Cyberon DSpotter
語音模型的建立與優化

Version: 1.2

Date of issue: 2020/8/25

Cyberon Corporation

Software solution provider for embedded system

http://www.cyberon.com.tw/

內容目錄

修改記錄：

版本	日期	更新內容
1.0	2020-8-4	1. 第一次撰寫。
1.1	2020-8-5	1. 修正 2.5 第 5 點。 2. 對調 3.3 第 5、6 點。
1.2	2020-8-25	1. 修改 1.1 的說明。 2. 修正 1.2 第 5 點關於 RAM 和 model size 的使用量。 3. 在 3.3 第 2、8 點中，增加一些說明。 4. 在 4.1 第 2 點中，增加連音處理的說明與範例。 5. 在 4.2.第 4 點中，增加 garbage command 的範例。

1. Cyberon DSpotter

1.1. 簡介

賽微 DSpotter 是用於語音喚醒的識別引擎，也可以拿來辨識預先定義好的指令，具有以下特色：

1. 使用 DNN 的技術，具有良好的抗噪能力。
2. 計算資源需求不高，全部使用 fixed point 運算，適合在 MCU 上運行，並在 ARM CM4、RISC-V with P extension 平台使用其 SIMD DSP 指令做過最佳化，並且增加一個辨識指令只會多用一點資源，詳細規格請參考 1.2。
3. 每個語言都有兩種大小的語音模型可選，level 0 的語音模型比 level 1 較小、使用較少的資源，但誤觸發高一點、噪音環境的效果差一點 (SNR 10、5 dB 時，約差 3、8%)，開發商可依系統資源彈性選擇。
4. 提供 Windows 工具程式 DSMT，只需輸入文字就可快速建立辨識模型，目前支援 30 種語言。
5. 需 10 男 10 女的指令錄音檔(至少 5 男 5 女，每人每個指令唸 5~10 次)，就可自行用 DSMT 做參數的優化調整和測試驗證(包括噪音測試和誤觸發測試)。
6. 如果有 50 男 50 女的指令錄音檔(多多益善，至少 30 男 30 女，每人每個指令唸 10 次)，我們就可以用自有的運算雲重新訓練、調適語音模型，可讓口音相容性更好、提高各種環境的辨識率、使誤觸發降得更低，可得到效果比 level 1 好、但使用資源和 level 0 差不多的語音模型。

1.2. 規格

DSpotter 的詳細規格如下：

1. CPU (最佳化選項都使用 O3)
   以 ARM CM3 為例，約需 60(level 1)、45(level 0) MCPS。
   ARM CM4 經使用 SIMD DSP 後為 38(level 1)、30(level 0) MCPS。
   RISC-V with P extension 經使用 SIMD DSP 後為 34(level 1)、26(level 0) MCPS。
2. RAM
   約需 40 KB(level 1)、37 KB(level 0)。
3. Code size
   約 40 KB(CM4)，不同的 IC 平台或是編譯選項會有些差異。
4. Model size
   約需 155KB(level 1)、91KB(level 0)，不同語言會略有不同。
5. 以中文四字詞為例，增加一個指令約增加
   Level 1：0.064 MCPS(CM4)，RAM 128 bytes，model 32 bytes。
   Level 0：0.05 MCPS(CM4)，RAM 128 bytes，model 32 bytes。

2. Cyberon DSpotter Modeling Tool

2.1. 簡介

Cyberon DSpotter Modeling Tool 簡稱 DSMT，是用於建立、測試、優化調整語音模型的工具，有以下功能：

1. 需帳號、連網才能使用，支援 Windows 平台。
2. 輸入指令文字後會傳到雲端伺服器，伺服器建立好辨識模型就回傳到 DSMT，可馬上做測試驗證，目前支援 30 種語言。
3. 伺服器內建字典，會從輸入文字查出發音及其多重發音，DSMT 可連到雲端 TTS(Text To Speech)將發音播放出來，如果發音不對，使用者可自行修改，亦可自行新增多重發音。
4. 具有 online 測試工具，可從 PC 麥克風錄音，建完語音模型可馬上自行做測試，測完後可以將錄音存檔，用來做 offline 測試的語料。
5. 具有 offline 測試工具，讀 wave/script 檔來辨識，並可讓使用者混和各種類型的噪音並設定訊噪比(Signal Noise Ratial, SNR)來測試，辨識完會統計辨識率、誤辨識(FA)次數，並提供辨識出來的分數、能量、時間點，作為後續調整優化的依據，可以用來做辨識率測試、誤觸發測試。

