Music Video Production with Generative AI

Wang Ruixiang
Li Hensheng
Zhong Minjun
Fernando Ramírez González

Technische Universität München

NLP-Projekt Woche

Prof. Dr. Veronika Gamper

18. Juli 2024

Music Video Generation with Generative AI

Introduction

Music video generation is one of the most important parts of the entertainment industry, which can benefit from technological advances. Currently, through generative AI, text, video and images can be generated, which are necessary components for a music video, so if you want to generate it, there is an important challenge considering the duration of the video, as well as the quality of each component depending on the generative model to be used.

This work has the objective of generating a music video from specialized models in images, videos and music, outputting a coherent final product.

Related work

The use of AI (Artificial Intelligence) supported systems has seen a strong increase in recent years due to its positive impact in productivity, allowing workers, artist and companies to reduce the resources and amount of time needed to create text, code, images and videos needed for most of today's digital businesses, for the major part the technology that allows these productivity boost is based on NLP (Natural Language Processing), which focuses on extracting relevant insights from text information, field which saw a dramatic boost with surgency of the Transformer architecture and the Attention Mechanism (Varswani et.al, 2017) which allowed to capture semantic relations by given the same importance to all parts of the inputs and adjusting the weights of the neural network accordingly, shortly after that a lot of people started pretraining using the General Pretrained Transformer (GPT) architecture, which elevated the answer outputs of the Transformer Models. Two years after that, in 2020 Generative Pretrained Transformers became mainstream with the launch of ChatGPT application based on GPT-3.5, the tendency continued and open-source alternatives like Llama-70-B emerged, as of today the state of the art has three main alternatives the first one being ChatGPT-4, which is privately owned model, created by OpenAI, Claude AI 3.5, privately owned model created by Amazon and Llama 3-70B open source model released by Meta. Due to its easy accessibility through open source this last model is the one that was chosen for this task.

On the other side Generative AI, leverages the power of statistical modelling to learn different characteristics of the input data and produce further samples that belong to the learned distribution, a great example of this is Diffusion Models (Rombach, 2022), which use statistical diffusion modelling to generate high quality images from noise, or from text in the case of conditional generation. Naturally, when pairing up diffusion modelling with time we can produce synthetic videos, a recent example of this Stable Video Diffusion (Blatmann, 2023) which incorporates image diffusion modelling and diffusion interpolation to generate 5 second videos, alternative models with higher image quality like LUMA.AI or Runaway Gen3 have appeared, although their papers have not been published yet, they are also available in a freemium version, for the purpose of this task LUMA.AI's Dream Machine and Stable Diffusion v1.0 will be used due to its high quality image generation and coherent motion interpolator.

Another area where generative AI has achieved great results is, audio generation, in this area there are two types of distinguishable models, the models that are focused voice cloning and voice generation and general audio models that can produce signing voices or other sounds like environmental noises or music compositions. These general audio models are often trained on vast datasets containing a wide range of audio samples, allowing them to learn complex patterns and produce diverse outputs beyond just human speech. The generative audio models include Bark, DTS and Suno.

When uniting these technologies one can produce extremely interesting results in the form of music videos, which was the assigned task for this project.

General objective: Produce high-quality coherent musical videos with the use of generative AI.

Specific objectives:

- Ensure coherence throughout the different video scenes.
- Produce appealing lyrics and music.
- Make sure synchronization is correct.

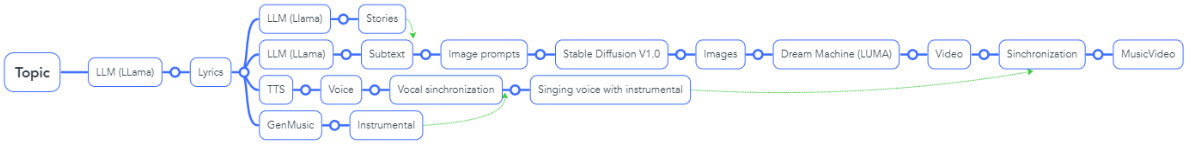
Methodology & Discussion

For the creation of music videos, the task needs to be broken down into 4 main steps:

1. Lyric and text description generation
2. Video/Image generation
3. Music generation
4. Integration of all individual modules

For this reason, we decided to introduce the following workflow

Version 1.0



Over the model selection

For this first approach we decided to focus on open source for feature reproducibility for other people as well as their implied zero cost, the is the reason why the models chosen where the following:

1. Llama 3-7B: This model was used for lyrics, story and image prompt generation, giving access to control of the inputs and outputs via natural language.
2. Stable Diffusion V1.0: This model was used to generate a representative image of each key frame in the video.
3. TTS/Bark: This model where used interchangeably for voice generation.
4. GenMusic: This model was used to produce the instrumental that would be played in the background of each music video.

