



Write code to create a text-to-speech for Hindi language with a state-of-the-art AI model.



To create a text-to-speech for Hindi language with a state-of-the-art AI model, you can use a library or framework like OpenAI's GPT-3 or DeepSpeech. Here is an example of how you might use GPT-3 to create a text-to-speech function in Hindi:



Copy code

```
import openai

# Set the API key for openai
openai.api_key = "<your-api-key>"

# Define the text-to-speech function
def text_to_speech(text: str, language: str) -> str:
    response = openai.Completion.create(
        engine="text-davinci-002",
        prompt=f"text-to-speech: {text} in {language}",
        max_tokens=1024,
        temperature=0.5,
    )
```

```
# Return the generated speech in the specified language
return response["choices"][0]["text"]

# Use the function to generate speech in Hindi
hindi_speech = text_to_speech("नमस्ते दुनिया", "hi")
print(hindi_speech)
```

Note that this is just an example and may not produce the best results. You may need to experiment with different settings and parameters to get the best text-to-speech output for Hindi.



Is there a way to melodise the generated speech to make it sound like a gong (without having data for the melody)??



To melodize the generated speech to make it sound like a gong, you could try using a synthesizer or audio effects processor to add a gong-like melody to the speech. This would typically involve using a synthesizer to create a gong sound and then using audio effects like pitch shifting and delay to create a melody.



Alternatively, you could try using a machine learning model trained on musical data to generate a gong melody and then combine it with the generated speech. However, this would require having a dataset of gong melodies to train the model on, which you mentioned you do not have.

It is also worth noting that adding a melody to the generated speech may not always produce the desired effect, as the melody and speech may not always be harmonically compatible. In general, it is difficult to accurately control the melody of a generated speech without having specific training data for the desired melody.



Can you provide me the code to do this?



As mentioned earlier, adding a melody to the generated speech would typically involve using a synthesizer or audio effects processor to create a gong sound and then using audio effects like pitch shifting and delay to create a melody. Here is an example of how you might do this using the `pydub` library in Python:



Copy code

```
from pydub import AudioSegment
from pydub.playback import play

# Load the generated speech as an audio file
speech = AudioSegment.from_file("generated_speech.mp3")

# Use a synthesizer to create a gong sound
gong = AudioSegment.from_synth("gong.mp3")

# Use pitch shifting and delay effects to create a melody with the gong
sound

melody = gong.pitch_shift(semitones=2) + gong.delay(duration=100)

# Combine the melody with the generated speech
melodized_speech = speech.overlay(melody)

# Play the melodized speech
play(melodized_speech)
```

Note that this is just an example and may not produce the best results. You may need to experiment with different synthesizers, effects, and parameters to get the desired melodized speech. You may also need to adjust the volume levels of the different audio tracks to ensure that the melody and speech are balanced.

As mentioned earlier, using a machine learning model trained on musical data to generate a gong melody and then combining it with the generated speech is another option, but it would require having a dataset of gong melodies to train the model on.