

Google 雲端語音轉文字

我最近嘗試了 Google Cloud 的語音轉文字 API。以下是我用來執行轉錄的 Python 函數。

```
import os
import json
import time
import argparse
from google.cloud import speech
from pydub import AudioSegment
import tempfile

# 固定輸出目錄
OUTPUT_DIRECTORY = "assets/transcriptions"

def speech_to_text(audio_file, output_filename):
    print(f" 正在生成轉錄：{output_filename}")
    try:
        client = speech.SpeechClient()

        # 使用 pydub 加載音頻文件以確定參數
        audio_segment = AudioSegment.from_file(audio_file)
        sample_rate = audio_segment.frame_rate
        channels = audio_segment.channels

        # 根據文件擴展名確定編碼
        file_extension = os.path.splitext(audio_file)[1].lower()
        if file_extension == '.mp3':
            encoding = speech.RecognitionConfig.AudioEncoding.MP3
        elif file_extension in ['.wav', '.wave']:
            encoding = speech.RecognitionConfig.AudioEncoding.LINEAR16
        elif file_extension == '.flac':
            encoding = speech.RecognitionConfig.AudioEncoding.FLAC
        else:
            print(f" 不支持的文件格式：{file_extension}")
            return

        # 配置識別
        config = speech.RecognitionConfig(
```

```

        encoding=encoding,
        sample_rate_hertz=sample_rate,
        audio_channel_count=channels,
        language_code="en-US", # 根據你的邏輯設置
    )

    with open(audio_file, "rb") as f:
        audio_content = f.read()

    audio = speech.RecognitionAudio(content=audio_content)

    # 執行長時間運行的語音識別
    try:
        operation = client.long_running_recognize(config=config, audio=audio)
        response = operation.result(timeout=300) # 根據需要調整超時時間
    except Exception as e:
        print(f" 轉錄過程中出錯：{e}")
        return

    print(response.results)

    transcription = ""
    for result in response.results:
        transcription += result.alternatives[0].transcript + "\n"

    with open(output_filename, "w", encoding="utf-8") as f:
        f.write(transcription)
    print(f" 轉錄已寫入 {output_filename}")

except Exception as e:
    print(f" 在生成 {output_filename} 的轉錄時出錯：{e}")

def process_audio_files(input_dir, output_dir):
    os.makedirs(output_dir, exist_ok=True)

    all_audio_files = [f for f in os.listdir(input_dir) if f.endswith((''.mp3', '.wav', '.m4a'))]
    total_files = len(all_audio_files)
    print(f" 需要處理的音頻文件總數：{total_files}")

```

```

if total_files == 0:
    print(f" 在 '{input_dir}' 目錄中未找到音頻文件。")
    return

files_processed = 0

for filename in all_audio_files:
    audio_file_path = os.path.join(input_dir, filename)
    output_filename = os.path.join(output_dir, f"{os.path.splitext(filename)[0]}.txt")
    if os.path.exists(output_filename):
        print(f" 跳過 {filename} : {output_filename} 已存在。")
        continue
    print(f"\n正在處理 {files_processed + 1}/{total_files} : {filename}")
    try:
        # 根據文件名後綴確定語言
        if filename.endswith('-zh.mp3') or filename.endswith('-zh.wav') or filename.endswith('-zh.m4a'):
            language_code = "cmn-CN"
        else:
            language_code = "en-US"

        # 如果需要，可以在 speech_to_text 中更新配置
        # 為了簡化，我們將在 speech_to_text 中設置 language_code

        speech_to_text(
            audio_file=audio_file_path,
            output_filename=output_filename,
        )
        files_processed += 1
        print(f" 文件 {files_processed}/{total_files} 已處理。")
    except Exception as e:
        print(f" 處理 {filename} 失敗：{e}")
        continue

print(f" 處理完成！已處理 {files_processed}/{total_files} 個文件。")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description=" 處理音頻文件以生成轉錄。")
    parser.add_argument('--input_dir', type=str, default="assets/audios", help=" 音頻文件的輸入目錄。")

```

```
args = parser.parse_args()
```

```
process_audio_files(  
    input_dir=args.input_dir,  
    output_dir=OUTPUT_DIRECTORY,  
)
```