RagDoll
Music Chatbot

Ironhack - AI Engineering Bootcamp
Sept. 2024 - Final Project

Katharina Krux

Introduction

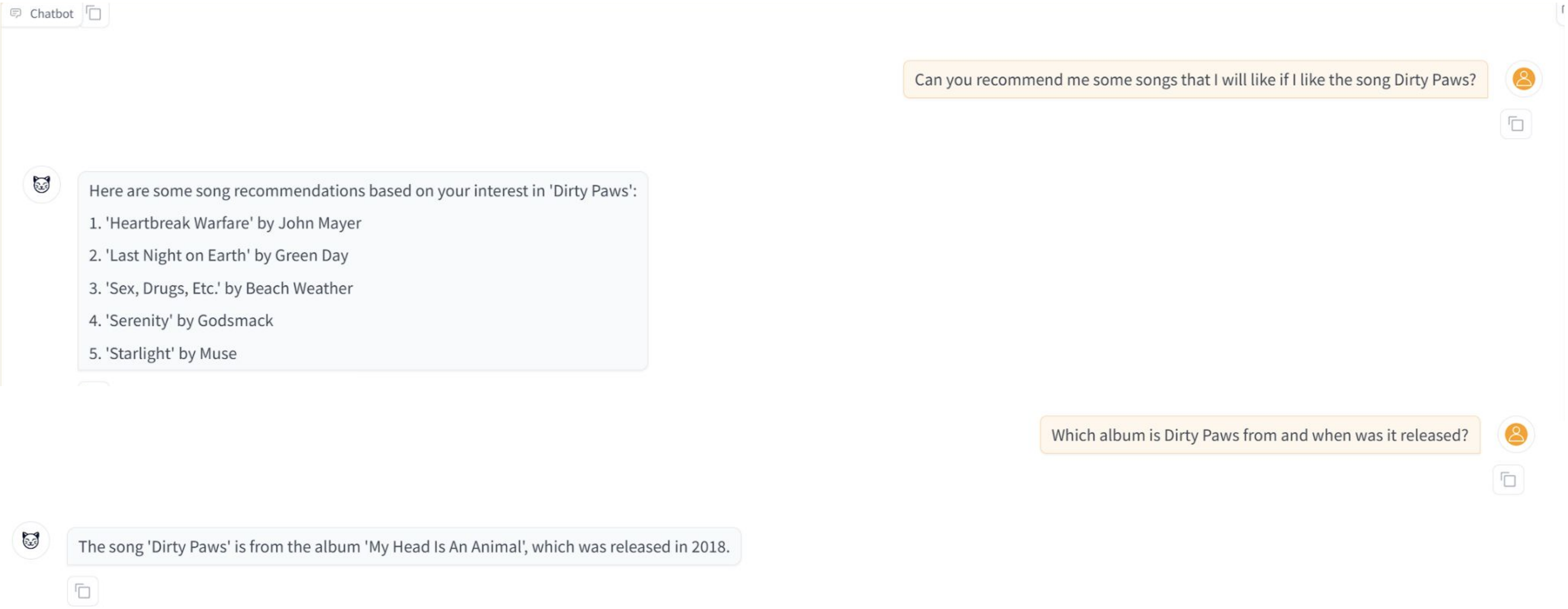
The problem:

Does this feel familiar?

- **Endless** amount of **songs** available - how to **find** the **good ones?**
- Spotify and YouTubeMusic already **do a good job**, but…
- Mainly work **based** on what you **currently listen to**
- If you want to check out a **new artist** or **album**, you **still have to listen to it all!**

The Solution: A Music Chatbot

that can do more than recommend songs...



Can you recommend me some songs that I will like if I like the song Dirty Paws?

