Study of transformer-based TTS and its embedded implementation
基於變換器之文字語音轉換研究及嵌入式實現

Sian-Yi Chen

Advisors : Tay-Jyi Lin and Chingwei Yeh

論文章節規劃

1. Introduction
   - 研究背景
     - 提及語音轉換應用的實際情況，像是科技訪談中提及的有聲書，或是中正團隊在射月計畫下發展的DVC研究的故事，來描述語音轉換的重要性
     - 介紹語音轉換
   - 研究動機
     - 概述語音轉換分類
       - 古典
       - 現今
         1. 機器
         2. 深度
     - 看到古典、機器學習的缺點，或是說在相似度、自然度表現不佳，因此選了VCC2020提供了3種baseline的其中一種，希望藉由選用一個最具有競爭力的架構來改善自然度、相似度的表現
   - 三點貢獻
     - 生成指定對象聲音，並且音質、音色良好
     - C版本音質更好
     - Transformer C-based implementation，未來可以移植到嵌入式系統
   - 論文大綱
2. TTS Related work
   - 傳統
     - 發音合成 (Articulatory Synthesis)
     - 共振峰合成 (Formant Synthesis)
     - 拼接合成 (Concatenative Synthesis)
       - 單元選擇合成 (Unit Selection Synthesis)
       - 統計參數合成 (Statistical Parametric Synthesis)
   - 現今
     - 機器學習
       - Speech Model-Based
     - 深度學習
       - Encoder-Decoder
         - **ASR-TTS**

3. Multi-speaker, x-vector Transformer-TTS model
   - 系統架構
   - Transformer
     - 優點特色
     - 網路架構
       1. Word embedding
       2. Multi-head self-attention
       3. Masked Multi-head attention

4. Implementation of Transformer-based TTS
   - 環境設置：使用的電腦硬體規格、軟體框架版本、使用的平台、使用的語言
   - 轉換流程
     - 英文模型轉中文模型，中間需要特別轉換成拼音
     - fine-tune
   - 實驗
     - 調整網路中的超參數(更改特徵提取的頻率、FFT point)
     - 調整訓練語句數量
     - 使用不同人選、性別進行微調
   - C code程式架構
   - 軟體實現過程
     - 將軟體從linux移植至windows
     - 神經網路實現步驟
       1. 完成各零件：embedding、scaled-positional、linear、layernorm…等等
       2. pytorch取得bias、weight語法
       3. input、output格式
   - 將輸出經過cmvn、normalize、pwg可合成語音
   - 實驗結果

5. Conclusion

Outline

1. Introduction
   - 研究背景、研究動機、研究貢獻
2. TTS Related work
   - 傳統TTS
   - 現今TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - 架構
   - 前處理
   - Transformer
4. Implementation of Transformer-based TTS
   - 環境設置
   - 訓練流程
   - 實驗結果
5. Conclusion

Outline

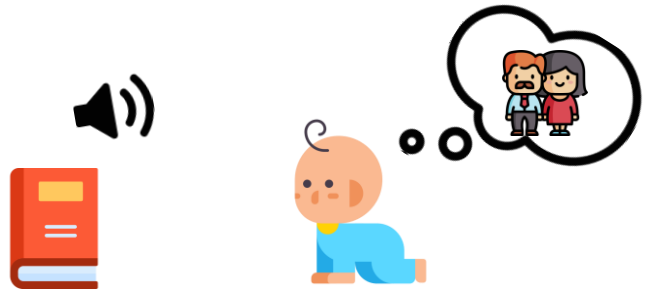
1. Introduction
   - 研究背景、研究動機、研究貢獻
2. TTS Related work
   - 傳統TTS
   - 現今TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - 架構
   - 前處理
   - Transformer
4. Implementation of Transformer-based TTS
   - 環境設置
   - 訓練流程
   - 實驗結果
5. Conclusion

研究背景

- 身處科技日新月異的時代，與機器的互動成為生活中不可或缺的一部分。有別於過往使用按鍵互動，如今使用語音進行互動已是常見的互動型態。

- 過去曾閱讀過一篇科學訪談，因為這次新冠肺炎的爆發，電子書的線上閱讀流量大量增加，也因此帶動了有聲書，而有聲書其中最重要的技術就是語音轉換，若能將有聲書內容替換為父母的聲音，孩子對熟悉的聲音產生安心感，亦減輕父母的照護壓力

