Towards an AI Rapper: Creating an Interactive Rap Battle Experience with Generative AI

Nikita Kozodoi, PhD
Elizaveta Zinovyeva, PhD
Egor Krasheninnikov
Zainab Afolabi

Helping customers accelerate MORE
Generative AI Innovation Center

NEURAL INFORMATION PROCESSING SYSTEMS

Freestyle Rap Battles

- Verbal competitions between rappers, who **take turns rapping** without pre-written lyrics
- The goal is to **outperform your opponent** with clever wordplay, rhymes, and punchlines
- Each rapper is **given time to perform** while a beat is playing (e.g., 30 seconds)

Can we build an AI rapper to engage in real-time rap battles?

- Generate **rap lyrics** based on an input from a human rapper
- Understand **multiple languages** and provide **expressive audio**



Creating an AI Rapper

Voice
- Speaker profile selection
- Voice cloning

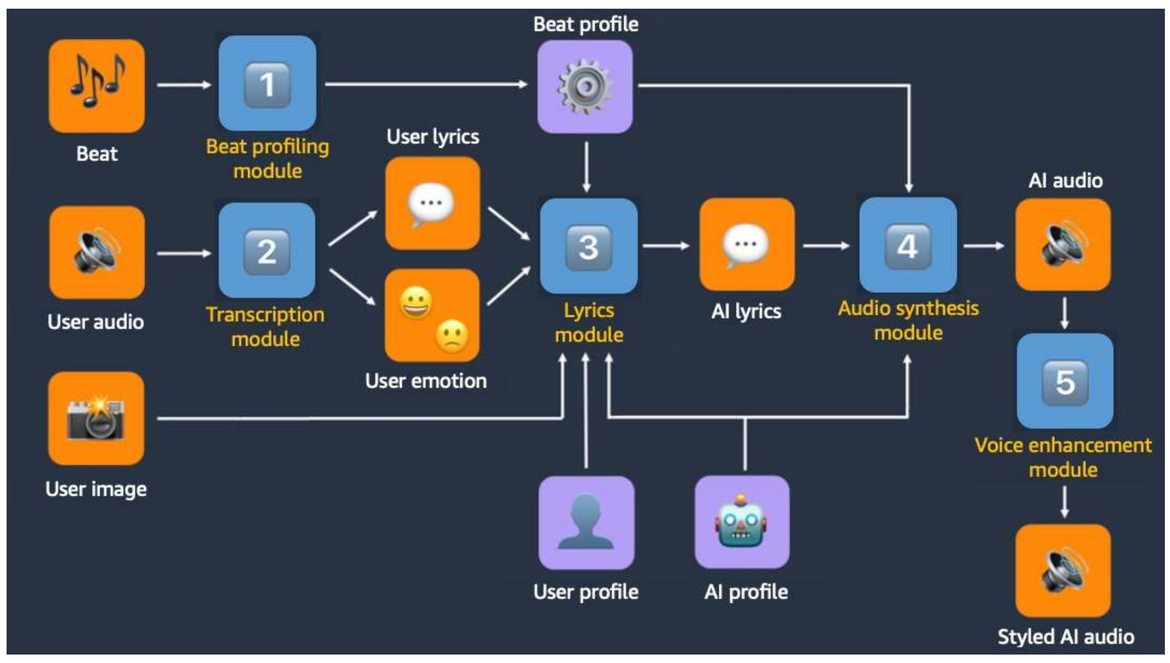
Style
- Profiling the music beat
- Adding emphasis & pauses



