SRA の技術情報誌
GS LetterNeo
Vol.189　2024 年 4 月

# Hugging Face Transformers を試してみました

佐藤 峻矢

アドバンストクラウドエンジニアリング事業部

## はじめに

近年の AI ブームにより、毎日のように AI に関する話題を聞くようになりました。ChatGPT に代表されるように AI 技術の活用も盛んになってきました。そこで AI モデルを簡単に試したい、チューニングしてモデルを作成したいと考える方も多いでしょう。

一方で学習済みモデルについては、Git hub に研究者やエンジニアなど、有志の方がそれぞれモデルを作成し、アップロードしているため、実行環境、フレームワーク、モジュール等の依存関係、データの前処理が存在し、ローカル環境に展開し実行することが難しいという課題がありました。これらの課題を解決し、標準化されたインタフェースを提供するため Hugging Face 社から Transformers が提供されています。

# Hugging Face Transformers とは

Hugging Face とは、自然言語処理（NLP）モデルの利用だけでなく共有も行うことができるサービスで、以下のような特徴があります。(参考[1])

- **学習済みモデルの利用**：Hugging Face のライブラリを使用することで、簡単に学習済みモデルを活用できます。これにより、新たなモデルをゼロから訓練する手間を省くことができます。
- **モデルとデータセットの共有**：Hugging Face Hub は、モデルやデータセットを共有し、探索するためのプラットフォームです。他の人々とモデルを共有することができるので、他の人が構築した強力なモデルをダウンロードできます。
- **簡単に利用できること**:
  - Hugging Face は、API を通じて簡単に利用できます。ユーザーは Python などで書いたプログラムを使用して、モデルの評価、テキスト生成、質問応答などのタスクを実行できます。
- **多様なツール**：Hugging Face は、Spaces、API、推論ポイント、Transformers ライブラリなど、多くのツールを提供しています。また、自然言語処理だけでなく、画像や音声の処理に適したライブラリも提供しています。例えば、以下のライブラリがあります:
  - **Transformers**: 自然言語処理モデルを簡単に利用できるライブラリ。テキスト分類、情報抽出、質問応答、テキスト生成などのタスクに適しています。
  - **Tokenizers**: トークン化に用いられるライブラリ。文章をトークンに分割するプロセスを効率的に実行することができます。
  - **Datasets**: 大規模なデータセットの処理と操作を効率的に行うためのツール。一般的な NLP データセットからカスタムデータセットまで扱えます。
-

# 自然言語の生成モデルを試してみます

GPT1 の日本語モデル「rinna/japanese-gpt-1b」を使用して文章生成を試してみます。実行環境は Google 社が提供している「Google Colaboratory」を使用します。必要なモジュールは Colaboratory にデフォルトでインポート済みのため、import の記述なしで実行できます。(参考[2])

```
import torch
from transformers import T5Tokenizer, AutoModelForCausalLM

# japanese-gpt-1bモデルをロード
tokenizer = T5Tokenizer.from_pretrained("rinna/japanese-gpt-1b")
model = AutoModelForCausalLM.from_pretrained("rinna/japanese-gpt-1b")

# 質問応答の関数
def ask_question(question):
    prompt = f"ユーザー: {question}<NL>システム: "
    token_ids = tokenizer.encode(prompt, add_special_tokens=False,
return_tensors="pt")
    with torch.no_grad():
        output_ids = model.generate(
            token_ids.to(model.device),
            max_length=200,
            min_length=100,
            do_sample=True,
            top_k=500,
            top_p=0.95,
            pad_token_id=tokenizer.pad_token_id,
            bos_token_id=tokenizer.bos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    output = tokenizer.decode(output_ids.tolist()[0])
    return output

user_question = "墾田永年私財法とは、"
response = ask_question(user_question)
print(response)
```

上のプログラムを実行すると、以下が出力されます。

```
ユーザー: 墾田永年私財法とは、<NL>システム:新紀元社の中にあった公地の公有化プログラムである。江戸時代の地主たちは、租税の代わりに、例えば京都御所の南に位置し、三十三間堂や清涼殿に近いなどの土地の利用権を幕府から与えられ、さらに一定の賃料を幕府から受け取っていた。この土地を里と称しており、里ごとにも地租が課されていた。いわゆる“郷原(ごうばら)”である。この“郷原”の上に寺社などの公地が成立していった。江戸幕府が田地の寄進を認めた者は幕府から免許を与えられていた。これが“郷原”の上に成立した公地である。
```