- 不僅如此，語音轉換為中研院近年來的發展重點，本團隊在科技部半導體射月計畫下透過語音轉換的技術與醫院合作，協助中風病患等構音異常患者，讓他們的聲音能轉換成我們容易理解的資訊



- 擬父母有聲書示意圖

研究背景

- 語音轉換（voice conversion）意旨通過語音處理、人工智慧等技術，像是對一個人的聲音進行數位複製（digital cloning），或是修改音訊的波形，達到音調、音色、語速等聲學特徵的變化，讓來源語音（source voice）如同目標語音（target voice）說話

- 在語音轉換中，又有著許多不同追尋的面向，例如生成速度、自然度、清晰度、相似度...等。

研究動機

- 語音轉換中，由人工智慧的使用分為古典方法與現今方法，現今方法則又分為機器學習與深度學習，在古典方法與機器學習中不管是自然度或是相似性都仍然像是機器人

- 其中深度學習有著眾多方法，在Voice Conversion Challenge（VCC）2020中提供了三種baseline，PPG-VC、CycleVAE、ASR+TTS，其中選擇了架構最簡單且具有競爭力的Automatic Speech Recognition（ASR）+ Text to Speech（TTS），而在這個架構中ASR所辨識的結果準確率並不會影響到TTS的表現結果，彼此互相獨立，因此本投影片會針對TTS做說明

- 語音轉換中所使用的其中一種架構（ASR+TTS）

研究貢獻

1. 生成指定對象聲音，並且音質、音色良好

2. 在baseline中，女生合成的品質良好，但男生的結果會有明顯的雜音，而C版本Transformer的男生合成結果音質更好，沒有雜音

3. 使用C實現Transformer，並可以移植到嵌入式系統

Outline

1. Introduction
   - 研究背景、研究動機、研究貢獻
2. TTS Related work
   - 傳統TTS
   - 現今TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - 架構
   - 前處理
   - Transformer
4. Implementation of Transformer-based TTS
   - 環境設置
   - 訓練流程
   - 實驗結果
5. Conclusion

傳統TTS

1. 發音合成 (Articulatory Synthesis)：為模擬人類發聲器官的行為來產生語音
2. 共振峰合成 (Formant Synthesis)：根據 source-filter model 的規則生成語音
3. 拼接合成 (Concatenative Synthesis)
    - 單元選擇合成 (Unit Selection Synthesis)：從資料庫中選擇適合的單元進行拼接
    - 統計參數合成 (Statistical Parametric Synthesis)：使用GMM-HMM

- 缺點：不管是自然度或是相似性都仍然像是機器人

現今TTS

- 在 speech synthesis 分類中，傳統與現今技術我認為可以用 E2E model 作為分界點，而目前的聲學 (acoustic) 演算法主要分為三類，RNN based、Transformer based、CNN based。

- 我所使用的TTS是Transformer based，其中參考了VCC2020 (Voice Conversion Challenge 2020) 作為我的 baseline

- 在大會中有出現幾種神經網路Tacotron、LSTM、Transformer等，其中Transformer在MOS評分上分數又居於最高，且大會提供了baseline，因此選用它作為我的baseline

Outline

1. Introduction
   - 研究背景、研究動機、研究貢獻
2. TTS Related work
   - 傳統TTS
   - 現今TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - 架構
   - 前處理
   - Transformer
4. Implementation of Transformer-based TTS
   - 環境設置
   - 訓練流程
   - 實驗結果
5. Conclusion