DSMT 內附詳細的使用手冊，請參考使用手冊來操作這些功能。

2.2. 噪音測試的準備

DSMT 內建 car, babble, office 等三種噪音測試音檔，也可以使用外部輸入的音檔，例如：

1. Babble noise
   可以從 https://github.com/microsoft/MS-SNSD/tree/master/noise_test 下載 Babble_X.wav。
2. Pink/White/Brown noise
   可以用 Audacity 產生 120 秒的 noise wave 檔。
   Audacity 是免費、開源的音訊編輯軟體(https://www.audacityteam.org/)。
3. Music noise
   Alexa Voice Service(AVS)是用 Happy - Pharrell Williams HD 這首歌來測試。

2.3. 誤觸發測試的準備

每個語言都需準備其各自的誤觸發測試語料，可以錄電視(或是 Youtube)上新聞、綜藝、

劇集等節目的聲音，總計 24 小時。

Amazon AVS 有提供英、德、法、義、西、日的誤觸發測試語料：
https://developer.amazon.com/alexa/console/avs/preview/resources/details/AVS%20acoustic%20audio%20files
是一個 24 小時的 mp3，請先轉成 12 個 wav 檔，並準備 script(*.spt)檔。

誤觸發的 script 檔案編碼需用 unicode(UTF16LE)，每一行的格式如下：
full_path_file_name \r\n，例如 D:\WAVE_OOV\CHT1.wav

2.4. 測試標準

辨識結果有以下幾種狀況：

1. 辨識成功：成功次數除以測試次數就可得到辨識率。
2. 誤拒絕(FR, False Reject)：有說指令但沒有辨識出任何結果。
3. 誤辨識(FA, False Acept)：
   A. 辨識出的指令和說的不一樣，例如"我要去一樓" => "我要去七樓"。
   B. 沒有說指令但因其他聲音辨識出結果(誤觸發 False Trigger or False Alarm)，做誤觸發測試時，也會稱做 OOV(Out Of Vocabulary)測試。

客戶可依照需求訂定測試驗收的辨識率、誤觸發標準，以下提供 Amazon 替 Alexa 訂的各種標準作為參考：
https://developer.amazon.com/alexa/console/avs/preview/resources/details/AVS%20Acoustic%20Self-test%20Checklist

1. 辨識率
   安靜環境：90%
   穩態(stationary)噪音(如 pink noise)環境：80%
   音樂噪音環境：80%
   當測試設備播放音樂(播放的音樂檔 AVS 也有定義)：67%
2. 誤觸發
   24 小時不超過 3 次

對於近、中距離(1.5 公尺內)使用的裝置，如耳機、智慧手錶、手環，DSpotter 應可通過這項標準，對於需要遠距離使用的裝置，如智慧音箱，由於噪音測試的 SNR 會低於 5 dB，甚至實際使用的環境可能會有嚴重的迴響，則一定需要麥克風陣列、訊號前處理系統才有可能通過測試(Amazon Echo 採用 7 顆麥克風陣列)。

2.5. 優化調整的參數

在 DSMT 中，可優化調整的參數介紹如下：

1. DSpotter 收到語音命令後會計算出 SG Difference 的分數、以及和每個指令辨識相像程度的 Confidence Score ，這兩個分數必須符合檢測條件，指令才會被辨識出來。
2. Confidence Score(CS)：
   代表語音和辨識指令相像的程度，分數越高代表越像、越好。
3. SG Difference(SG)：
   代表語音和 Silence、Garbage 不像的程度，分數越高代表差距越大、越不像、越好。Garbage 代表雜音、環境噪音。
4. 檢測的條件為：
   CS >= 10 && SG >= 0 或 10 > CS >= 0 && SG >= 10
5. 調整每個指令的這兩項指標的加權分數(reward)，就會改變辨識率、誤觸發次數，降低加權分數(reward 為負值)會使誤觸發次數變低但辨識率變差，提高加權分數(reward 為正值)會使則辨識率變好但誤觸發次數會變高，優化調整的目的就是在辨識率和誤觸發次數之間取得適當的平衡點。

