

2121-8/japanese-parler-tts-large-bate

like 21



Text-to-Audio



Transformers



PyTorch



ylacombe/libritts_r_filtered



ylacombe/libritts-r-filtered-descriptions-10k-v5-without-accent



Japanese

parler_tts

text-generation

text-to-speech

annotation

japanese



License: other



Train

Deploy



Use this model



Model card

Files

xet



Community 2

Downloads last month

461



Inference Providers

NEW

Text-to-Audio

This model isn't deployed by any Inference Provider.



Ask for provider support

Model tree for 2121-8/japanese-parler-tts-large-bate

Base model

parler-tts/parler-tts-large-v1



Finetuned (1)

this model

Datasets used to train 2121-8/japanese-parler-tts-large-bate

Space using 2121-8/japanese-parler-tts-large-bate 1

Collection including 2121-8/japanese-parler-tts-large-bate

Japanese Parler-TTS

Collection

Parler-TTSを基に、日本語でのテキスト... • 12 items • Updated Apr 26 • Δ 3

Japanese Parler-TTS Large (β版)

このリポジトリは、[parler-tts/parler-tts-large-v1](#)を基に、日本語でのテキスト読み上げを可能にするよう再学習したモデルを公開しています。本モデルは、軽量でありながら高品質な音声生成を提供します。

注意: 本家の[Parler-TTS](#)で使用されているtokenizerとは互換性がありません。本モデル用に独自のtokenizerが採用されています。

現在このリポジトリはβ版です。機能やモデルの最適化は正式リリースに向けて進行中です。

正式リリース版のURL

- [Japanese Parler-TTS Mini \(878M\)](#).
- Japanese Parler-TTS Large (878M) 準備中

🔗 Japanese Parler-TTS Index

- [Japanese Parler-TTS Mini](#)
- Japanese Parler-TTS Large (計算資源に余裕があったら学習します)
- [Japanese Parler-TTS Mini Bata](#)
- [Japanese Parler-TTS Large Bata](#)

🔗 注意点

Japanese Parler-TTS Large は音声表現力が非常に豊かで高品質な音声生成が可能ですが、学習不足のため動作が不安定な場合があります。そのため、安定性を重視する場合は、より軽量で動作が安定している Japanese Parler-TTS Mini の使用を推奨します。

本モデルは学習データの構成上、男性の声に関するデータが少ないため、男性の声の生成が期待通りに行えない場合があります。特に、自然なイントネーションや音質の調整が難しい場合がありますので、ご了承ください。

[🔗 📖 クイックインデックス](#)

- [👤 インストール](#)
- [🎲 ランダムな音声での使用方法](#)
- [🎯 特定の話者を指定する方法](#)
- [🙏 謝辞](#)

[🔗 🛠️ 使用方法](#)

[🔗 👤 インストール](#)

以下のコマンドでインストールできます。

```
pip install git+https://github.com/huggingface/parler-tts.git
pip install git+https://github.com/getuka/RubyInserter.git
```

[🔗 🎲 ランダムな音声での使用方法](#)

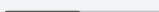
```
import torch
from parler_tts import ParlerTTSForConditionalGeneration
from transformers import AutoTokenizer
import soundfile as sf
from rubyinserter import add_ruby

device = "cuda:0" if torch.cuda.is_available() else "cpu"

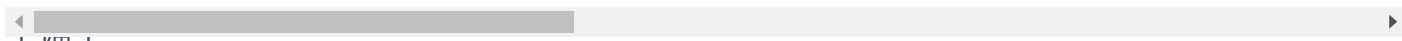
model = ParlerTTSForConditionalGeneration.from_pretrained("2121-8/japanese-parler-
tokenizer = AutoTokenizer.from_pretrained("2121-8/japanese-parler-tts-large-bate")
```

```
prompt = "こんにちは、今日はどのようにお過ごしですか？"  
description = "A female speaker with a slightly high-pitched voice delivers her w  
  
prompt = add_ruby(prompt)  
input_ids = tokenizer(description, return_tensors="pt").input_ids.to(device)  
prompt_input_ids = tokenizer(prompt, return_tensors="pt").input_ids.to(device)  
  
generation = model.generate(input_ids=input_ids, prompt_input_ids=prompt_input_ids  
audio_arr = generation.cpu().numpy().squeeze()  
sf.write("parler_tts_japanese_out.wav", audio_arr, model.config.sampling_rate)
```

🔗 サンプル音声

▶ 0:00 / 0:05  🔊 ⋮

🔗 🎯 特定の話者を指定する方法



🔗 謝辞

本モデルの開発にあたり、以下の資源提供をしていただいた方々に感謝いたします。

- [saldra](#)
- [Witness](#)

彼らの貢献がなければ、このプロジェクトは実現しませんでした。

🔗 著作権および使用に関する免責事項

本モデルおよびリポジトリは、研究、教育、商用利用を含む幅広い目的での利用が許可されています。ただし、以下の条件を遵守してください。

1. 商用利用に関する条件

本モデルを使用して生成された音声や成果物を商用目的で利用することは可能ですが、本モデルそのもの（ファイルや重みデータなど）の販売は禁じられています。

2. 適切性についての免責

本モデルの利用により得られる結果の正確性、合法性、または適切性について、作成者は一切保証しません。

3. ユーザーの責任

本モデルを使用する際は、適用されるすべての法律や規制を遵守してください。また、生成されたコンテンツに起因する責任はすべてユーザーに帰属します。

4. 作成者の免責

本リポジトリおよびモデルの作成者は、著作権侵害やその他の法的問題に関する責任を一切負いません。

5. 削除要求への対応

著作権問題が発生した場合、問題のあるリソースやデータを速やかに削除します。