系統架構圖

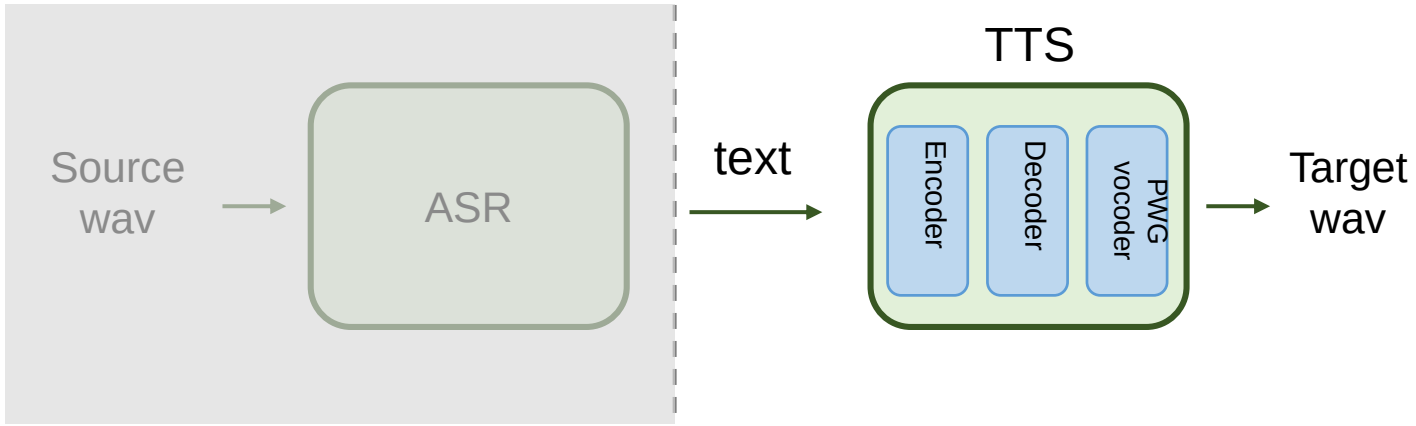


Figure 1: System structure

輸出入音檔：各 25 個音檔
(取其中一個音檔作為 demo)

source_1.wav

Conversion

Source speech

Using ESPnet pretrained model

Trained ASR model

Transcription

Using ESPnet pretrained model and finetune

Trained TTS model

Converted acoustic features

Trained neural vocoder

Converted speech

converted_1.wav

文字到JSON

Baseline JSON (for English model)

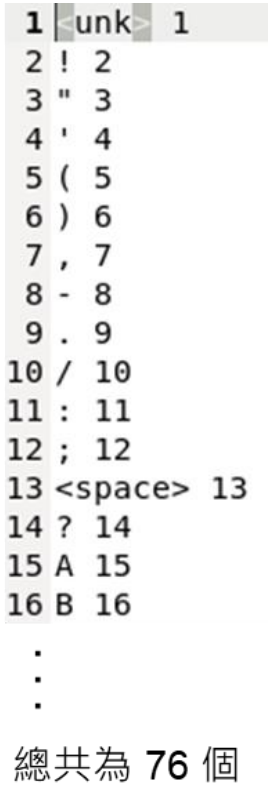
- 字典形式為將所有**字母**編號
- **for example**：Sentence: “There are”
  - 34 50 47 60 47 13 43 60 47

T h e r e ↓ a r e

<space>

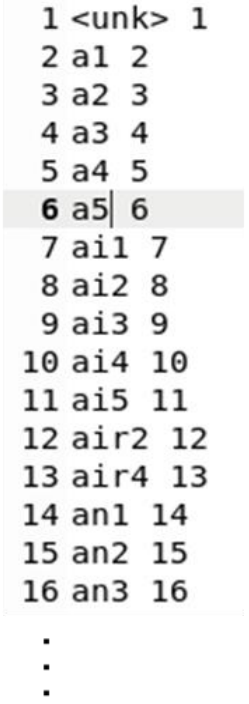
Mandarin pretrain JSON

- 字典形式為將所有**拼音**編號
- **範例**
  1. 小朋友們在看螞蟻搬家
  2. 轉換成拼音：x iao3 p eng2 iou3 m en5 z ai4 k an4 m a3 i3 b an1 j ia1
  3. 將拼音結果轉換成數字表示
  4. 將數字表示結果輸出成JSON檔



總共為 76 個

Baseline JSON
(for English model)



