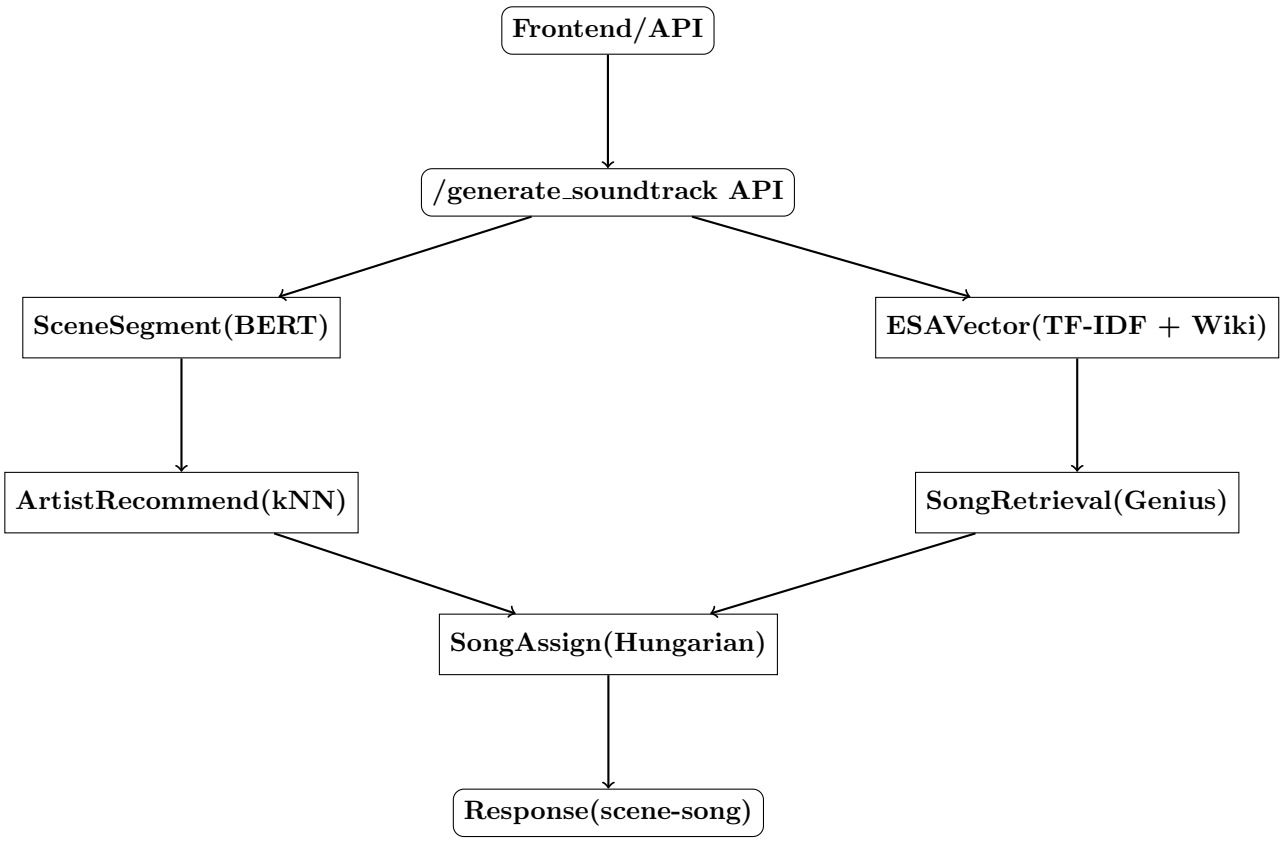
artistify

Architecture Diagram

Aadit Mahajan and Anirudh Rao



Explanation of Architecture Blocks

Frontend/API Consumer

This block represents any client application or user interface that interacts with the `artistify` system. It sends a textual narrative to the `/generate_soundtrack` API and receives the generated scene-song mappings as a response.

/generate_soundtrack API

This is the central entry point of the `artistify` backend. It receives the storyline from the frontend and orchestrates the subsequent processing steps to generate the soundtrack.

Scene Segmentation (BERT-based)

Functionality: This component takes the input textual narrative and divides it into semantically coherent scenes. It utilizes Sentence-BERT embeddings to represent the meaning of each sentence and calculates the cosine similarity between adjacent sentence embeddings. A new scene is demarcated when the similarity between consecutive sentences falls below a predefined threshold, indicating a shift in the narrative's focus or emotion.

Rationale: Using BERT-based embeddings allows for a nuanced understanding of sentence semantics, leading to more contextually relevant scene breaks compared to simpler rule-based methods.

ESA Vector Generation (TF-IDF + Wikipedia Corpus)

Functionality: This block transforms each segmented scene (and potentially song lyrics) into a high-dimensional semantic vector using Explicit Semantic Analysis (ESA). ESA leverages a term-document matrix derived from a lemmatized Wikipedia corpus. For each scene, the TF-IDF (Term Frequency-Inverse Document Frequency) weighted terms are mapped to their corresponding Wikipedia articles, and the resulting vector represents the scene's semantic content within the Wikipedia concept space.

Rationale: ESA provides interpretable semantic representations by linking text to explicit concepts in Wikipedia, offering a balance between performance and explainability compared to opaque deep learning embeddings.

Artist Recommendation (kNN on pre-computed vectors)

Functionality: If the user does not specify an artist, this component suggests relevant artists based on the emotional "vibe" of the input scenes. It employs a k-Nearest Neighbors (kNN) algorithm. Pre-computed ESA vectors representing the typical lyrical themes of various artists (derived from their top songs) are stored. The scene vector is then compared to these artist vectors using cosine similarity, and the k most similar artists are recommended.

Rationale: This allows for personalized soundtrack generation even without explicit artist preference, selecting artists whose lyrical content aligns semantically with the narrative.

Song Retrieval (Genius Lyrics API)

Functionality: Once an artist is chosen (either by the user or the recommendation system), this component interacts with the Genius Lyrics API to fetch the lyrics of the artist's top songs. The retrieved lyrics are then cleaned (e.g., removing bracketed annotations) and passed to the ESA Vector Generation component to create vector representations for each song.

Rationale: Genius API offers a comprehensive database of song lyrics and a reliable way to access them. Using lyrics for vectorization ensures that the musical selection is based on the textual content and emotional tone of the songs.

Song Assignment (Hungarian Algorithm)

Functionality: This component is responsible for optimally matching each scene with an appropriate song. It constructs a cost matrix where each cell (i, j) represents the cosine similarity between the ESA vector of scene i and the ESA vector of song j. The Hungarian algorithm (also known as the Kuhn-Munkres algorithm or linear sum assignment problem) is then applied to this matrix to find a one-to-one assignment of scenes to songs that maximizes the total sum of similarities.

Rationale: The Hungarian algorithm guarantees a globally optimal assignment, preventing suboptimal pairings that might occur with simpler greedy matching strategies.

Response JSON with scene-song mappings

This is the final output of the `artistify` system. It is a JSON (JavaScript Object Notation) object that contains the mapping between each segmented scene from the input narrative and the most semantically similar song selected by the system. This response is sent back to the Frontend/API Consumer.