Personalized offline DVC

Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

Outline

會議記錄

討論內容

- 最後實作出的樣子是甚麼? 要怎麼 demo？先把規格定義好，scenario 是甚麼？要做到甚麼程度
- 老師理想中的環境是在手機 demo，這方面需要有手機及 server 的程式要處理，有很多部分需要規劃

老師指示

- 先規劃一版理想中的 demo scenario，再去討論實際的實作細節

個人化離線語音裝置

- Creation purpose
  1. 實現一手持裝置可以不管在何時何地將 A(輸入) 聲音轉換成 B(目標) 聲音
- Implementation flow
  1. 在 PC 上完成功能流程
  2. 做一個手持裝置 on line
  3. 將第二部的輸入輸出傳上雲端並等待運算完下載 model
  4. Off line 時，第三步的 model 可以發揮作用
- Demo scenario
  1. 操作裝置 (擇一)
     A. 電腦 (使用 Python)
     B. 手機 (使用 Java)
  2. 產品特點
     A. 相較於傳統方法，它可以利用更少的儲存空間做到更好的轉換效果
     B. 隨時隨地都可以轉換，不再受限於有網路的狀態下
  3. 展示流程
     A. 使用我的聲音做 Demo
     B. 操作者持裝置在有網路狀態下，對裝置說話並轉換
     C. 使裝置進入無網路環境 (開飛航模式)，對裝置說話並轉換

程式架構 (裝置端)

1. 麥克風輸入音檔
2. 是否連上網際網路

if (能連網) {

3. 將輸入的音檔上傳至雲端虛擬機
4. Speech-to-Text
5. Text-to-Speech
6. 將轉換後的音檔上傳至雲端虛擬機
if(觸發條件)
7. 下載 DNN 模型參數
8. 寫 Python 或是 Java 版 DVC

} else if (不能連網){

9. 使用儲存的 DNN 模型執行儲存的參數

}

10. 播放音檔

- 初次使用，請在有網路的狀況下至少說 200 句話再使用離線功能
- 觸發條件 (按下 button 或是更新下載)
  - 模型參數手動操控虛擬機下載or 自動下載
  - 轉換/下載期間跳出預估等待時間

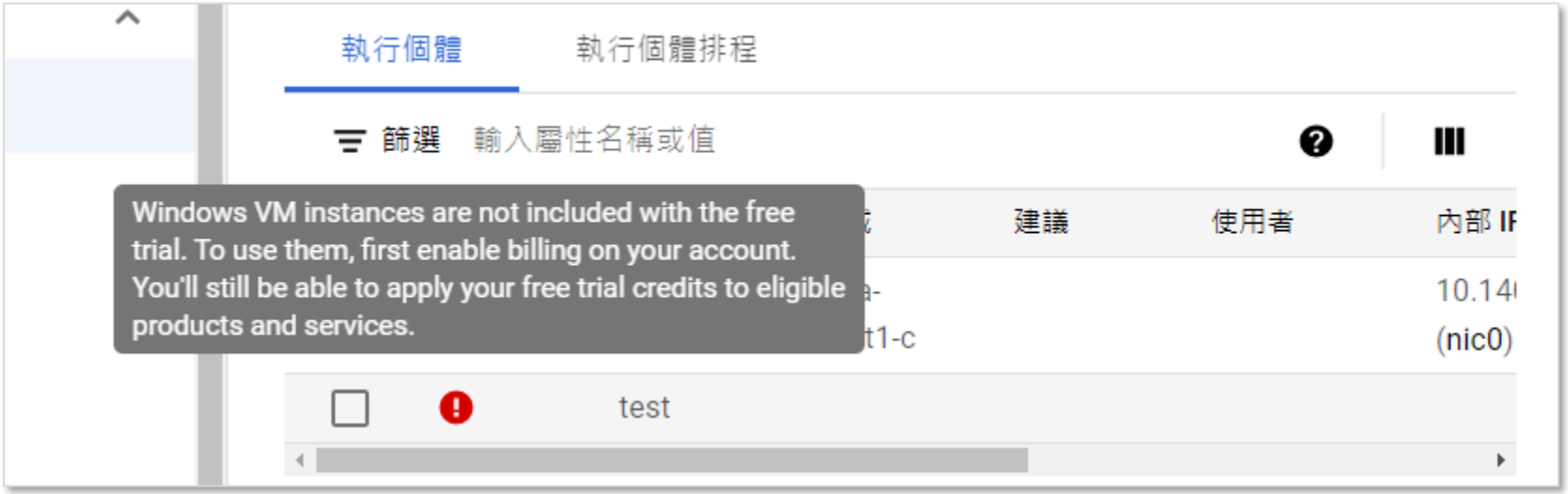
程式架構 (雲端)