總共為 259 個

Mandarin pretrain
JSON

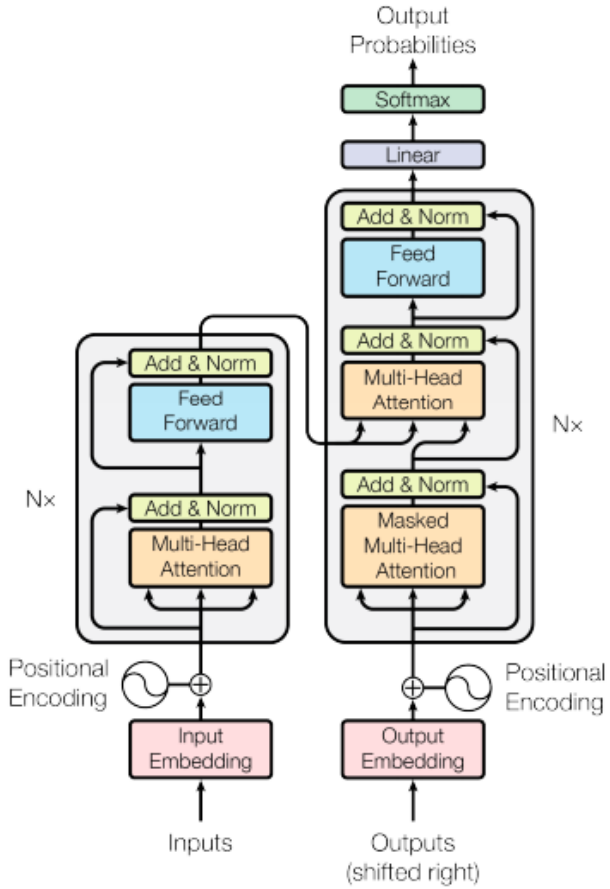
Transformer

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。

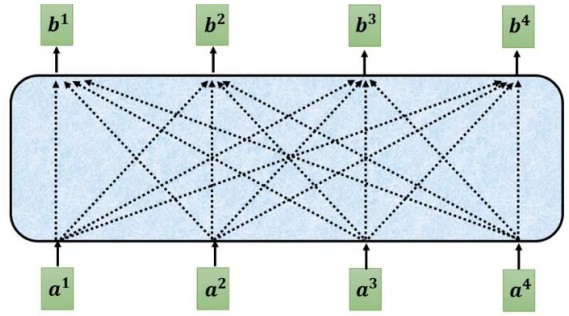
Sequence to sequence model 輸入一個sequence 就會輸出一個 sequence，而它的輸入、輸出的長度均由神經網路自行學習，兩者長度沒有對應關係。

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，還提升了運算效率。



The Transformer - model architecture.



Self-attention (自注意力機制)

Transformer – scaled position encoding

因為Transformer是一次性將資料平行輸入，因此若沒有scaled position encoding，就沒有位置資訊，意思就是就算句子全部都打亂，隨便輸入，最後結果也會是相同的，因為 Transformer 是看全域的資訊，所以還要再加上位置編碼（Position Embedding）

位置編碼產生的方式很多，論文作者採用的是右方的公式：

- pos：即位置
- 2i：編碼內偶數的位元，i ∈ {0, 1, 2, 3, …}
- 2i + 1：編碼內奇數的位元
- d_{model}：模型的 embedding 大小