実行環境に 2.66G の学習済みモデルが読み込まれます。モデルに prompt と文章を与えると、学習済みモデルから文章が生成されます。応答結果から、文章が生成されていることが確認できます。シードは固定していないため、実行のたびに実行結果が変わります。GPT1 のため多少不自然な文章が生成されますが、日本語モデルの GPT3「rinna/japanese-gpt-neox-3.6b」を使用することもでき、「rinna/japanese-gpt-1b」モデルを使用する場合より自然な日本語になります。

## 画像生成モデルを試してみます

画像生成モデルとして、Stable diffusion の日本をイメージした画像を生成するモデル「aipicasso/cool-japan-diffusion-2-1-1-1」を使用して画像生成を試してみます。今回は画像生成を行うため計算リソースが必要になります。そのため、推論に GPU を使用します。Google Colaboratory には無料で使用できる GPU 枠があります。Colaboratory から「ランタイムのタイプの変更」を選択し、ハードウェアアクセラレータから「T4 GPU」を選択することで GPU を使用できます。(参考[3])

まず、必要なモジュールのインストールを行います。Colaboratory のセルに下記を入力します。

```
!pip install --upgrade git+https://github.com/huggingface/diffusers.git transformers accelerate scipy
```

最新パッケージのインストールが完了したら、以下のコードを実行して、画像生成を行います。

```
 from diffusers import StableDiffusionPipeline,
EulerAncestralDiscreteScheduler
 import torch

 model_id = "aipicasso/cool-japan-diffusion-2-1-1-1"

 scheduler = EulerAncestralDiscreteScheduler.from_pretrained(model_id,
subfolder="scheduler")
 pipe = StableDiffusionPipeline.from_pretrained(model_id,
scheduler=scheduler, torch_dtype=torch.float16)
 pipe = pipe.to("cuda")

 prompt = "Nature,landscape,mountain,autumn,sunrise,crow,masterpiece,best
quality,ultra detailed"
 negative_prompt="worst quality,ugly,bad anatomy,jpeg artifacts,worst
quality,out of focus,JPEG artifacts,low resolution,error"
 images = pipe(prompt,negative_prompt=negative_prompt,
num_inference_steps=20).images
 images[0]
```

上のプログラムを実行すると、学習済みモデルから画像が生成されていることが確認できます。prompt には生成したい画像に関する単語をカンマ区切りで記入し、negative_prompt には除外したい条件を記入しています。実行のたびに生成される画像が変化しますが、山や秋に関する風景画像が生成されると思います。

## おわりに

本記事では、AI モデルを Hugging Face の Transformers を使用する試みとして、GPT1 の日本語モデルを使用した文章生成と、画像生成モデルの Stable diffusion を使用して、Google Colaboratory 環境で簡単に実行できることを確認しました 。AI モデルを試すファーストステップとして、学習済みモデルを使用してローカル環境で実行してみることが非常に参考になるかと思います。ぜひ、Hugging Face にアクセスして、その他のモデル等を試してみてください。注意点として、ローカル環境で自身の学習以外の目的で使用する場合には、使用するモデルごとにライセンスを確認してください。

# 参考文献

[1] "Hugging Face" https://huggingface.co/ (参照: 2024/04/03)

[2] "japanese-gpt-1b" https://huggingface.co/rinna/japanese-gpt-1b (参照: 2024/04/03)

[3] "Cool Japan Diffusion 2.1.1.1 Model Card" https://huggingface.co/aipicasso/cool-japan-diffusion-2-1-1-1(参照: 2024/04/03)

**GSLetterNeo** Vol.189
2024 年 4 月 20 日発行
発行者　株式会社 SRA 技術本部 先端技術研究室

編集者　熊澤努　方 学芬
バックナンバー https://www.sra.co.jp/public/sra/gsletter/
お問い合わせ　gsneo@sra.co.jp

株式会社SRA
〒171-8513 東京都豊島区南池袋 2-32-8


夢を。Yawaraka Innovation
やわらかいのべーしょん