Code description

From the task breakdown and with the first version of the workflow the code was wrote in the following individual modules:

1. *Language processing related tasks*

- Lyric generation from topic
- Story generation from lyrics
- Image prompt generation
- Utilities for extracting certain necessary inputs from LLM output

For this a class called LLM manager was created, the class is initialized with the topic that one wants to generate and has the following variables:

- Self.intel: This variable describes the initial intel of the large language model that is predefined to be the following:
  ```
  "You are one of the best song writers in the world.\Knowledge
  cutoff: 2021-09-01\nCurrent date: 2023-03-02"
  ```
- Self.together_client: Containts the together api connection to the Llama3 model.
- Self.__intel_story_writer: Containts the intel for the story generation.
- Self.__lyric_prompt: Which stores the basic template for the generation of the lyric and contains the following fields:
    - Song title
    - Genre
    - Tempo
    - Time signature
    - Key
    - Lyrics
    - Audio recording
    - Vocal performance

And the following functions:

- ask_llama_3_8B_together_API: Makes a request to the Llama3 over together AI.
    - Input: prompt
    - Output: LLM response
- generateTextGeneralVideo: Formats the topic into the lyrics prompt and generates the lyrics in the format stated above.
    - Input: topic
    - Output: lyrics
- generateStory: Takes in the lyrics and outputs a story composed a story that serves for generating keyframes.
    - Input: Lyrics
    - Output: Story
- getTitle: Uses regex to extract the song title.
    - Input: Lyrics
    - Output: Song title
- getLyrics: Uses regex to extract the lyrics section.
    - Input: Lyrics

    - Output: Lyrics section
- getTempo: Uses regex to extract the tempo.
    - Input: Lyrics
    - Output: Song tempo
- generateKeyframes: Makes a call to the Llama3 model to convert the story into a set of keyframes in the form of image prompts, by using a predefined template to be filled in.
    - Input: Story
    - Output: set of 6 keyframes description

2. *Image and video generation*
    - Individual frame generation
    - Video animation

    For this task two libraries where created, the first, imageGenerator.py, that contained a class called diffusion with the following variables:

    - Self.__key: Which contains the Together API key.
    - Self.__client: Which contains the Together API client.

    And the following functions:

    - generateImage: which calls Stable Diffusion V1.0 over together API and returns the produced image.
        - Input: image prompt
        - Output: response image as PIL image.

The second library is called utils and contains modules that help make a request to the dreamMachine model, and contains the following functions:

- dreamMachineMake: takes in the image prompt, the access token and the image file and makes a request to the dreamMachine server to produce a 5 second video.
    - Input: video prompt, access_token, image_file_name
    - Output: JSON response for the request
- refreshDreamMachine: function that checks the state of the request over the dreamMachine server, and retries in case of necessary.
    - Input: access_token
    - Output: response in json format
- get_signed_upload: function that checks the state of the request over the dreamMachine server.
    - Input: access_token
    - Output: State_response in json format
- Upload_file: Which allows to upload the image file
    - Input: access_token, file_path
    - Output: None, check is upload was successful

3. *Song and voice generation:*
    - Voice generation
    - Music generation

This module where tested individually, using the following notebooks as reference:

- Bark AI: https://colab.research.google.com/drive/1eJfA2XUa-mXwdMy7DoYKVYHI1iTd9Vkt?usp=sharing
- MusicGen: https://colab.research.google.com/github/camenduru/MusicGen-colab/blob/main/MusicGen_colab.ipynb
- TTS: https://colab.research.google.com/github/camenduru/tortoise-tts-colab/blob/main/tortoise_tts_colab.ipynb
- Then the vocals were extracted and merged using librosa and exported as an .MP3.

4. Integration: All of the functions are integrated over a main function which uses the support of other 3 relevant functions that can be found in the main.py library.
    - uniteTags: That take in the different tags from suno and makes them into only 1 string
        - Input: lyrics, LLManager
        - Ouput: solo string with all tags
    - download_video: helper module that donwolads the LUMA produced video and stores it as an .MP4 output.
        - Input: url, filename,
        - Output: None, downloads and saves the video.
    - Process_topic_completeVideo: Generates the full video using only the topic and the above predefined pipeline.
    - Process_topic: Is the alternative version of process_topic_completeVideo that uses images as the intermediate step for the generation.