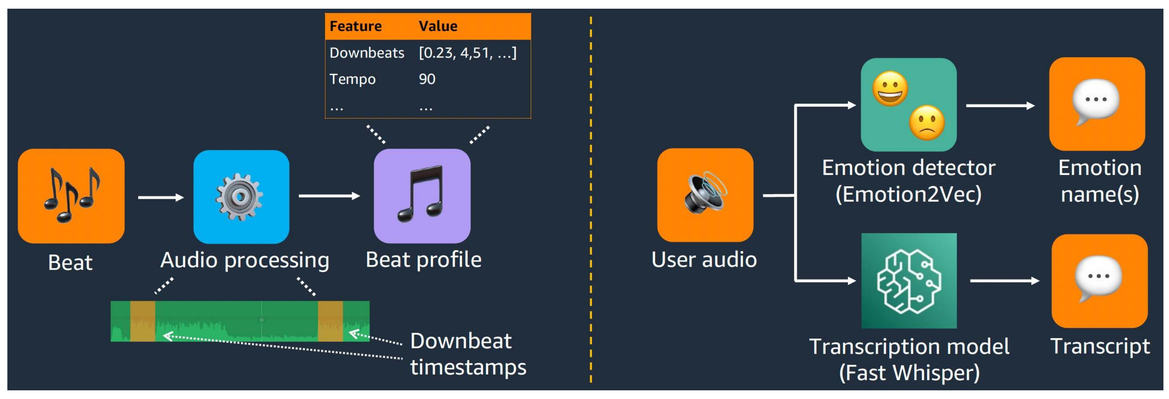
Lyrics
- LLM prompt engineering
  - Rap context and examples
  - Battle memory and opponent bio

Expressiveness
- Detecting user emotion
- Adding special sounds
  - E.g., breath & laughter

Solution Overview



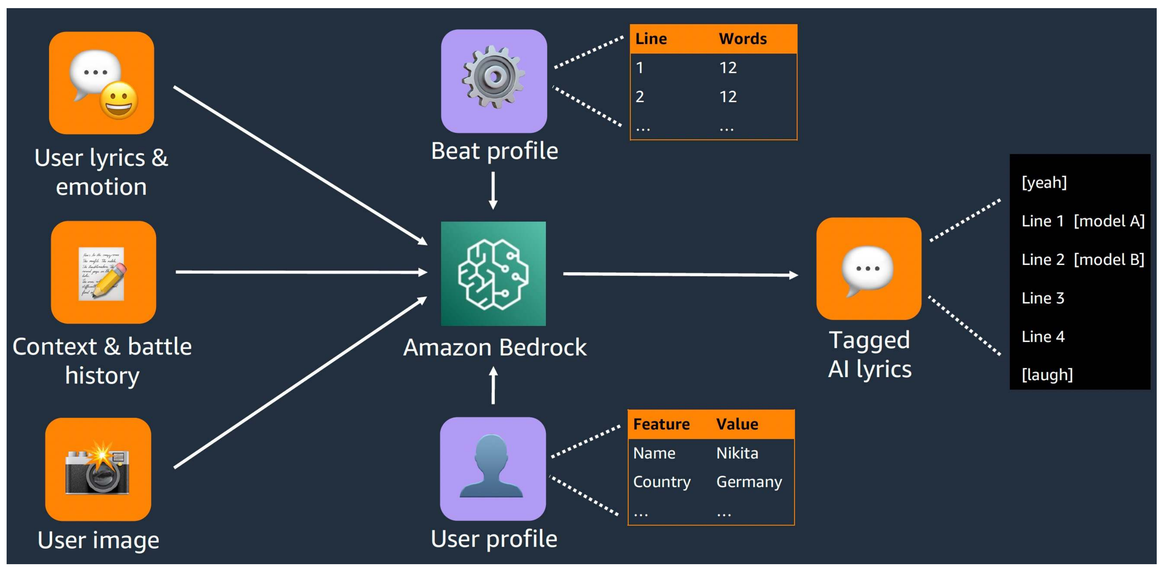
Beat Profiling & Transcription



Beat analysis:
- Librosa: github.com/librosa

Speech recognition:
- Fast Whisper: github.com/Vaibhavs10/insanely-fast-whisper
- Emotion2Vec: github.com/ddlBoJack/emotion2vec

Lyrics Generation & Guardrails



Lyrics generation:
- Amazon Bedrock: aws.amazon.com/bedrock
- Anthropic Claude: anthropic.com/news/claude-sonnet-4-5

