

Wang TTS (based on VCC 2020 ref. design): zhenwei (震威) results & TTS training process

W.-C. Huang, T. Hayashi, S. Watanabe, T. Toda, "The sequence-to-sequence baseline for the voice conversion challenge 2020: cascading ASR and TTS," arXiv preprint arXiv:2010.02434, 2020.

Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

Outline

Action item

使用震威的語料進行 TTS 微調並報告 TTS 的訓練流程

Status report

背景概要

- 使用 VCC 2020 reference 的 VC baseline，架構由 2 個 model 組成，分別是 ASR + TTS，而兩個 model 互相獨立，ASR 將文字辨識出來後就可以與整個系統切開來，因此我們著重在 TTS model 的部分 (如圖一)。
- 先前使用不同的語料，有不同的合成品質，因此再使用不一樣的 input 作為輸入，這次使用震威的語料進行實驗。

本周進度

- 介紹 TTS
 1. 使用的 TTS 規格、優點
 2. 訓練前的準備資料 (預訓練模型、音檔、文本)
 3. 微調步驟分為 7 個步驟
- 實驗過程
 1. 使用了震威的語料，進行語句增量的實驗
 2. 總共做了 6 個版本，以訓練語句數量可以分為 60 / 310 / 320 句，共 3 類
 3. 實驗結果
- 結論：先前使用相同的預訓練模型下，使用了 VCC20 一位女性語者與繆詠青共兩位，訓練完合成出來的聲音都沒有什麼雜音，而使用竣尹或是震威的語料生成出來的聲音雖然音色相似，但都會有明顯雜音，因此認為是預訓練模型使用 CSMSC 中文標準女聲資料庫訓練完後的模型，對於高頻的學習能力較好，而男生高頻大部分都在氣音出現。