Here are some song recommendations based on your interest in 'Dirty Paws':

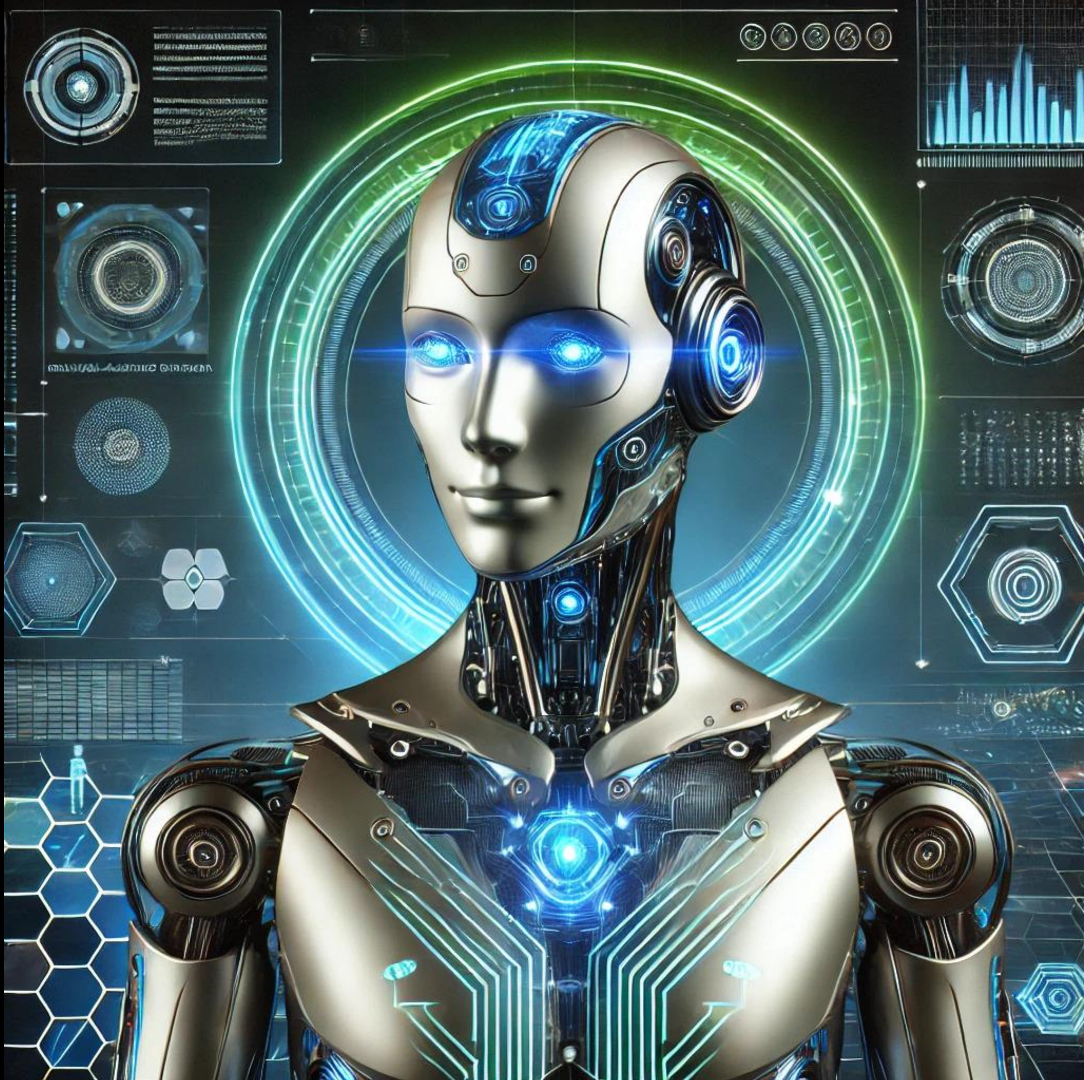
1. 'Heartbreak Warfare' by John Mayer
2. 'Last Night on Earth' by Green Day
3. 'Sex, Drugs, Etc.' by Beach Weather
4. 'Serenity' by Godsmack
5. 'Starlight' by Muse

