Required Libraries****

Ensure you have the required dependencies installed.

pip install chromadb gensim numpy

## ****Step 2: Train a Word2Vec Model****

Train a Word2Vec model on a simple sentence to generate word embeddings.

from gensim.models import Word2Vec

# Define a sample corpus

sentences = [["richard", "spends", "million", "dollars", "on", "a", "private", "jet"]]

# Train Word2Vec model

model = Word2Vec(sentences, vector\_size=5, window=3, min\_count=1, workers=4)

## ****Step 3: Extract Word Embeddings****

Convert words into their corresponding numerical vectors.

words = ["richard", "spends", "million", "dollars", "private", "jet"]

vectors = {word: model.wv[word].tolist() for word in words}

## ****Step 4: Setup ChromaDB****

Initialize ChromaDB and create a collection.

import chromadb

chroma\_client = chromadb.PersistentClient(path="./chroma\_db")

collection = chroma\_client.get\_or\_create\_collection(name="word\_embeddings")

## ****Step 5: Store Embeddings in ChromaDB****

Add word embeddings to the vector database.

for word, vector in vectors.items():

collection.add(ids=[word], embeddings=[vector], metadatas=[{"word": word}])

## ****Step 6: Query ChromaDB****

Find similar words based on vector similarity.

query\_word = "dollars"

query\_vector = model.wv[query\_word].tolist()

results = collection.query(query\_embeddings=[query\_vector], n\_results=3)

## ****Step 7: Display Results****

Print the retrieved similar words along with similarity scores.

for i, (word, score) in enumerate(zip(results["ids"][0], results["distances"][0])):

print(f"Rank {i+1}: Word '{word}' (Similarity Score: {score:.4f})")

## ****Table: Line-by-Line Code Explanation****

| **Code** | **Explanation** |
| --- | --- |
| pip install chromadb gensim numpy | Installs required libraries. |
| from gensim.models import Word2Vec | Imports Word2Vec for training word embeddings. |
| sentences = [["richard", "spends", "million", "dollars", "on", "a", "private", "jet"]] | Defines a single sentence as a training corpus. |
| model = Word2Vec(sentences, vector\_size=5, window=3, min\_count=1, workers=4) | Trains a Word2Vec model with 5-dimensional embeddings. |
| words = ["richard", "spends", "million", "dollars", "private", "jet"] | Specifies words whose embeddings will be stored. |
| vectors = {word: model.wv[word].tolist() for word in words} | Retrieves word embeddings and converts them to lists. |
| import chromadb | Imports ChromaDB. |
| chroma\_client = chromadb.PersistentClient(path="./chroma\_db") | Initializes a persistent ChromaDB client. |
| collection = chroma\_client.get\_or\_create\_collection(name="word\_embeddings") | Creates a new collection named "word\_embeddings". |
| collection.add(ids=[word], embeddings=[vector], metadatas=[{"word": word}]) | Stores embeddings in ChromaDB with metadata. |
| query\_word = "dollars" | Defines the word for which similar words need to be found. |
| query\_vector = model.wv[query\_word].tolist() | Converts the word into a vector representation. |
| results = collection.query(query\_embeddings=[query\_vector], n\_results=3) | Searches for 3 nearest words based on vector similarity. |
| for i, (word, score) in enumerate(zip(results["ids"][0], results["distances"][0])): | Iterates over the retrieved words and scores. |
| print(f"Rank {i+1}: Word '{word}' (Similarity Score: {score:.4f})") | Displays the retrieved words and similarity scores. |