1. 開 Google 虛擬機
2. 架設 YMDVC 執行環境
3. 利用裝置端錄製的聲音 & 轉換後的聲音自動進行 DTW 對齊
4. 將對齊後聲音進行自動訓練 (一個禮拜或是一個月一次)
5. 將取得的模型自動做資料整理 (模型轉換 pre/post norm, DNN model)
6. 將轉換完模型放進資料夾內方便裝置端存取

附錄

GCP VM 無法使用 Windows

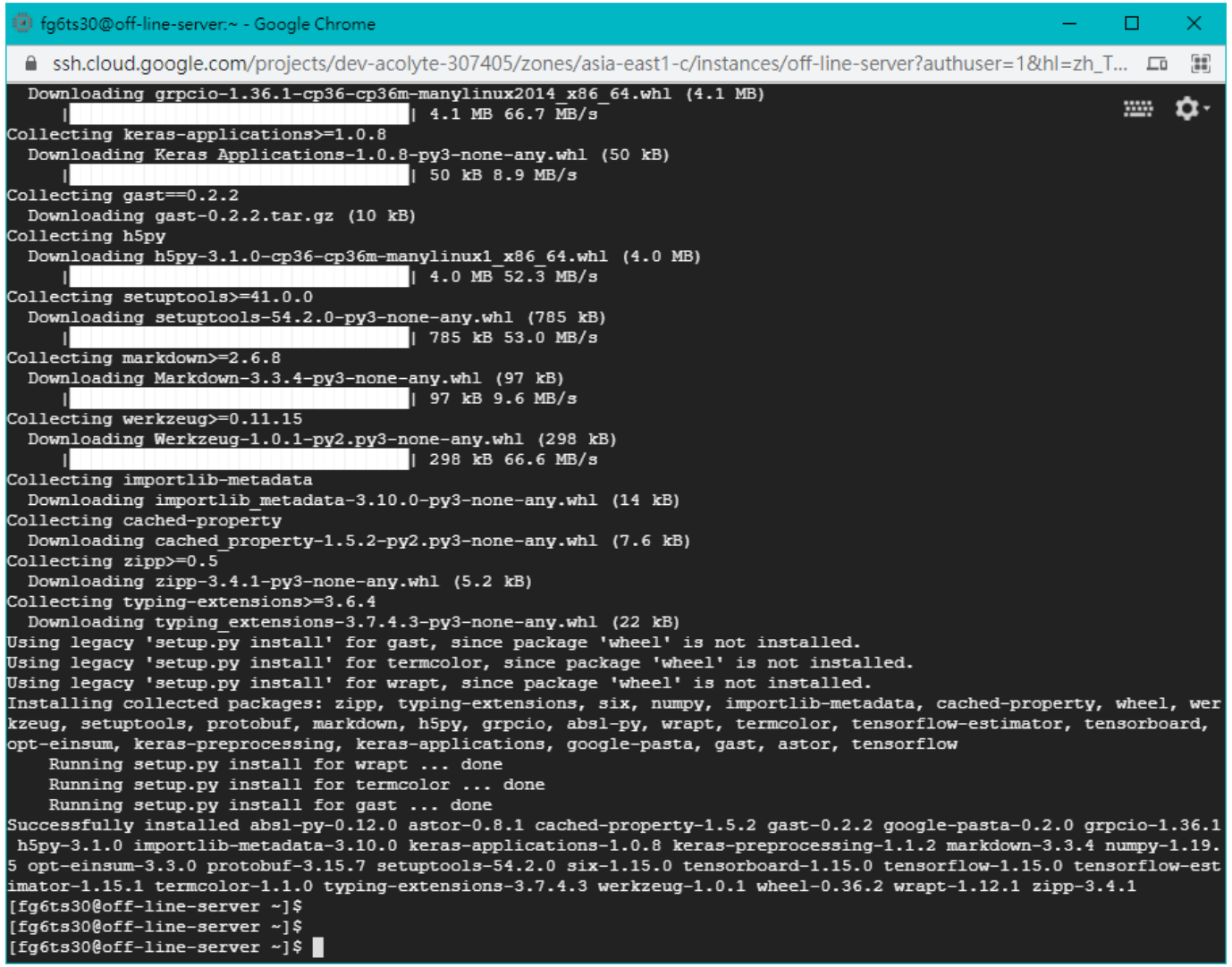
Compute Engine (VM 執行個體)