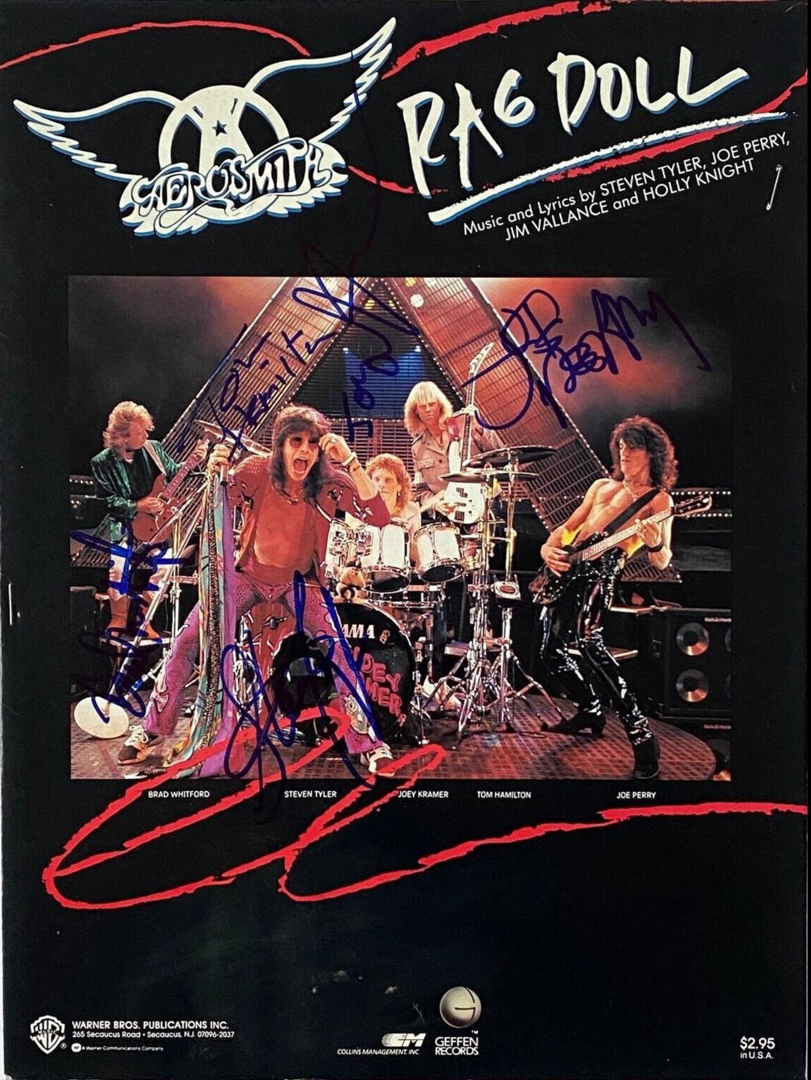
Which album is Dirty Paws from and when was it released?

The song 'Dirty Paws' is from the album 'My Head Is An Animal', which was released in 2018.

The Name

R.A.G





AEROSMITH

RAG DOLL

Music and Lyrics by STEVEN TYLER, JOE PERRY,
JIM VALLANCE and HOLLY KNIGHT

BRAD WHITFORD
STEVEN TYLER
JOEY KRAMER
TOM HAMILTON
JOE PERRY

WARNER BROS. PUBLICATIONS INC.
265 Secaucus Road • Secaucus, N.J. 07096-2037
A Warner Communications Company

COLLINS MANAGEMENT, INC.