5. User interface: The user interface was done over a flask app with a really simple Web UI and is not a finished product.

Experiments

The selected topic for the experiments was: “I met my ex on TikTok”, with that the upper part of the pipeline was run together producing lyrics images and videos, meanwhile the lower part of the pipeline the pipeline was not integrated and run on its own. Integration was not done in this step due to unsatisfactory results from individual components.

Results

From the upper part of the pipeline the following images were obtained:

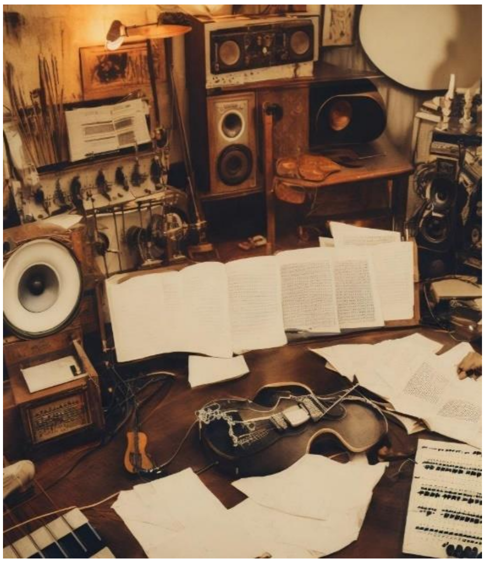


Figure 1. First scene for the song

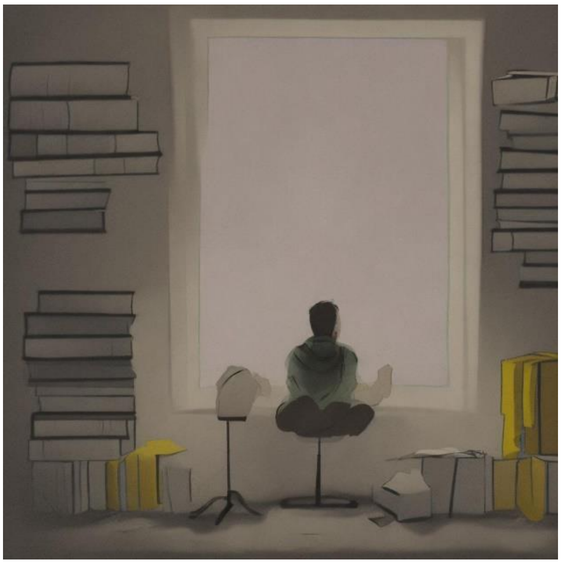


Figure 2. Second scene for the song

From the image it can be clearly observed that they show no relationship between one another, showing a coherence problem and generally low image quality that further extends when doing the animation of image to a 5 second video.

For the lower part of the pipeline, the audio and music generation TTS/Bark models were tested, but they both presented limitation in the generation of singing voices and length of audio.

For the case of TTS, the output was simply "robotic" and not appealing, and bark presented the limitation of only being able to generate audios with a maximum length of 14s. After that MusicGen generated high quality audio but without any singing voice, the vocal synchronization was tried through the key vocals of the song and voice transformation, but no satisfactory results were obtained, since the result was extremely noisy.

With the observations made from the pipeline it was clear that some further refinement was required to fix the image generation inconsistency, low image quality and non-compelling songs, so changes were decided to be made, making the following workflow.

Version 2.0

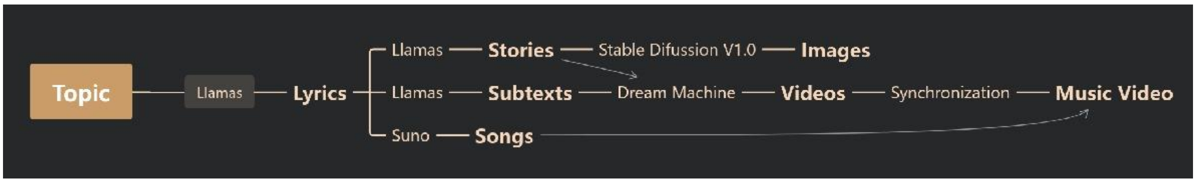


Figure 3. Workflow version 2.0

The module of image, video generation and LLM processing remained the same and were reused for this task.