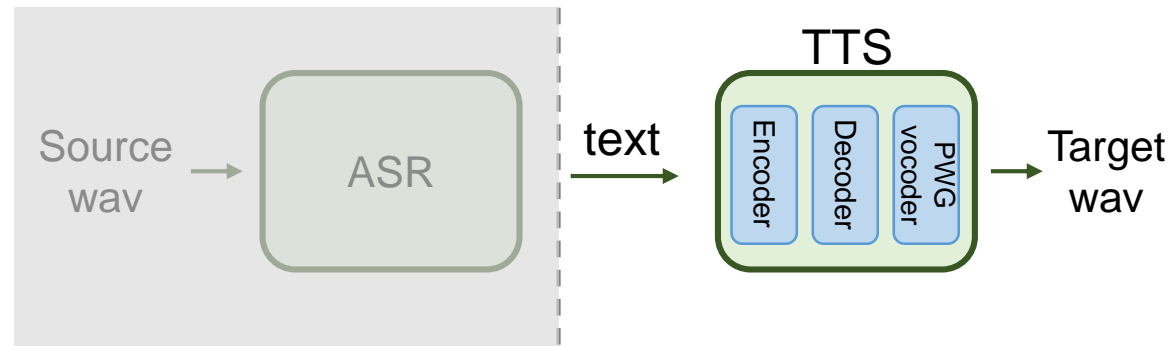


Figure 1: System structure

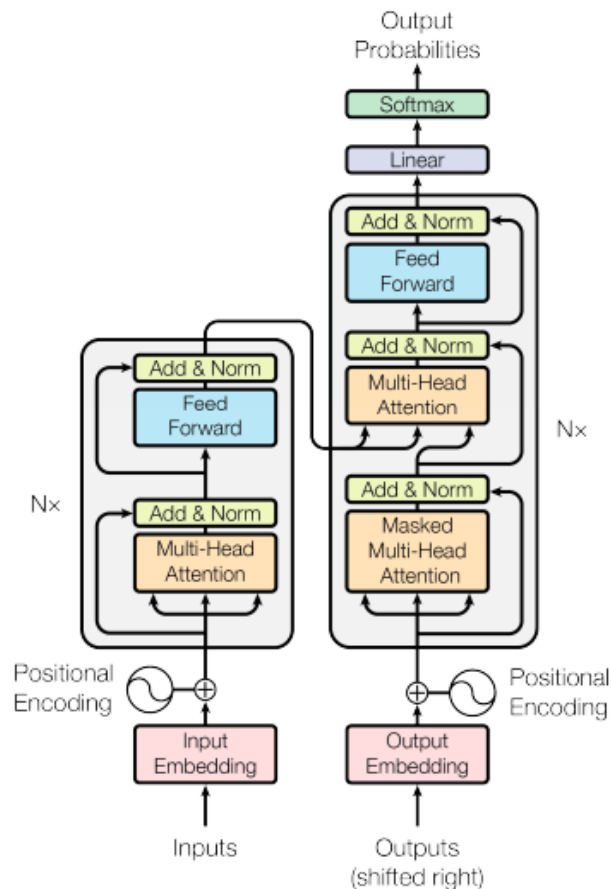
TTS - Transformer

Transformer 是近代語音處理的一個熱門神經網路，在 2017 年由 Google 提出 “**Attention Is All You Need**” 這篇論文 [1] 中出現，它由一組 encoder 與 decoder 組成，架構是基於 Seq2Seq + self-attention。

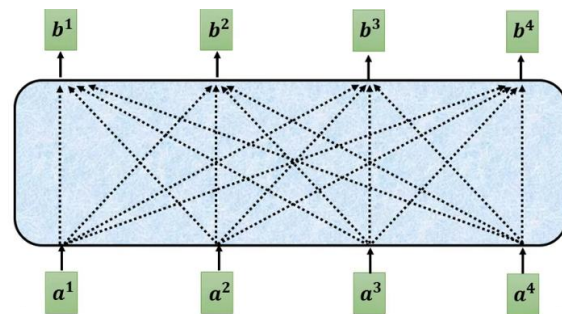
Sequence to sequence model 輸入一個 sequence 就會輸出一個 sequence，而它的輸入、輸出的長度均由神經網路自行學習，兩者長度沒有對應關係。

Self-attention (自注意力機制) 則是 Transformer 主要取代 RNN (LSTM、GRU) 的主要原因，這個機制有兩大優點，一是他可以平行化運算，去掉 RNN 遞迴的結構，二是每一個輸出的向量，都會看過整個輸入的序列，因此當輸入的 sequence 太長的時候，RNN 容易忘記一開始的訊息，但使用 self-attention 的 Transformer 就可以知道要把注意力放在哪些資訊上，而不會因為輸入的資訊太長而忘記一開始的資訊。

所以 Transformer 的優點是除了可以處理更長的序列之外，還減輕了運算效率。



The Transformer - model architecture.



Self-attention (自注意力機制)

[1] A. Vaswani, et al., "Attention Is All You Need", *arXiv preprint arXiv:1706.03762*, 2017.

Pre-trained TTS model

1. VCC2020 Baseline **English pre-training model**: multi-speaker, x-vector Transformer-TTS model
2. ESPnet **Chinese pre-training model**: Initial Transformer
- ✓ 3. VCC20 task2 **English + Chinese pre-training model**: multi-speaker, x-vector Transformer-TTS model

1、2 均為 ESPnet 上提供的預訓練模型，使用的預訓練 dataset 分別是 librispeech 英文資料集和 CSMSC 中文標準女聲資料庫，先前目標為從 1 換成 2，但在替換遇到問題時，在 baseline 的 task2 程式中發現了 3，它是由 librispeech + CSMSC 混和微調而成。

