Whisper　高速化したfaster-whisperを簡単に動かしてみる

2023年3月25日 2023年3月27日

kaji0620

Whisper　音声・動画の自動書き起こしAIを無料で、簡単に使おうの記事を紹介していましたが、高速化された「Faster-Whisper」が公開されていましたので、Google Colaboratoryで実装していきます。

また、「large-v2」と言うアップデートされたモデルが提供されていました。
こちらも合わせて使用してみたいと思います。

目次

Faster-Whisper

これまでのWhisperとの違いを記載していきたいと思います。
Faster-Whisperから引用します。

> faster-whisper is a reimplementation of OpenAI's Whisper model using CTranslate2, which is a fast inference engine for Transformer models.
>
> This implementation is up to 4 times faster than openai/whisper for the same accuracy while using less memory. The efficiency can be further improved with 8-bit quantization on both CPU and GPU.

> faster-whisperは、OpenAIのWhisperモデルを、Transformerモデルの高速推論エンジンであるCTranslate2を使って再実装したものです。

この実装は、同じ精度でopenai/whisperより最大4倍高速で、かつ少ないメモリしか使用しません。CPUとGPUの両方で8ビット量子化を行うことで、さらに効率を向上させることができます。

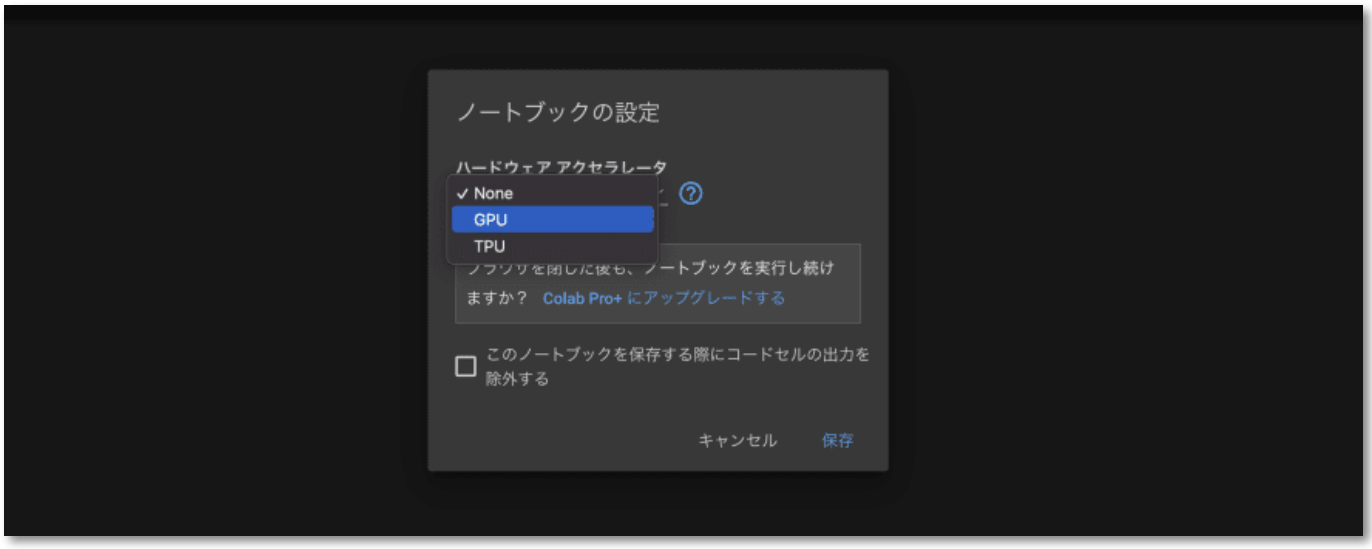
Large-v2 model on GPU

Implementation	Precision	Beam size	Time	Max. GPU memory	Max. CPU memo
openai/whisper	fp16	5	4m30s	11325MB	9439MB
faster-whisper	fp16	5	54s	4755MB	3244MB

CTranslate2と言う高速推論エンジンで変換したモデルを実装することで、
処理速度が約4倍に早くなって、メモリーの使用率は約1/3になったみたいです。

Fast-Whisperの実装

最初に、「編集」→「ノートブックの設定」から、
ハードウェアアクセラレートを「GPU」に設定しましょう。



以下のコマンドを入力し、Shift+Enterを押して実行しましょう。
GPUの情報が出てくれば、正常に設定変更できています。

```
!nvidia-smi
```

続けてコードを記載してきます。
これまでのWhisperの使用方法については、Whisper　音声・動画の自動書き起こしAIを無料で、簡単に使おうを参照ください。

```
!pip install git+https://github.com/guillaumekln/faster-whisper.git
```

```
from faster_whisper import WhisperModel

model_size = "large-v2"

# Run on GPU with FP16
model = WhisperModel(model_size, device="cuda", compute_type="float16")

# or run on GPU with INT8
# model = WhisperModel(model_size, device="cuda", compute_type="int8_float16")

# or run on CPU with INT8
# model = WhisperModel(model_size, device="cpu", compute_type="int8")
```

以前のCHATGPT + SHOTCUT　半自動で動画を作成してみようの記事で使用した動画を用います。

```
! pip install yt-dlp
! rm input.mp3
! yt-dlp -x --audio-format mp3 https://youtu.be/1XQVbMz4j-0 -o "input.mp3"
```

```
segments, info = model.transcribe("input.mp3", beam_size=5)

for segment in segments:
    print("[%.2fs -> %.2fs] %s" % (segment.start, segment.end, segment.text))
```

うん、早いぞ。
これまでのWhisperとは出力結果の吐き方が異なるのかな？（要勉強です…）

Fast-Whisperは、WhisperModel(“large-v2”)のようなモデルを読み込む場合、対応するCTranslate2モデルはHugging Face Hubから自動的にダウンロードしているとのことです。

もし変換したCTranslate2モデルが手元に欲しい場合は、以下のコマンドで作成することが可能です。
実行後「whisper-large-v2-ct2」フォルダに変換されたモデルが作成されます。

```
!pip install git+https://github.com/guillaumekln/faster-whisper.git
!pip install transformers

!ct2-transformers-converter --model openai/whisper-large-v2 --output_dir
whisper-large-v2-ct2 --quantization float16
```

この変換したモデルを用いてfast-whisperを動かす場合、「model_size」を作成されたフォルダのパスに変更するだけでOKです。

```
!pip install git+https://github.com/guillaumekln/faster-whisper.git
```

```
from faster_whisper import WhisperModel

# 変換されたモデルの格納されたフォルダを指定する
model_size = "whisper-large-v2-ct2/"

# Run on GPU with FP16
model = WhisperModel(model_size, device="cuda", compute_type="float16")

# or run on GPU with INT8
# model = WhisperModel(model_size, device="cuda", compute_type="int8_float16")

# or run on CPU with INT8
# model = WhisperModel(model_size, device="cpu", compute_type="int8")
```

以後は、先ほどと同じになりますので、省略します。

確かに早いですしメモリの使用量が少ないので、ローカルで動かすことを考えると便利です。
ひとまず、ここまで。

この記事の投稿者　**最新記事**