Content Check:
- Amazon Bedrock Guardrails

Data Flow

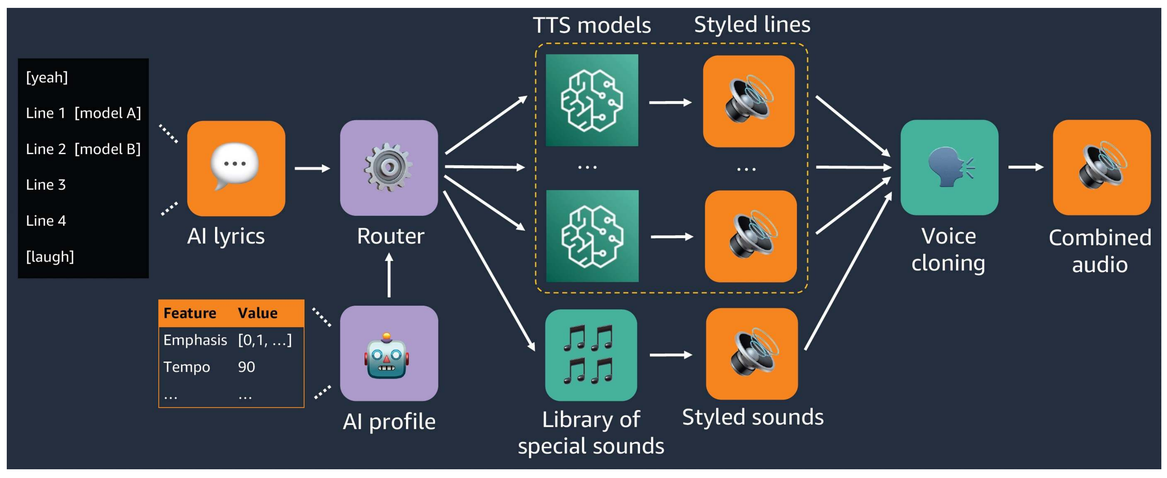
Inputs Processing:

- **Beat audio:** Audio processing tools analyze the beat to extract the metadata describing the audio. The beat profile includes characteristics such as downbeat times and bar length.
- **User audio:** User's vocal input is recorded with the microphone and processed with two ML models: i) audio transcription using Whisper; ii) emotion classification using Emotion2Vec.
- **User image:** Optionally, a user's photo is captured via a camera. The image is provided to a vision LLM on the next step together with the transcript, emotion label, and other text data.
- **Opponent profiles:** User can provide additional data on rapper personas (e.g. country of origin, favorite food, hobbies, etc.). The data is formatted and passed to the LLM as text.

Rap Generation:

- **Lyrics**: Amazon Bedrock LLMs generate creative and context-aware rap lyrics in response to the user's transcript. The prompt describes the lyrics format: i) the beat profile defines the number of lines and line length in syllables; ii) the opponent profiles and the image provide additional context; iii) rap lyrics examples and rap context provide further guidance.
- **Content check**: Amazon Bedrock Guardrails check the LLM output and user inputs for prompt attacks and content filters. If one of the checks fails, we regenerate the lyrics.
- **AI audio:** Lyrics are converted into speech line by line using Chatterbox TTS. The base voice and expressiveness hyperparameters are selected based on the opponent profile. We generate one audio file per line of lyrics. The audios are then augmented with special sounds (e.g., laughter, breath, crowd cheering) sampled from a library of pre-recorded sounds.
- **Music alignment:** The beat profile is used to align each audio line with a corresponding downbeat time in the audio. If the audio duration exceeds the bar length, we speed it up.
- **Voice cloning:** Finally, the individual audios for each line are concatenated into a single audio file. OpenVoice is applied to the AI audio to further enhance the voice expressiveness.

Audio Synthesis & Enhancement



Speech synthesis:
- Chatterbox TTS: github.com/resemble-ai/chatterbox
- OpenVoice: github.com/myshell-ai/OpenVoice

Audio manipulation:
- Pydub: github.com/jiaaro/pydub

LLM Judge

- LLM judge compares lyrics on three dimensions: i) **Creativity**; ii) **Rhymes**; iii) **Personalization**
- The three judge votes are aggregated and used to determine the winner of the battle

Download PDF ←

Learn about our team →

Amazon Web Services, 2025