GEFFEN RECORDS

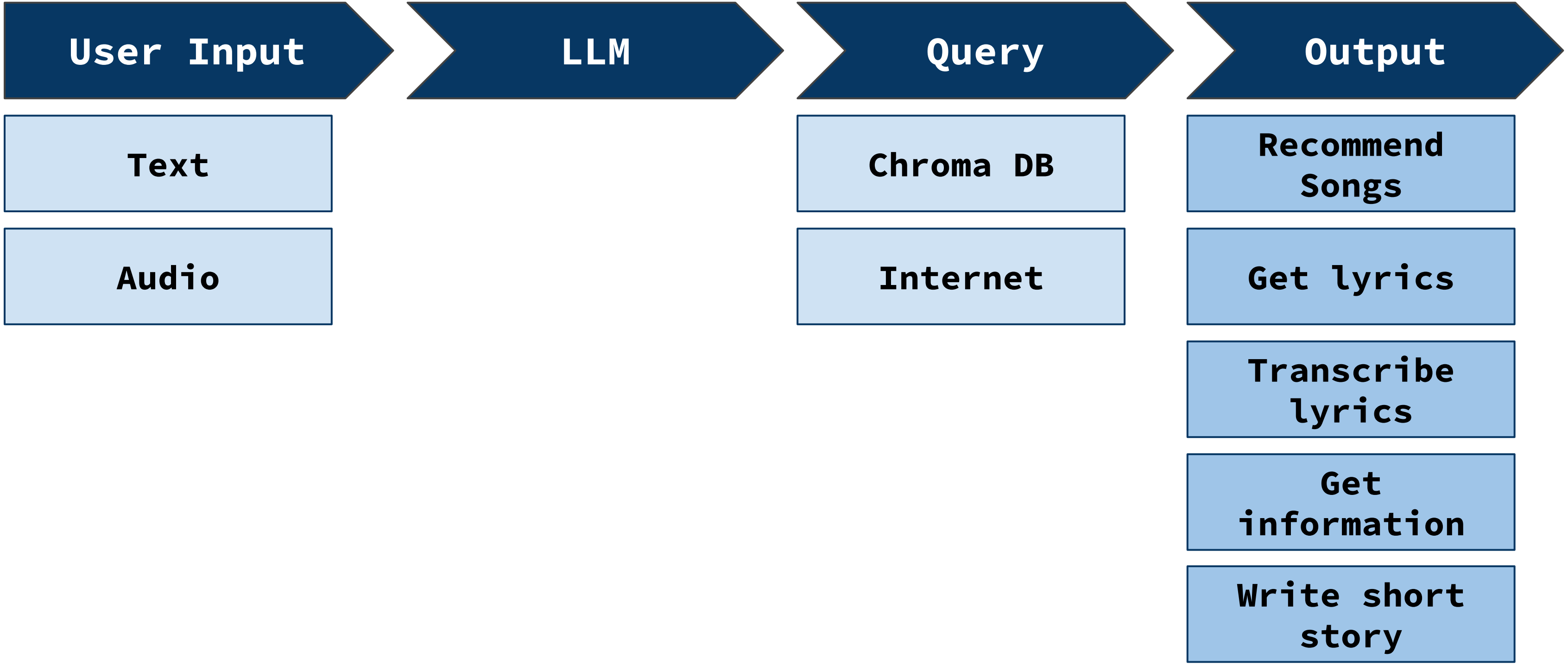
$2.95
in U.S.A.

Aerosmith

And CATS!!!

Technical Details and Functionality

The Basics:



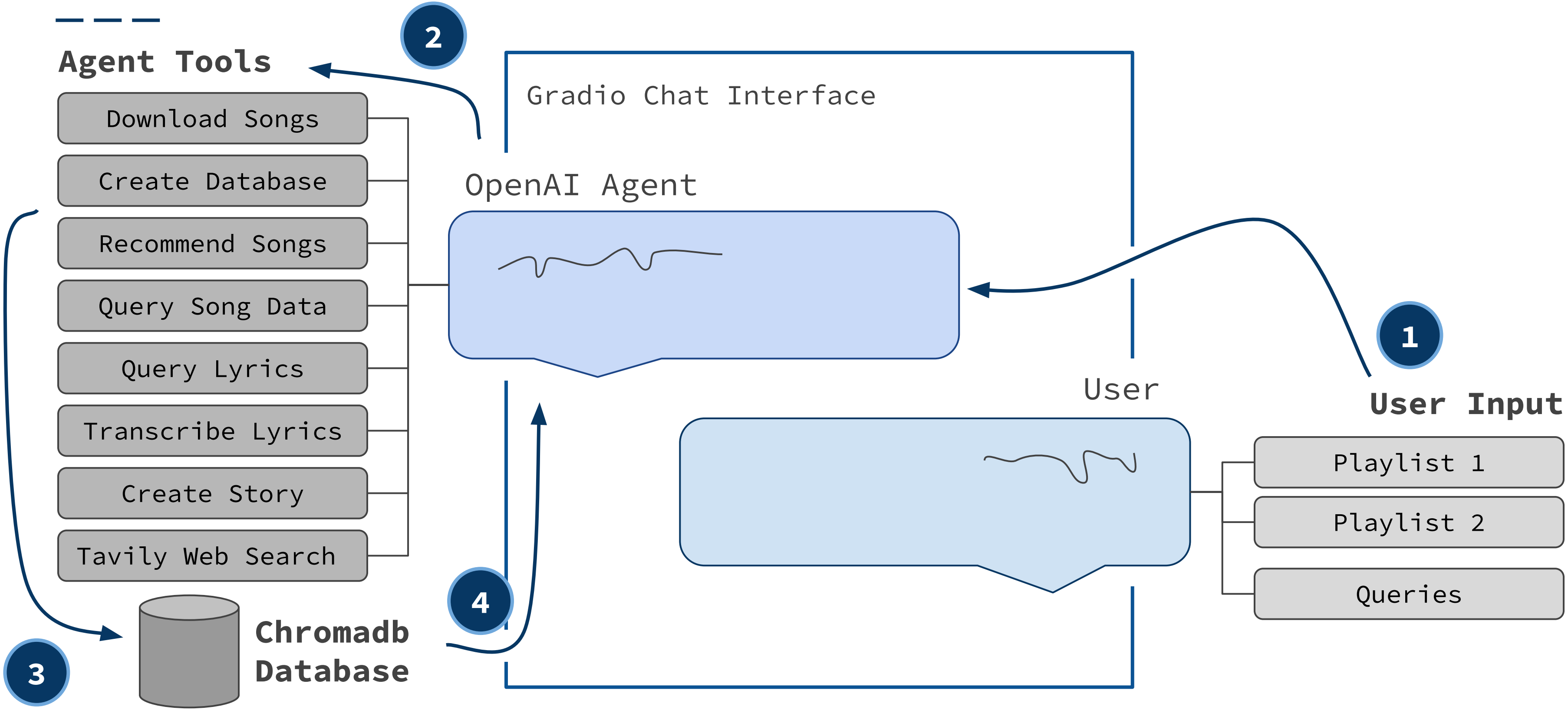
User Input
- Text
- Audio

LLM

Query
- Chroma DB
- Internet

Output
- Recommend Songs
- Get lyrics
- Transcribe lyrics
- Get information
- Write short story

Visualisation of Functionality:



Model Data:

- Model **built** around **YouTube playlists**
    - **User prompted** for **2 url links**
    - **Download** with yt-dlp
    - Could be adjusted to **any suitable dataset**
    - Needs downloadable **audio files** and **metadata**
- **Processing**
    - **Audio** downloaded as **mp3**
    - **Metadata** saved in **consolidated json file**
    - **Artist** and **song name** need to be **parsed** from the **video title** (since not always available in metadata)
    - **Lyrics added** via lyrics.ovh API interface (where available)
    - **Tried various** preprocessing **techniques**1) for audio files to **enhance transcription**, but **pure audio** yielded **best** results

---

1) Audio **processing techniques** included: **splitting vocals** with spleeter, **changing speed** (+/-10%), **normalising** and **limiting volume peaks**, **noise reduction** (pydub, scipy, librosa, soundfile)

