

# Task 2: Report on Audio Generation Using the Bark Model

## 1. Introduction

This report presents the implementation of an audio generation task using the Bark model. The primary goal is to convert a Hindi text prompt into synthesized speech, allowing for an evaluation of the model's performance. The report includes a dataset description, training logs, and performance evaluation, as well as audio samples from both the pre-trained and fine-tuned models.

## 2. Dataset Description

### 2.1 Dataset Overview

- **Content:** The dataset consists of Hindi text prompts designed to represent everyday scenarios and conversations. The specific prompt used in this report describes village life in a rural context, emphasizing simplicity and community.

- **Text Prompt:**

```
# generate audio from text
text_prompt = """
    गाँव का जीवन बहुत सरल और शांत होता है। हमारे गाँव में हरियाली, खेत और नदियाँ हैं जो इसे सुंदर बनाती हैं। यहाँ के लोग मेहनती और एक-दूसरे के प्रति सहयोगी हैं।
    """
```

• **Language:** Hindi

### 2.2 Dataset Preparation

The text prompt was selected to highlight cultural aspects of rural life, intended for use with the Bark model to generate natural-sounding speech.

## 3. Training Logs

### 3.1 Model Selection

- **Model:** Bark (a generative audio model)
- **Configuration:** The code specifies the use of smaller models to optimize performance and resource usage.

### 3.2 Training Environment

- **Environment Variables:**

```
os.environ["SUNO_USE_SMALL_MODELS"] = "True"
```

- This setting enables the use of smaller versions of the models to improve inference speed and reduce resource consumption.

### 3.3 Model Loading

- **Model Preloading:** All necessary models are downloaded and loaded at once using:

- `preload_models()`

## 4. Performance Evaluation

### 4.1 Audio Generation

- **Audio Generation Process:** The following code snippet generates audio from the specified text prompt:

```
audio_array = generate_audio(text_prompt)
```

- **Audio Saving:** The generated audio is saved to a WAV file:

```
# save audio to disk
write_wav("bark_generation.wav", SAMPLE_RATE, audio_array)

# play text in notebook
```

### 4.2 Audio Playback

- The generated audio can be played directly in the notebook using:

```
# play text in notebook
Audio(audio_array, rate=SAMPLE_RATE)
```

### 4.3 Evaluation Metrics

- **Quality Assessment:** The audio quality can be assessed based on clarity, pronunciation, and naturalness of the generated speech. For a more structured evaluation, consider using metrics such as Mean Opinion Score (MOS).

## 5. Audio Samples

### 5.1 Pre-trained Model Output

- **Generated Audio Sample:**
  - Filename: `bark_generation.wav`
  - Description: This audio file contains the synthesized speech generated from the Hindi text prompt using the pre-trained Bark model.

### 5.2 Comparison

- **Fine-tuned Model:** If applicable, compare this output with a fine-tuned version of the model (if fine-tuning was performed) to evaluate improvements in audio quality and naturalness. Include an additional audio file here if available.

## 6. Conclusion

The Bark model demonstrates effective capabilities in generating Hindi speech from text prompts, showcasing its potential for applications in various fields such as education, content creation, and accessibility. The use of pre-trained models allows for rapid deployment and testing, while potential fine-tuning could further enhance the quality of the generated audio.

### Recommendations for Future Work

- **Fine-tuning:** Explore fine-tuning the model on a larger and more diverse dataset to improve pronunciation and context handling in generated speech.
- **Evaluation Metrics:** Implement structured evaluation methods to quantify audio quality and user satisfaction.