The main change can be observed in the elimination of the Stable DifussionV1.0 intermediate step for image generation replacing it with a direct prompt to the Dream Machine model and integration of SUNO.AI over a request, in place of the GenMusic and TTS model, for making more compelling songs and allowing for complete integration.

Going deeper over each of the steps:

Language Processing

1. Lyrics generation: For the generation of the lyrics the following template was created:

```
   """Fill in the following template of song lyrics of the topic {topic},
   restrict yourself to filling the template.
   Template:
   **Song Title:** "[title]"
   **Genre:** [genre]
   **Tempo:** [X BPM]
   **Time Signature:** X/4
   **Key:** [Main chord] [Major/Minor]
   **Lyrics:**
   [Verse 1]
   [verse 1 lyrics]
   [Chorus]
   [chorus lyrics]
   **Audio Recording:**
   * [melody, example: nostalgia, longing, melacholy]
   * [beat in the form of adjective list]
   * [rythm in the form of adjective list]
   **Vocal Performance:**
   * [emotions in the form of adjective list]
   * [vocal range in the form of adjective list]
  * [expression in the form of adjective list]
```

This template was defined in order to extract all of the necessary information for the next, steps.

Conversion of lyrics into keyframes: For this purpose, there is a call into the LLM and the following template is filled as response, by using the generateKeyFramesFunction.

Here is the prompt, this prompt as specifically designed through trial and error to ensure high image quality and coherence throughout multiple shots.

```
‟Fill the following template, by describing each line as a musical video
scene, do a maximum of 6 scenes, incorporating detailed physical
appearances, actions, and postures,add the character description to each
scene integrate the character description into the prompt.
From scene 2 onwards if you see any character name repetition please
repeat the same physical description, wrong: the same protagonist, with the
same physical description or [character name], with the same physical
description.
Add at the end of each scene:
Negative prompt: Negative prompt: bad anatomy, bad proportions, blurry,
cloned face, deformed, disfigured, duplicate, extra arms, extra fingers,
extra limbs, extra legs, fused fingers, gross proportions, long neck,
malformed limbs, missing arms.
Lyrics:
{text}
Template:
*Scene 1: Introduction*
Prompt:
Physical scenario: [description]
```

```
Character/s physical description: [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material], [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material]
Character actions: [description]
Plot development: [description]
Negative prompt: bad anatomy, bad proportions, blurry, cloned face, deformed, disfigured, duplicate, extra arms, extra fingers, extra limbs, extra legs, fused fingers, gross proportions, long neck, malformed limbs, missing arms.
*Scene 2: scene 2 title*
Prompt:
Physical scenario: [description]
Character/s physical description: [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material], [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material]
Characters actions: [description]
Plot development: [description]
Negative prompt: bad anatomy, bad proportions, blurry, cloned face, deformed, disfigured, duplicate, extra arms, extra fingers, extra limbs, extra legs, fused fingers, gross proportions, long neck, malformed limbs, missing arms.
*Scene 3: scene 3 title*
Prompt:
Physical scenario: [description]
Character/s physical description: [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material], [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material]
Character/s actions: [description]
Plot development: [description]
Negative prompt: bad anatomy, bad proportions, blurry, cloned face, deformed, disfigured, duplicate, extra arms, extra fingers, extra limbs, extra legs, fused fingers, gross proportions, long neck, malformed limbs, missing arms.
*Scene 4: scene 4 title*
Prompt:
Physical scenario: [description]
Character/s physical description: [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material], [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material]
Character actions: [description]
Plot development: [description]
Negative prompt: bad anatomy, bad proportions, blurry, cloned face, deformed, disfigured, duplicate, extra arms, extra fingers, extra limbs, extra legs, fused fingers, gross proportions, long neck, malformed limbs, missing arms.
*Scene 5: [scene 5 title]*
Prompt:
Physical scenario: [description]
Character/s physical description: [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material], [name][man/woman][age][hair color][hair length][hair style][eye color][face descriptions][height][clothing style][clothing color][clothing material]
```

```
Character actions: [description]
Plot development: [description]]
Negative prompt: bad anatomy, bad proportions, blurry, cloned face,
deformed, disfigured, duplicate, extra arms, extra fingers, extra limbs,
extra legs, fused fingers, gross proportions, long neck, malformed limbs,
missing arms.
*Scene 6: Conclusion*
Prompt:
Physical scenario: [description]
Character/s physical description: [name][man/woman][age][hair color][hair
length][hair style][eye color][face descriptions][height][clothing
style][clothing color][clothing material], [name][man/woman][age][hair
color][hair length][hair style][eye color][face
descriptions][height][clothing style][clothing color][clothing material]
Plot development: [description]
Negative prompt: bad anatomy, bad proportions, blurry, cloned face,
deformed, disfigured, duplicate, extra arms, extra fingers, extra limbs,
extra legs, fused fingers, gross proportions, long neck, malformed limbs,
missing arms."""
first_output = self.ask_llama_3_8b_TOGETHER_API(key_frame_prompt)
#Add the regex pattern for extracting the keyframes as a list
keyframe_list = self.extractKeyFrames(first_output)
return keyframe_list
```