Q&A Agent:

- **Model:** gpt-4o-mini
- **Agent:** structured-chat-agent
- **Memory:** custom class to recreate ConversationBufferMemory
- **Tools:** 8 structured tools (incl. downloading songs and creating database)
- **Tried** and **discarded** due to errors:
    - Model: Anthropic Claude Haiku
    - Agent: LangChain Tools Calling Agent
    - Memory: ConversationBuffer(Window)Memory

Tools (1/2):

Tool	Description
Download Tool	**Input:** “**favourites**” & “**recommend**” playlists (two YouTube urls) **Function:** executes the **download**, **data processing** and **lyric search** code **blocks** **Output:** saved **audio** and **metadata files**, one **consolidated metadata file** with lyrics
Database Tool	**Input:** **None** (all parameters **hard-coded**) **Function:** **embeds** audio files and metadata **information into** the initialised **chromadb collection** **Output:** **populated database**
Recommendation tool	**Input:** **optional** (song, artist, n_results) **Function:** **queries database** for **embedding** of **song** or entire favourites **playlist**, **compares embeddings** to songs in **recommends playlist** and gets **top n closest matches** with **metadata** **Output:** **name** and **artist** of the **top n** (default 5) **songs** from **recommend playlist** that most closely match the user request

Tools (2/2):

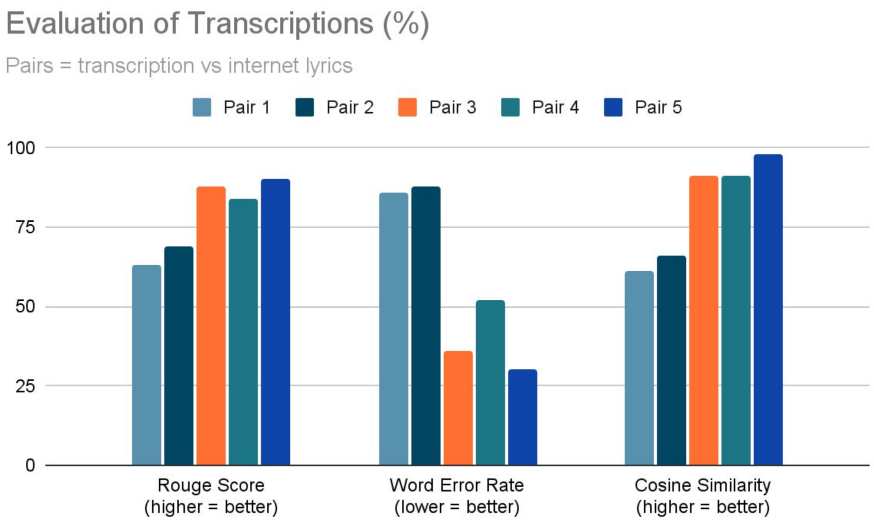
Tool	Description
Song Data Tool/ Lyrics Tool	Input: **song name**, **metadata field** (lyrics tool: “lyrics” default), (artist optional) Function: **queries** the **database** with **embedding** of **given song** Output: **entry** in **requested** metadata **field** (artist, album, release year, lyrics)
Transcription Tool	Input: **song name** Function: uses **faster-whisper** model to **transcribe audio** file of given song, usually **triggered** when **lyrics not found** in the **database** Output: **.txt file** with transcribed text, that is **read** by the **agent**
Story Tool	Input: **song name** (artist optional) Function: uses **lyrics tool** to **get lyrics** of specified **song**, **prompts** the **model** to write a story of **max. 200 words based** on the **lyrics** (starting with “once upon a time..) Output: **short story**
Search Tool	Input: **user query** Function: **Tavily search** run Output: search **result**

Evaluation (1/2):

