立達與李老闆討論的幾個聲音專案

1、連續語音辨識，即時顯示文字內容：

- 子問題描述：
  - 第一、訊號是連續不停進入，whisper進行辨識時，先考慮輸入語音訊號是否會有丟失的狀況；
  - 第二、由於新訊號進入與辨識同時進行，怎麼降低辨識延遲。
  - 第三、使用更精簡的辨識系統實作，降低辨識延遲。
- 可能採行的方法
  - 第一點採用方法：
    - 使用兩個buffer，交替做完輸入到模型及暫存尚未處理的訊號。
    - 使用Queue來做為輸入給模型及暫存尚未處理的訊號。
    - 使用輕巧快速的message-queue，例如ZeroMQ。
  - 關於第二點：
    - 採用更快的whisper，例如：faster-whisper。
    - 採用更有效率的資料結構來實作buffer或使用第三方buffer實作。
  - 可做開發參考的程式範例：
    - python_speech_features
    - whisper_streaming

2、開會即時記錄使用whisper產生逐字稿：

- 子問題描述：
  - 第一、會議開會時，語音來源考慮幾種情形：
    - 一次一個人發言，則問題如上述連續語音辨識。
    - 若一次有多人在討論，則同時會有不同語音訊號進入，那麼是辨識其中能量最大的語音，還是要進行語音分離，把每個不同的語音訊號都要辨識。
  - 第二、文字稿的產生中，由於語音辨識仍會有辨識出錯誤字的狀況，因此引進語詞校正後處理是否也納入考慮。
  - 第三、逐字高是否顯示時間點，怎麼對齊文字與時間。
- 可能採行的方法：
  - 使用Whisper等多國語言的預訓練ASR模型做為STT的核心。
  - 若一次只有一人在發言，則作法可參以上連續語音辨識的作法。

- 針對以上多人討論時，若考慮多人同時發出語音訊號，則要使用語音分離模型進行預處理，例如使用pyannote的speech-separation-ami-1.0。分離完後的不同說話人的語音，再進行辨識，這邊也多出了一個問題：不同說話人進行辨識的順序，要再分出一個項目來進行。
- 輸出逐字稿中的文字後校正，假設輸出了一個錯誤詞語的同時，在呈現這個錯誤詞語的同時，記錄位置，同時進行檢查及校正，並把正確結果置換掉錯誤的字詞。這部份包含使用到校正的語言模型、字詞前後文字的比對、並發行程的實作等。
- 對於逐字稿中文字與時間的對齊，可採用whisper-timestamped進行文字與時間戳記的對齊與輸出。

3、不同語言之間的語音翻譯，例：英到中，中到英。像是iToutTranslator。

- 子問題描述
  - 在觀看了iToutTranslator的功能展示後，其最主要的核心在於即時的二種語言間的實時翻譯，其中又分成二類翻譯：文字間的實時翻譯及語音間的實時翻譯。其中二種語音間的實時翻譯，進一步描述，即達成A講英文時的語音訊號輸入的同時，輸出中文語音的語音訊號，而不是A講完一句英文，才輸出一句中文，也就是同步翻譯。而二種語音在進行實時翻譯，也同步把二種語音實時輸出成文字呈現。
- 可能採行的方法
  - 同步輸出轉換的二種文字：
    - 作法大致可如下想法進行：
      - 來源語音放到A空間 --> [語音辨識] --> [來源語音對應的文字] --> [翻譯模型] --> [目標文字]。
        以上為大致的作法，其中的實作及優化細節要依照需求做進一步設計。
  - 同步輸出轉換目標語音：
    - 作法大致可如下想法進行：
      - 作法一：
        - 來源語音放到 --> [語音辨識] --> [來源語音對應的文字] --> [翻譯模型]
          --> [目標文字] --> [TTS] --> 目標語音。
      - 作法二：

        - 來源語音放到 --> [Direct Voice Conversion] --> 目標語音
    - 若結合二個需求 ，則作法一可以同時產生二種文字及目標語音，但經過的程序較多，因此進一步的思考及研究更優的解法，是必須的。

4、TTS+Adapt some specific person's voice, for example, TTS+林志玲聲音：

- 子問題描述
    - 給予一段文字，以特定人的聲音來說出。
- 可採行的方法：
    - Adapt a pre-trained TTS model to a new person voice using transfer-learning or zero-shot learning.可參考幾種實作：
        - Real-Voice Clone https://github.com/CorentinJ/Real-Time-Voice-Cloning
        - Coqui-TTS:採用voice-clone or finetune. https://docs.coqui.ai/en/latest/models/xtts.html
        - Adapting TTS models for new speaker using transfer-learning https://paarthneekhara.github.io/tlfortts

5、Speech 斷句。

- 子問題描述：
    - 經由SST輸出文字後，文字要能正確斷句。
    - 但目前所有的ASR模型，皆是輸入一段語音訊號，然後模型輸出這段語音對應的文字。
    - 承上，若在一段輸出的長文字中，要進行斷字，則要如何處理。
- 可能採行的方法：
    - 從語言模型的角度進行解決，使用transfer-learning，針對目標語言斷句的能力進行finetune。例如，準備沒有斷好的句子與完整斷句的句子，使用Contrastive-Learning進行finetune

Reference:

- CTranslate2 https://github.com/OpenNMT/CTranslate2/
- faster-whisper https://github.com/SYSTRAN/faster-whisper?tab=readme-ov-file
- whisper_streaming https://github.com/ufal/whisper_streaming?tab=readme-ov-file
- whisper-timestamped

https://github.com/linto-ai/whisper-timestamped

- Real-Voice Clone
  https://github.com/CorentinJ/Real-Time-Voice-Cloning
- Coqui-TTS:採用voice-clone or finetune.
  https://docs.coqui.ai/en/latest/models/xtts.html
- Adapting TTS models for new speaker using transfer-learning
  https://paarthneekhara.github.io/tlfortts
- speech-separation-ami-1.0
  https://huggingface.co/pyannote/speech-separation-ami-1.0