因此就使用了第三種預訓練模型作為替換方案，架構與 baseline 相同。

在預訓練模型中的 x-vector 是一個可以接受任意長度的輸入，然後轉換成固定長度的特徵表示的神經網路。

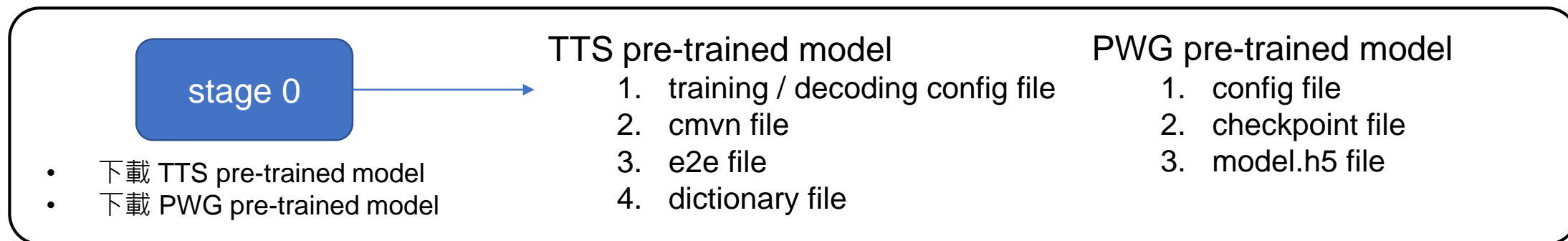
Before training

論文與 ESPnet 沒有標示至少要多少語句訓練效果會好，而論文遵從 VCC2020 競賽中，給予 70 句語句的限制，使用 60 句作為 training set，10 句作為 development set。

Baseline: 60/10

Baseline 使用了 60 句訓練，並使用 MOS(mean opinion score) 評估自然度以及相似度，分別是 3.5 分的自然度以及 90% 的相似度，在 VCC2020 上具有強大的競爭力。

在訓練前我們要先下載 TTS (Text-to-speech) 與 PWG (Parallel WaveGAN) 的預訓練模型，會得到以下 7 個檔案。並準備要訓練的語句與文本。



Training processes

TTS 訓練流程

1. Data preparation

- 選擇語料
- 建立語料與文本的關聯檔
- 降頻至 16kHz
- 檢查準備的資料目錄、格式是否正確

2. Feature Generation

(使用 TTS 預訓練中計算的統計資料，對特徵進行標準化)

- 切除空白音檔
- 生成 fbank
- 生成指定的 train、test 語句列表
- 使用預訓練 cmvn 取 train、test feature

3. Dictionary and Json Data Preparation

- 使用 TTS 預訓練中內置的字典對標記進行索引

4. x-vector extraction

- 生成 MFCC 並計算 energy-based VAD
- 對於 Kaldi-based X-vector pretrained model 提取 X-vector

5. fine-tuning

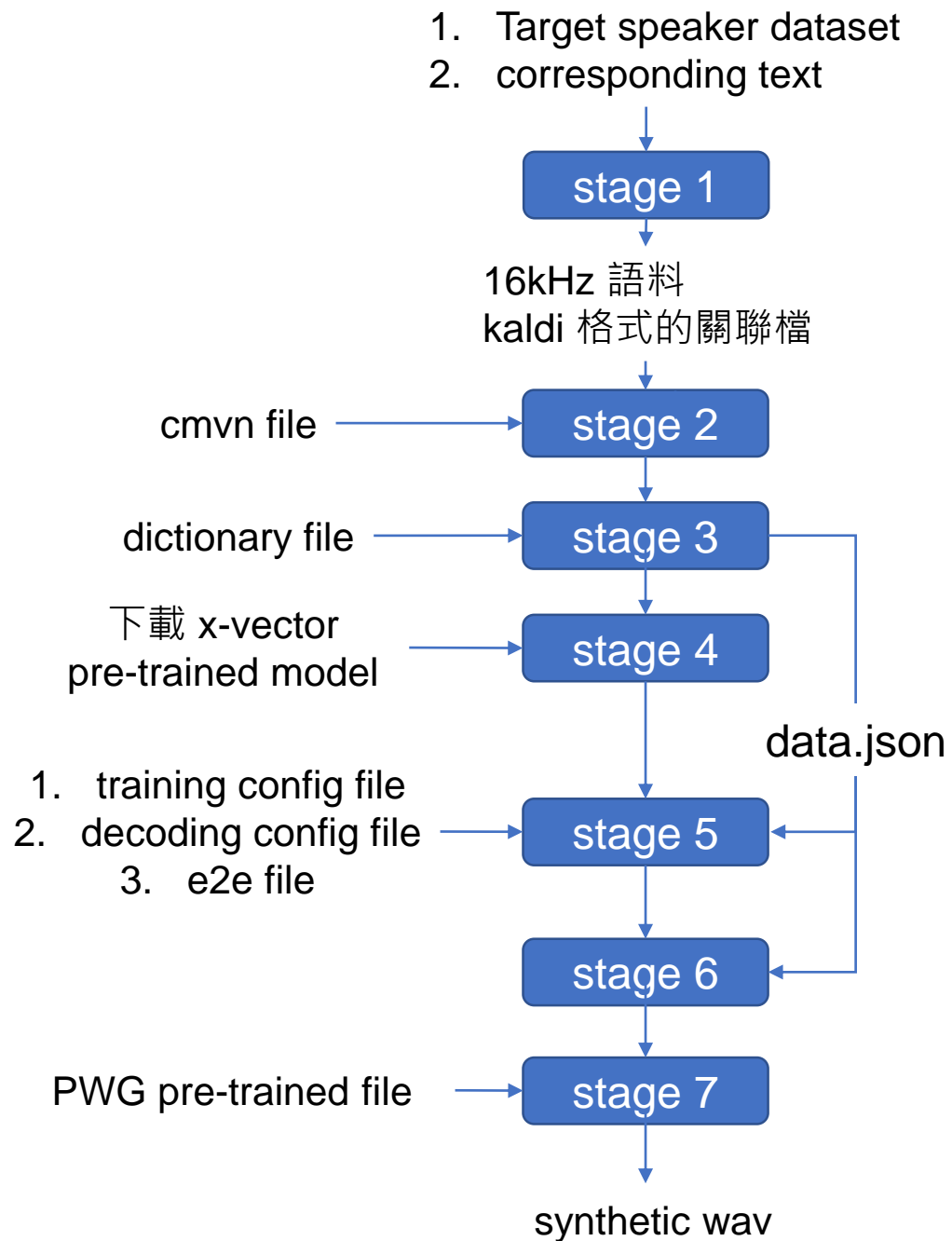
- Train E2E-TTS model (encoder)

