

テキスト読み上げ実験…

①プリセット音声 ②自分の声

佐藤 聖

リスト1 設定ファイル

```
[TTS1]
# VALL-E-Xフォルダのパス
vallex_path = ../VALL-E-X
# 読み上げるテキスト
text_prompt = """インターフェース誌は毎月25日発売です。"""
# サンプリング周波数(Hz)
samplerate = 44100
# 保存ファイル
save_file = ../user/vallex_generation.wav

[TTS2]
# VALL-E-Xフォルダのパス
vallex_path = ../VALL-E-X
# サンプル名(npzファイルのファイル名)
model_name = test
# 音声プロンプト(声のコピー元)
sample_voice = ../user/sample.wav
# 音声プロンプトのセリフ
sample_transcript = おかけになった番号は、
                    電波が届かない場所にあるか、電源が入っていません
text_prompt = インターフェース誌は毎月25日発売です。
# 保存ファイル
save_file = ../user/vallex_generation.wav
```

前章でVALL-E-Xを使った音声生成の設定を行いました。本章ではVALL-E-XをAPI経由で操作するPythonプログラムを作成して、音声生成を試みます。APIを使うと、外部のプログラムやアプリケーションからVALL-E-Xを操作できます。この仕組みを利用すると少ない操作で済みますので、頻繁に音声を作成するときに向いているでしょう。

実行する音声生成

● 実験①…プリセット音声でテキストを読み上げる

VALL-E-Xには英語、中国語、日本語の音声ファイルがVALL-E-X/promptsフォルダに用意されています。これを使うことで、その声を使って音声を生成します。

● 実験②…声をコピーしてテキストを読み上げる

皆さんが音声を録音して作成したwavファイルから声をコピーして音声生成します。

準備

● Pythonプログラムから読み込む設定ファイルを作る

PythonプログラムでVALL-E-XのWebUIに備わっているAPIを操作します。そこで必要となる設定内容は、settings.ini(リスト1)という設定ファイルに記録しています。このようにすると、例えば、クラウド上の仮想サーバで実行する場合に、設定変更時にPythonプログラムの書き換えが不要になり、settings.iniを修正するだけで済みます。

▶ 実験①の設定

リスト1の[TTS1]セクションから読み取ります。save_fileでは、wavファイルとその格納先を相対パスで設定しています。text_promptに渡すテキストは、前後に[""]を付けます。samplerateは、VALL-E-Xのプリセット音声のサンプリング周波数ですので変更せずに使用します。

▶ 実験②の設定

リスト1の[TTS2]セクションから読み取ります。sample_voiceとsave_fileも相対パスです。この部分は使用するPCやサーバのディレクトリ構造に合わせて柔軟に設定できるようにしました。ここではsample_transcriptとtext_promptに渡すテキストは、そのまま設定します。また、音声プロンプトはsample_voiceにwavファイルを設定しますが、サンプリング周波数を設定する必要はありません。

● 設定ファイルの置き場所

仮想環境(env-va)の中は、仮想環境を作成した際のPython関連のフォルダを除き、図1に示す構成とします。

VALL-E-XとFFmpegのフォルダ、作成したPythonプログラムや設定ファイル(settings.ini)などを保存するuserフォルダに分けています。自分が作成するファイルは、すべてuserフォル