3. 指令與裝置的準備工作

3.1. 辨識指令的設計

對於 24 小時一直都辨識的指令，稱作喚醒詞，為了提高喚醒詞的辨識率並降低誤觸發次數，有以下注意事項：

1. 至少 3 個音節，最好有 4 個音節，音節多則誤觸發天生就比較低，較易調整優化、並在調整過程中比較不會犧牲辨識率。
2. 結尾發音最好為母音，例如 Alexa、Hey Siri、Ok Google 都為母音結尾，Book、News 為子音結尾。
3. 選用發音能量強的字
   例如"我要去五樓"在安靜環境的辨識率都和其他的"我要去一、二、三、四樓"差不多但在 SNR 5 dB 的噪音測試則明顯低了約 20~30%，以英文來說，Hey Siri 的 Hey， 一開始的發音能量很強，在噪音環境比較不容易被掩蓋，所以辨識率表現比 Alexa、Ok Google 來的好。
4. 不要同時辨識太多個喚醒詞，喚醒詞越多誤觸發的機率越高，對於需要同時辨識很多個指令的需求，請設計成一個喚醒詞 + 多個命令詞的操作方式。
5. 如果會同時辨識多個指令，指令之間的發音差異要夠大(兩個音節以上)，才不容易因口音、環境噪音、裝置錄音品質的影響而辨識錯誤。
6. 每個音節最好都包含子音、母音，例如"小易小易"的"易"沒有子音，尤其不要有連續只有母音的字，因為說話快一點的時候容易和前面的音連起來，發音變得不明顯音程變短、甚至變音，辨識率就會變差。
7. 為了降低誤觸發的機率，喚醒詞請使用日常說話不易出現的詞句。
8. 可多準備幾組候選詞，做完調整優化後再來選擇效果最好的指令。

PS. "子音、母音"亦稱"輔音、元音"
https://www.tutorabc.com.cn/About/NewsDetail/7114.html
或稱"聲母、韻母"
http://www.baike.com/wiki/%E5%A3%B0%E6%AF%8D%E9%9F%B5%E6%AF%8D

3.2. 辨識率測試音檔

1. 用終端裝置錄音最為理想，如果前期來不及準備則可先用 iPhone 錄音，後期則需用終端裝置錄音再測試確認。
2. 10 男 10 女，甚至更多，至少 5 男 5 女
   理想上應該依據目標客群，涵蓋男女、各種口音、年齡的聲音。

3. 請在安靜、迴響低的環境內(如小會議室)錄音，錄音格式為單聲道、16 bits、16 KHz，請存成無壓縮的 wav 檔案。。
4. 每人、每種使用距離(Ex. 近距離、1M 、3M) 、每個指令唸 5~10 句存成一個音檔、每句間隔 2 秒，唸完最後一句時也等 2 秒再停止錄音。
5. 如果是用 iPhone 錄音，須將.m4a 格式轉成單聲道、16 bits、16 KHz 的 wav 檔。
6. 使用 audio editor 播放、檢查、編輯(補 ending silence)所有音檔，並撰寫 DSMT 測試用的 script file(*.spt)。
7. spt 檔的編碼需用 unicode(UTF16LE)，每一行的格式為：
   full_path_file_name\tCmd1Text/Cmd2Text/... \r\n
   例如：D:\M1\Command1.wav　智能管家/智能管家/智能管家/智能管家/智能管家

3.3. 裝置的錄音品質

錄音品質對辨識率有很大的影響，以下是語音辨識對錄音品質的需求：

1. 格式：16 KHz、16 bits(-32768~32767)、單聲道、無壓縮
2. 可能要關閉 Auto Gain Control(AGC)、Noise Reduction(NR)
   一般來說，除非 AGC、NR 有針對語音辨識做特殊調整，否則反而會因為破壞聲音頻譜而降低辨識率。
   PS. 適合做語音辨識的 AGC 不會把 gain 調太快，適合做語音辨識的 NR 不會破壞聲紋。
3. 訊噪比：
   安靜環境、在裝置所規範的最遠使用距離以 70dB 的音量說話，訊噪比可以超過 20 dB，至少要 15dB，如果無法達到則需考慮使用適合做語音辨識的前處理系統，例如麥克風陣列。
   PS. 說話音量是人距離分貝計一公尺說話，分貝計所量測的平均能量。
4. 錄音音量(recording gain 的調整)：
   在裝置所規範的最遠使用距離以 64dB 的說話音量，錄音振幅最大峰值需在 1000 以上，最好能有 4000，如果沒有則須用前處理系統。
   在裝置所規範的最近使用距離以 70dB 的說話音量，不可有爆音現象(音量超過-32768~32767、聲音被截斷)，如果有請換 AOP 較高的麥克風。
5. 暫態響應或 VAD 切音：
   VAD 的切音長度，或是剛開始錄音的資料，如果有不穩定的暫態響應、須丟掉的長度，要小於 100 ms，以免辨識率受到太大影響。
6. 穩定性：
   錄一百次，每次 8 秒，檢查是否有一樣的品質，暫態、切音的長度最好也是保持相同長度，以利後續調整處理。
7. 沒有雜訊頻帶：

