

 suno / **bark** 

 like 1.45k

Follow  Suno 474



Text-to-Speech



Transformers



PyTorch



13 languages

bark

text-to-audio


audio




License: mit



 Train ▾

 Deploy ▾

 Use this model ▾



Model card

 Files

 xet



Community 60

Downloads last month

22,812



## ⚡ Inference Providers NEW

 Text-to-Speech

This model isn't deployed by any Inference Provider.



10

Ask for provider support

## 🗃 Model tree for suno/bark

Finetunes

2 models

 Spaces using suno/bark 100

## 🗃 Collection including suno/bark

**Bark** Collection

Bark is a transformer-based text-to-audio ... • 3 items • Updated Sep 14, 2023 • △ 18

## 🔗 Bark

Bark is a transformer-based text-to-audio model created by [Suno](#). Bark can generate highly realistic, multilingual speech as well as other audio - including music, background noise and simple sound effects. The model can also produce nonverbal communications like laughing, sighing and crying. To support the research community, we are providing access to pretrained model checkpoints ready for inference.

The original github repo and model card can be found [here](#).

This model is meant for research purposes only. The model output is not censored and the authors do not endorse the opinions in the generated content. Use at your own risk.

Two checkpoints are released:

- [small](#)
- [large \(this checkpoint\)](#)

## [🔗](#) Example

Try out Bark yourself!

- Bark Colab:



- Hugging Face Colab:



- Hugging Face Demo:



## [🔗](#) 🧠 Transformers Usage

You can run Bark locally with the 🧠 Transformers library from version 4.31.0 onwards.

1. First install the 🧠 [Transformers library](#) and scipy:

```
pip install --upgrade pip
pip install --upgrade transformers scipy
```

2. Run inference via the Text-to-Speech (TTS) pipeline. You can infer the bark model via the TTS pipeline in just a few lines of code!

```
from transformers import pipeline
import scipy

synthesiser = pipeline("text-to-speech", "suno/bark")

speech = synthesiser("Hello, my dog is cooler than you!", forward_params={"do_sample": True})

scipy.io.wavfile.write("bark_out.wav", rate=speech["sampling_rate"], data=speech["data"])
```

3. Run inference via the Transformers modelling code. You can use the processor + generate code to convert text into a mono 24 kHz speech waveform for more fine-grained control.

```
from transformers import AutoProcessor, AutoModel

processor = AutoProcessor.from_pretrained("suno/bark")
model = AutoModel.from_pretrained("suno/bark")

inputs = processor(
    text=["Hello, my name is Suno. And, uh – and I like pizza. [laughs] But I also have some tricks up my sleeve!"],
    return_tensors="pt",
)

speech_values = model.generate(**inputs, do_sample=True)
```

4. Listen to the speech samples either in an ipynb notebook:

```
from IPython.display import Audio

sampling_rate = model.generation_config.sample_rate
Audio(speech_values.cpu().numpy().squeeze(), rate=sampling_rate)
```

Or save them as a `.wav` file using a third-party library, e.g. `scipy`:

```
import scipy

sampling_rate = model.config.sample_rate
scipy.io.wavfile.write("bark_out.wav", rate=sampling_rate, data=speech_values.cpu()
```

For more details on using the Bark model for inference using the 🧠 Transformers library, refer to the [Bark docs](#).

## 🔗 Suno Usage

You can also run Bark locally through the original [Bark library](#):

1. First install the [bark library](#).
2. Run the following Python code:

```
from bark import SAMPLE_RATE, generate_audio, preload_models
from IPython.display import Audio

# download and load all models
preload_models()

# generate audio from text
text_prompt = """
    Hello, my name is Suno. And, uh – and I like pizza. [laughs]
    But I also have other interests such as playing tic tac toe.
"""

speech_array = generate_audio(text_prompt)

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

[pizza.webm](#)

To save `audio_array` as a WAV file:

```
from scipy.io.wavfile import write as write_wav

write_wav("/path/to/audio.wav", SAMPLE_RATE, audio_array)
```

## [🔗](#) Model Details

The following is additional information about the models released here.

Bark is a series of three transformer models that turn text into audio.

### [🔗](#) Text to semantic tokens

- Input: text, tokenized with [BERT tokenizer from Hugging Face](#)
- Output: semantic tokens that encode the audio to be generated

### [🔗](#) Semantic to coarse tokens

- Input: semantic tokens
- Output: tokens from the first two codebooks of the [EnCodec Codec](#) from facebook

### [🔗](#) Coarse to fine tokens

- Input: the first two codebooks from EnCodec
- Output: 8 codebooks from EnCodec

## [🔗](#) Architecture

Model	Parameters	Attention	Output Vocab size
Text to semantic tokens	80/300 M	Causal	10,000
Semantic to coarse tokens	80/300 M	Causal	2x 1,024

Model	Parameters	Attention	Output Vocab size
Coarse to fine tokens	80/300 M	Non-causal	6x 1,024

## [🔗](#) Release date

April 2023

## [🔗](#) Broader Implications

We anticipate that this model's text to audio capabilities can be used to improve accessibility tools in a variety of languages.

While we hope that this release will enable users to express their creativity and build applications that are a force for good, we acknowledge that any text to audio model has the potential for dual use. While it is not straightforward to voice clone known people with Bark, it can still be used for nefarious purposes. To further reduce the chances of unintended use of Bark, we also release a simple classifier to detect Bark-generated audio with high accuracy (see notebooks section of the main repository).