- **Lots** of **manual** text **inspection**
- **“Chain-of-thought”** prompting and **inspection** of model **thought-process**
- **No** implementation of **LangSmith** or **Giskard**
    - **Information retrieval** was **relatively simple**
    - **No documents**, mainly “lookup” through **hard-coded functions**
    - Evaluation via **LangSmith/ Giskard too complex** for my case
- **Transcription** with whisper **evaluated** via **inspection** but **final evaluation** made by calculating **three metrics:**
    - Rouge Score
    - Word Error Rate
    - Cosine Similarity

Evaluation (2/2):

PERFORMANCE COMPARISON



COMMENTS

- **Transcriptions mainly** evaluated through **inspection** during **experimentation** with **audio processing**
- **Big improvement** when transcribing **raw audio file**
- **3 metrics** subsequently calculated for **5 pairs** of **translated vs online** sourced **lyrics** (Rouge Score, Word Error Rate, Cosine Similarity)
- Metrics **not consistent** across pairs - outcome **inconclusive**
- Transcription **overall very good** (1-2 words incorrect)

Conclusions

Takeaways:

- **Challenges:**
  - Getting model **memory** to **work**
  - **Segmentation** faults (when trying to create database)
  - **Output** parsing (Claude Haiku did not answer in json blobs)
  - **ChatGPT has** its **limits**
  - **Time!** (as always)
- **Possible improvements:**
  - Display **audio files** to be played in chat
  - **Nicer** chat **interface** (custom CSS only worked partially)
  - Work with **multiple playlists** (might already work, didn’t test)
  - Work with **alternative** audio **sources**

Story Time!

Once upon a time, in a land where the sun set in hues of gold and crimson, there stood a grand hotel known as the Hotel California. It was a place of beauty and mystery, where travelers from far and wide would come to rest and revel in the enchanting atmosphere. The hotel was adorned with lush gardens, shimmering fountains, and a haunting melody that seemed to echo through the halls.

One evening, a weary traveler named Juan arrived at the hotel, drawn by its allure. As he stepped inside, he was greeted by the warm smiles of the staff and the intoxicating scent of blooming jasmine. The lobby was filled with laughter and music, and Juan felt an inexplicable sense of belonging.

As the night wore on, Juan mingled with other guests, sharing stories and laughter. However, he soon noticed something peculiar. The guests seemed entranced, as if under a spell, lost in the hotel's charm. They spoke of never wanting to leave, of the comfort and joy that enveloped them.

Curiosity piqued, Juan explored the hotel further. He wandered through opulent ballrooms and dimly lit corridors, each turn revealing more beauty and more secrets. Yet, as he ventured deeper, he felt an unsettling presence, a whisper of something darker lurking beneath the surface.

In the heart of the hotel, he stumbled upon a hidden room, its door slightly ajar. Inside, he found a mirror that reflected not just his image but the essence of his soul. The reflection showed him his dreams, his fears, and the choices he had made. It was then that he realized the hotel was not just a place of rest; it was a trap, a place where time stood still, and the outside world faded away.

Determined to escape, Juan rushed back to the lobby, but the atmosphere had changed. The laughter had turned to whispers, and the guests were now shadows of their former selves, trapped in an eternal cycle of indulgence. He called out to them, urging them to leave, but they only smiled, their eyes vacant.

With a heavy heart, Juan made his way to the exit, but the doors were locked, the windows barred. Panic set in as he realized he was not the first to seek freedom. The hotel had claimed many souls, and now it sought his.\n\nIn a final act of defiance, Juan turned back to the mirror, confronting the truth of his desires. He understood that the only way to break the spell was to embrace his reality, to accept the imperfections of life outside the hotel's walls. With newfound strength, he shattered the mirror, and in that moment, the hotel began to crumble.

As the walls fell away, Juan found himself standing in the moonlight, the hotel a distant memory. He had escaped, but the echoes of the Hotel California would forever linger in his heart, a reminder of the allure of temptation and the importance of choosing one's own path.

Thank You!

Questions?