$$PE = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

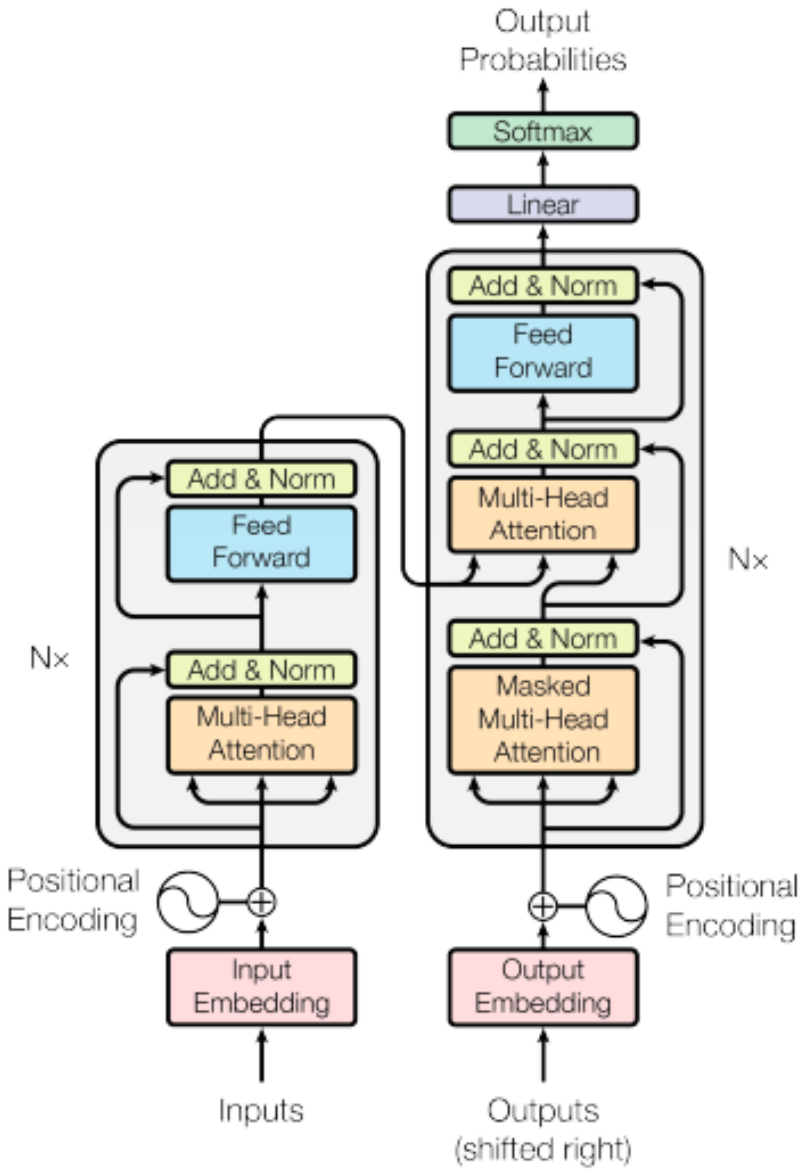
$$PE(pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}})$$

$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}})$$

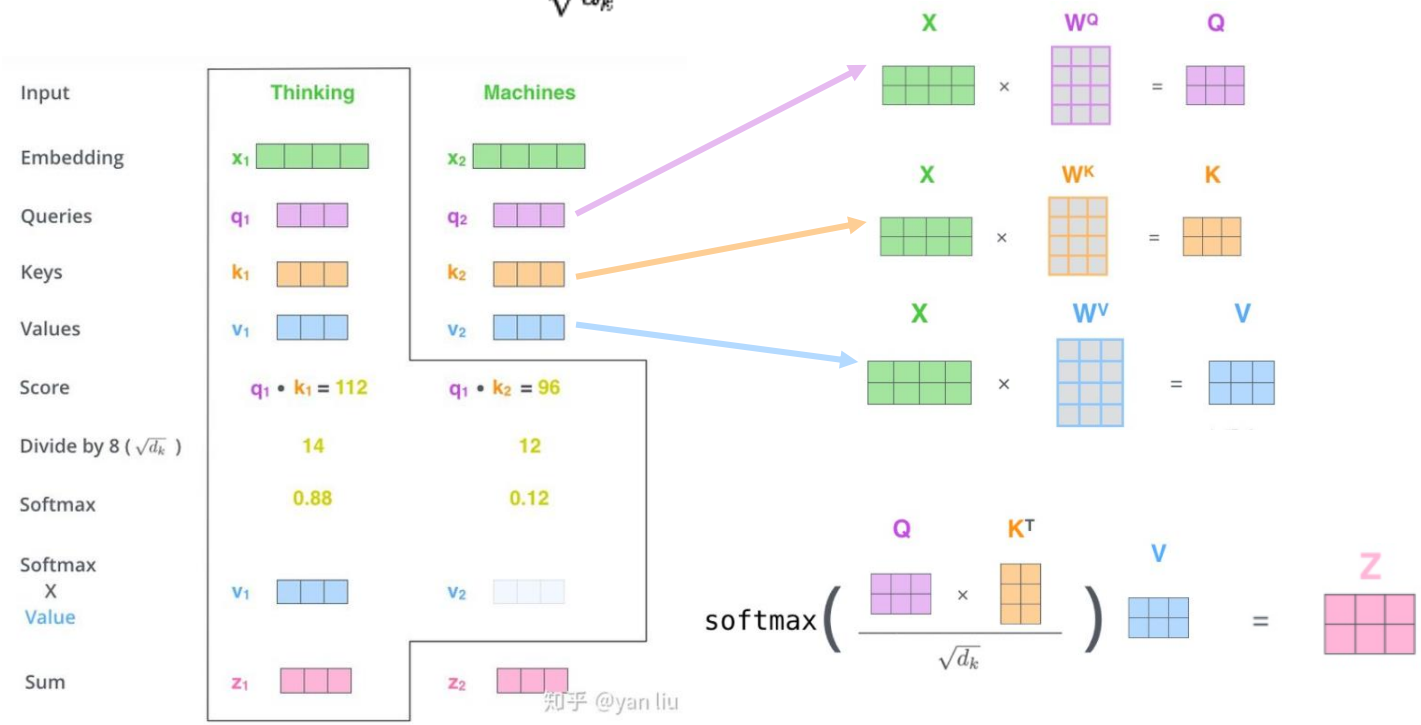
Positional Encoding(PE) formula and table