6. Decoding

- 對於 test set 做 TTS 解碼 (decoder)

7. Synthesis


- 使用訓練過的 Parallel WaveGAN 將生成的 mel filterbank 轉回波形



Conversion result


版本 1、2 使用腳本中切除訓練用音檔前後的無聲區域，但會有部分音檔沒有切乾淨，以致**部分**合成結果音檔前會有無聲的問題 (圖二)，因此延伸三種方法在執行腳本前去除訓練音檔前後空白


版本	版本區別	訓練 / 測試 語句數量	音質表現
1	在執行腳本前沒經過處理	60 / 10	4
2	在執行腳本前沒經過處理	310 / 10	2
3	使用 Audacity 的截斷靜音	310 / 10	3
4	使用 IA 的 DTW	310 / 10	3
5	手動切除前後空白音檔	310 / 10	2
6	手動切 + 額外錄製語料	320 / 10	1



音色：  這學期學校有書法比賽

 全家人都非常  爸爸戴老花眼

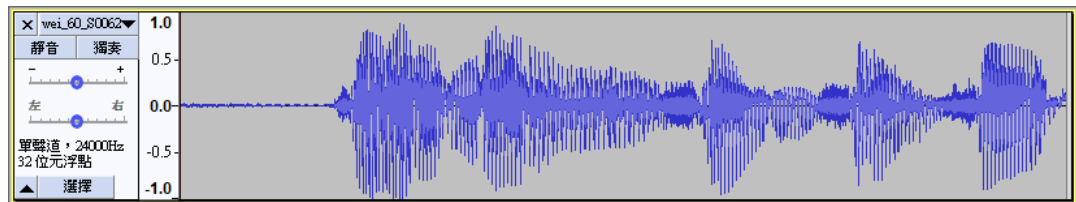
 他不小心把茶杯  他做完功課才

 他不小心把茶杯  他做完功課才

 他不小心把茶杯  他做完功課才

 他不小心把茶杯  他做完功課才

 這禮拜的天氣早  人生就像騎腳踏車想保持平衡



圖二：合成結果音檔前有空白

使用 Audacity 截斷靜音功能

使用 IA Lab 提供的 DTW 功能

手動切除前後空白音檔