Google Cloud Free計劃

計劃覆蓋	您的免費試用額度適用於所有Google Cloud資源，包括 Google Maps Platform的使用情況，但以下情況除外： • 您無法將GPU添加到您的VM實例。 • 您不能請求增加配額。有關Compute Engine配額的概述，請參閱 資源配額。 • 您無法創建基於Windows Server 映像的VM實例 。 要執行上面列表中的任何操作，您必須 升級到付費Cloud Billing帳戶。

GCP VM



目前建置了 CentOS 7，並使其可以使用 pip install 套件

參考文獻

Google Cloud 建立服務帳戶：
https://support.google.com/cloudidentity/answer/7378726?hl=zh-Hant

第一次開Google VM就上手 – Compute Engine操作簡介：
https://blog.cloud-ace.tw/compute-engine/start-a-google-vm-compute-engine/

使用其他 SSH 用戶端登入到 Google Compute Engine 的 Linux VM：
https://dotblogs.com.tw/supershowwei/2017/08/23/120313

在 CentOS 7 上安裝 Python3
https://kirin.idv.tw/python-install-python3-in-centos7/

程式架構(1/3) (使用 **Python** 製作雛形)

麥克風輸入音檔 // speech_recognition.microphone()，函式可使麥克風輸入

是否連上網際網路 // 使用系統指令 “os.system()” 函數 ping 網域，來測試是否有網路，回傳 0 or 1

if (能連網) {

upload

Speech-to-Text // SpeechRecognition 套件

Text-to-Speech // &

儲存音檔 // gTTS 套件

下載 DNN 模型參數 // 使用 paramiko 套件使用 ssh 連線遠端 VM 複製檔案

寫 python 版 DVC1.0

} else if (不能連網){

使用儲存的 DNN 1.0 模型執行儲存的參數

// 寫 DVC 1.0 python 版本，需處理傳回來的參數

}

播放音檔 // 使用 pygame 套件 mixer() 函式播放音檔

程式架構(2/3) (使用 **Java** 實作手機 APP)

```
麥克風輸入音檔              // Android 提供 MediaRecorder 達成手機收音、儲存、播音
是否連上網際網路            // 加入權限，再使用內建提供函數判斷是否連網
if ( 能連網 ) {
        Speech-to-Text
        Text-to-Speech      // google 有提供 libraries
        儲存音檔
        下載 DNN 模型       // 利用 ssh 遠端連線到 Google VM 複製檔案
} else if (不能連網){
        使用儲存的 DNN 1.0 模型執行儲存的參數
}                           // 寫 DVC 1.0 java 版本，需處理傳回來的參數
播放音檔
```