Transformer - attention

Transformer定義為：
第一個完全依賴 self-attention 去計算 input、output 之間關係
並不使用序列對齊的 RNN 或 CNN 的 transduction model。



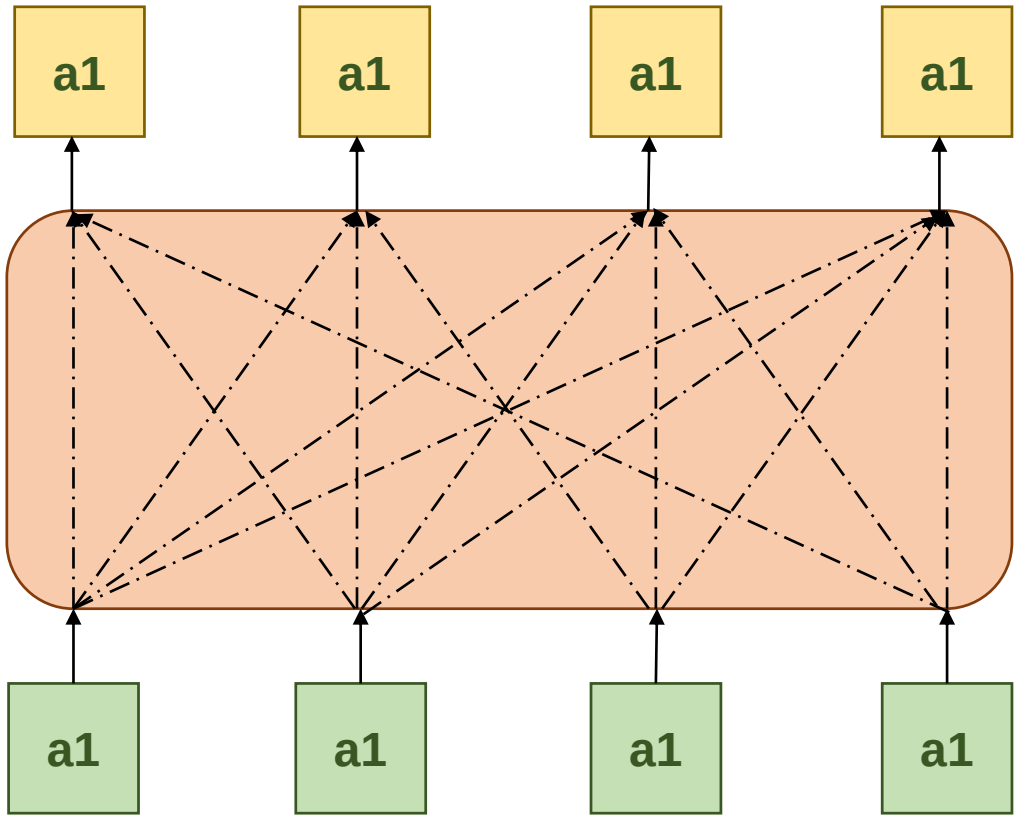
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