在安靜環境下錄音的頻譜，沒有特定的雜訊頻帶，如果有通常是電路干擾所產生的。

8. 沒有弱頻帶：

   用人工嘴(需經等化器做頻響校正)或監聽喇叭播放 white noise(全頻帶等強度 noise)做近距離錄音(10 cm)，最好是在無響箱、無響室操作，將錄到的聲音做頻譜分析，沒有特別弱的頻帶，如果有，通常為機構設計問題。

   機構設計注意事項：

   A. 機構開孔要大於 MEMS 麥克風的開孔，最好在 1 mm 以上。

   B. 麥克風開孔需對齊、緊貼機構開孔，避免形成諧振腔體造成聲音失真。

   C. 麥克風四周可用橡膠墊保持氣密、避免噪音從後方傳入。

   D. 產品設計時需確定機構開孔和使用者聲源之間沒有阻隔(例如設計在產品背面)，否則聲波都是經過空間反射傳入麥克風，也會造成較嚴重的失真。

9. 100~8000Hz 頻率響應曲線越接近水平曲線越好、總諧波失真越小越好。

4. 語音模型的調優

4.1. 先用多重發音解決口音或連音問題

1. 不調 CS reward、SG reward，用 DSMT offline 測試工具統計安靜環境辨識率。
2. 檢查辨識不出來(FR)、錯誤(FA)的音檔：
   如果在某一個音檔剛開始的位置有辨識結果，代表前一句的 ending silence 不夠長，請用 audio editor 加長。
   如果是口音或連音問題導致無法辨識則增加多重發音來解決：
   例如北方人最後一個字發音結尾可能會有ㄛ化音，所以"微信支付"要加"微信支佛"。
   例如台灣比較不會發捲舌音，所以"我要去二樓" => "我要去餓樓"。
   例如"支付寶支付"要加"珠寶支付"來解決快語速的連音問題，但是"上一首"就不能加"上首"，因為會使誤觸發增加太多。
   如果很確定是發音錯誤，則略過、從 spt 中移除。
   PS. 為了相容各地口音、節省計算量，DSpotter 中文沒有辨識聲調(一二三四聲)。
3. 如果 FA 中有很多互相混淆的錯誤辨識，代表發音差異不夠大，盡可能更換指令，因為 DSpotter 為了能在有限資源的 MCU 上執行，簡化了很多運算，聲音的解析度比不過雲端辨識系統，尤其在噪音環境較易混淆發音差異不夠大的指令。

4.2. 優化調整的步驟

1. 基準測試：
   不調 CS reward、SG reward，測試安靜、噪音(請依據使用環境決定噪音種類和強度)的辨識率，以及 24 小時誤觸發次數。
2. 調整 SG reward
   如果誤觸發辨識結果的 SG 都比較低，辨識率測試結果的 SG 都比較高，可以調低 SG reward 來降低誤觸發，定好後再做辨識率測試、誤觸發測試。
3. 依據前次辨識率、誤觸發測試的測試結果，設定每個指令適當的 CS reward，以能降誤觸發、辨識率影響不太多(和基準測試相比)為原則，此步驟可能會反覆做數次。
4. 如果還不能滿足誤觸發條件(通常為不到 4 個音節的指令)、或是導致辨識率降太多，則開始加 garbage command：
   A. 加 partial command，例如"接聽電話" => "接聽"、"電話"
   B. 將其中一個字換掉成發音很不像的、或是換兩個字…
   C. 從誤觸發測試結果(有發生的音檔和位置資訊)去聽音檔，由導致誤觸發的聲音來決定 garbage command，例如 Hey Siri 加了 have very、in serial...。
   加完 garbage command 一定要做安靜、噪音的辨識率測試，誤觸發測試，留下對辨識率影響比較小、降誤觸發比較多的 garbage，此步驟可能需要反覆做很多次。