Vector Database Guide

Overview

This project uses **ChromaDB** as a vector database to store, search, and analyze conversation history with semantic search capabilities. Each message is vectorized and stored with rich metadata, enabling advanced retrieval and analytics.

How It Works

- **Every message** (user or AI) is vectorized and stored in ChromaDB as soon as it is processed.
- **Metadata** for each message includes:
  - `session_id`: Groups messages by conversation/session.
  - `speaker`: Who spoke the message.
  - `role`: "user" or "assistant".
  - `is_gemma_mode`: Whether the message was part of an AI conversation.
  - `timestamp`: When the message was recorded.
  - `feedback_helpful`: User feedback (if provided).
- **Text format**:
  `"Speaker A (user): what is your name [GEMMA]"`
  The `[GEMMA]` marker indicates AI-assisted messages.
- **Persistent storage**:
  All data is saved in `program_files/data/vector_db/` and survives restarts.

Setup Instructions

1. Install Dependencies

On any new machine, run:

```
pip install chromadb
```

If you want to use custom embeddings, also install:

```
pip install sentence-transformers
```

2. Directory Structure

Ensure the following exists (ChromaDB will create `vector_db/` automatically):

```
program_files/
    data/
        vector_db/
```

3. Migrating to a New Computer

- Copy the entire `program_files/` directory to your new machine.
- Install dependencies as above.
- If you want to preserve conversation history, copy the `data/vector_db/` folder as well.

Usage

Automatic Operation

- **No manual setup or teardown is needed.**
- The vector database is initialized automatically by the program.
- No need to close or clean up the database—ChromaDB handles this.

Manual Access (for analysis, notebooks, etc.)

```python
from utils.conversation_vector_db import ConversationVectorDB

vector_db = ConversationVectorDB()

# Search for similar conversations
results = vector_db.search_conversations("machine learning", top_k=5)

# Filter by session
session_results = vector_db.search_conversations(
    "", filter_metadata={'session_id': 'session_20250803_203546'}
)

# Get statistics
stats = vector_db.get_conversation_stats()
print(stats)
```

Key Features

- **Semantic search**: Find conversations by meaning, not just keywords.
- **Rich metadata**: Filter/search by session, speaker, role, feedback, etc.
- **Automatic persistence**: Data is always saved, no manual closing required.

- **Session tracking**: Every message is linked to its conversation via `session_id`.
- **Feedback integration**: User feedback is stored and searchable.

Troubleshooting

- **Import errors**: Make sure you are running code from within the `program_files` directory.
- **Permission errors**: Ensure you have write access to `data/`.
- **Dependency issues**: Reinstall `chromadb` and (optionally) `sentence-transformers`.

Summary

- **ChromaDB** is used for all vector storage and search.
- **No manual DB management** is required.
- **Moving to a new computer** is as simple as copying the project folder and running `pip install chromadb`.

For more details, see the code in `utils/conversation_vector_db.py` and the usage examples in your notebook or main program pipeline.