

artistify

High-Level Design Document

Aadit Mahajan and Anirudh Rao

1 Overview

artistify is a soundtrack generation system that takes a textual narrative, segments it into scenes, detects the emotional "vibe" of each scene, and assigns appropriate songs using semantic similarity. It integrates NLP-based processing with ESA vector space modelling and artist-specific lyric retrieval to recommend the best musical match for each scene.

2 System Objectives

- Automatically generate scene-wise music assignments from a storyline.
- Allow optional artist input to personalise the song recommendations.
- Return similarity-based justification for each song-scene match.

3 Architecture Overview

Frontend/API Consumer → /generate_soundtrack API →

- Scene Segmentation (BERT-based)
- ESA Vector Generation (TF-IDF + Wikipedia Corpus)
- Artist Recommendation (kNN on pre-computed vectors)
- Song Retrieval (Genius Lyrics API)
- Song Assignment (Hungarian Algorithm)

→ Response JSON with scene-song mappings

4 Components and Design Rationale

4.1 Scene Segmentation (`process_storyline.py`)

Scenes are segmented using Sentence-BERT embeddings. Adjacent sentences are compared using cosine similarity. If the similarity drops below a threshold, a new scene begins. This method captures semantic transitions rather than arbitrary breaks, improving the contextual granularity of each scene.

Rationale: Chosen over rule-based segmentation for flexibility and generalisation across domains.

4.2 Vibe Representation (`esa.py`)

Each scene is transformed into a semantic vector using Explicit Semantic Analysis (ESA) based on a lemmatised Wikipedia-derived corpus. This vector embeds the scene in a concept space.

Rationale: ESA offers interpretable, concept-based embeddings and avoids the opaqueness of deep learning models while maintaining decent performance.

4.3 Artist Selection (`model.py`)

If the user does not provide an artist, a nearest-neighbour classifier is used to suggest one. The classifier compares the scene vector with pre-computed ESA vectors for artists (from their top song lyrics) and recommends the most similar ones.

Rationale: Using semantic matching allows artist selection that aligns with narrative mood, offering meaningful relevance beyond genre tags.

4.4 Song Retrieval (`genius_handler.py`, **Genius API**)

The top songs for the chosen artist are fetched from the Genius API. Lyrics are cleaned and passed through the ESA encoder to generate vector representations.

Rationale: Genius offers wide coverage and reliable API access. Lyrics-based ESA ensures lyrical mood alignment.

4.5 Song Assignment (`generate_soundtrack.py`)

Using cosine similarity matrices between scenes and song vectors, the Hungarian algorithm (linear sum assignment) ensures optimal one-to-one assignment maximising total similarity.

Rationale: Hungarian algorithm guarantees global optimum for assignment, preventing greedy local mismatches.

5 Monitoring and Metrics

Prometheus metrics are integrated to monitor:

- Request count
- Time for each major sub-process (scene splitting, vector generation, etc.)

Rationale: Allows performance tuning and bottleneck detection via Grafana or Prometheus dashboards.

6 Technology Stack

- **FastAPI:** Lightweight, asynchronous API framework
- **NLP:** NLTK, Sentence-Transformers, Wikipedia API
- **Lyrics:** Genius API
- **Similarity/Vectorisation:** Scikit-learn, NumPy
- **Monitoring:** Prometheus Client
- **Frontend UI:** Flutter
- **Hosting:** ngrok

7 Extensibility

- Swap ESA for more modern models (e.g., BERT, USE) via abstraction.
- Add multi-song per scene or song transition smoothing.
- Expand recommendation system with genre, tempo, or theme filters.

8 Design Summary

`artistify` offers a modular, interpretable, and efficient architecture for semantic soundtrack generation. Its design prioritises explainability, simplicity, and extensibility. While ESA and cosine similarity provide solid baseline performance, the system allows for future upgrades to transformer-based models or hybrid retrieval strategies.