Experiments

Video generation

Videos were generated individually for each keyframe by calling the dreamMachineMake function as described in the main.py, for that regex preprocessing is done for extracting each individual keyframe.

Results:

For the generation of the videoframes the following results were obtained:

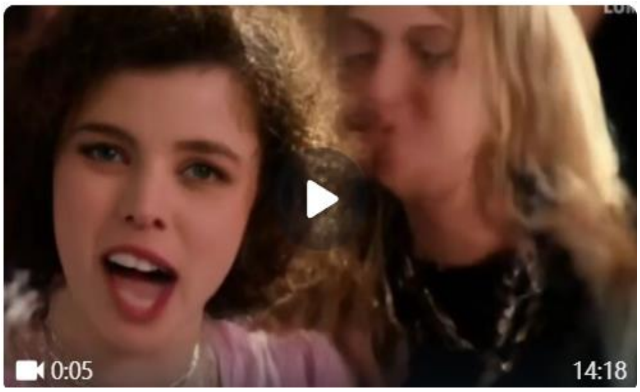


Figure 4. First generated scene by LUMA.AI

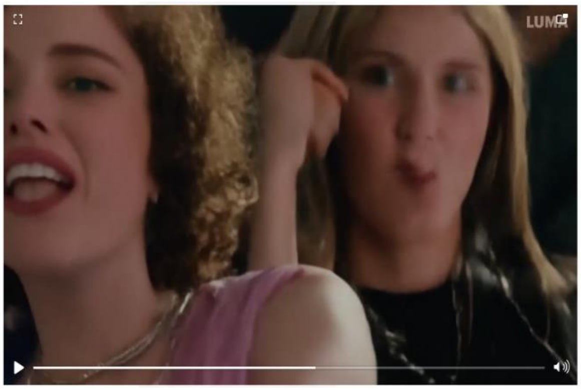


Figure 5. Second generated scene by LUMA.AI

This generation shows further improvements in the coherence and image quality departments due to the integration of a predefined template and the negative prompts.

This can also be seen across an add video of T-Shirts.

Music generation

For the music generation we incorporated a SUNO call via requests for that purpose a library called suno_api was created, using the same style of the dreamMachineModule this module comprises, 3 relevant parameters.

- Lyrics: Which is the actual lyrics of the song
- Tags: Where parameters from the vocal performance, tempo and key were used to generate the song
- Title: The song title

Results:

Multiple experiments were done over varying tags, but it was found out the most effective tags are the following:

- Style: Electronic or rock
- Key: C Major/ G Major
- Tempo 4/4
- BPM: 110- 120 BPM depending on the style of the song
- Voice: Female, male are both good

For further results an alternative song style can be found inside the results folder.

Video synchronization

Due the varying length of the different components of the song, the synchronization was done manually via, video editing.

Relevant remarks:

- LUMA.AI allows a maximum of 5 videos generated per day so multiple google accounts had to be created
- Suno only allows for 10 songs per day
- The video was synchronized manually.

Conclusion

We successfully achieved the generation of a realistic, coherent music video, with appealing music and high image quality, still there is the need for manual synchronization due to the inability of the models to change output length at will from the video.

Reference

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., ... & Rombach, R. (2023). Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

GitHub - mikezzb/lyrics-sync: A deep learning lyrics-to-audio alignment system, generating synchronized lyrics from a song and its lyrics. GitHub

Llama. (n.d.). *Getting started with Llama | Documentation*. Retrieved from: https://llama.meta.com/docs/model-cards-and-prompt-formats/meta-llama-2

OpenAI. (n.d.). *New models and developer products announced at DevDay*. Retrieved from: https://openai.com/index/new-models-and-developer-products-announced-at-devday/

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023, July). Robust speech recognition via large-scale weak supervision. In *International conference on machine learning* (pp. 28492-28518). PMLR.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684-10695).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.