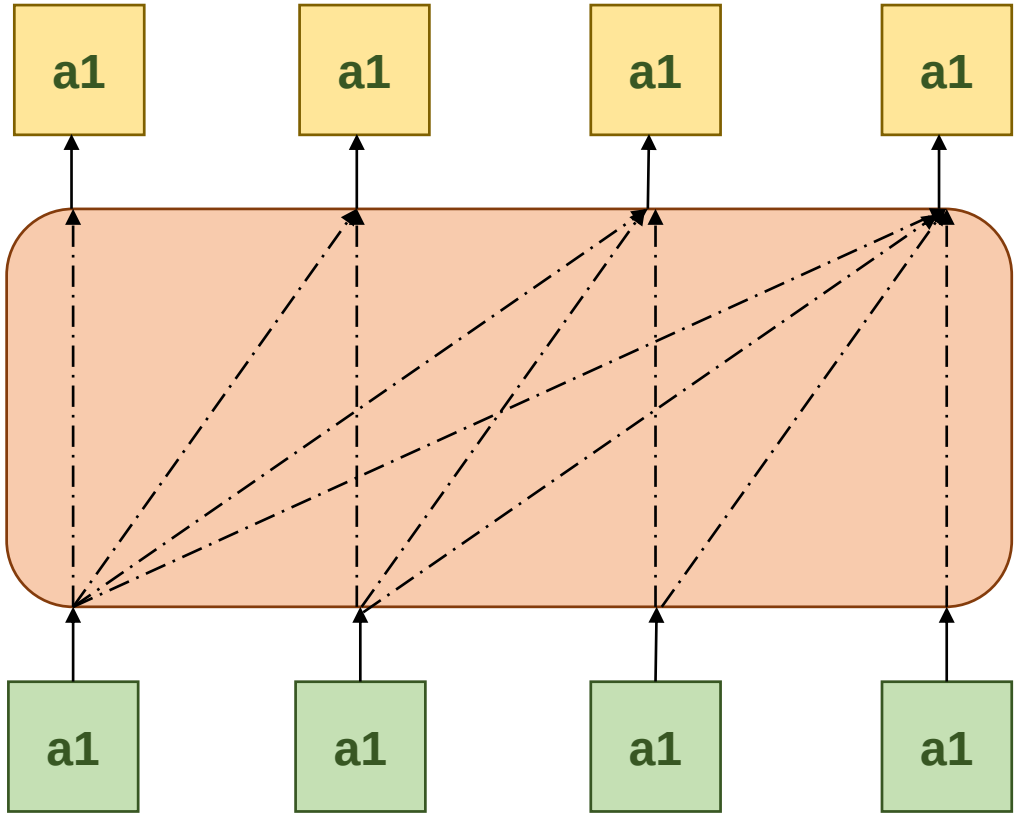
Query，Key，Value的概念取自於資訊檢索系統，舉個簡單的搜索的例子來說。
當你在網購平臺搜索某件商品（年輕女士冬季穿的紅色薄款羽絨服）時
你在**搜尋引擎**上輸入的內容便是 **Query**
然後搜尋引擎根據Query為你匹配 **Key**（例如商品的種類，顏色，描述等）
然後根據Query和Key的相似度得到匹配的內容（**Value**）。

Transformer – (mask) self-attention

Self -attention



Masked self -attention



原本是全部都考慮，現在不能用後面不知道的資訊

Outline

1. Introduction
   - 研究背景、研究動機、研究貢獻
2. TTS Related work
   - 傳統TTS
   - 現今TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - 架構
   - 前處理
   - Transformer
4. Implementation of Transformer-based TTS
   - 環境設置
   - 訓練流程
   - 實驗結果
5. Conclusion

實驗環境設置

- python 實驗環境
  - 作業系統：Linux Ubuntu20.04 LTS
  - python：3.8.8
  - torch：1.9.1+cu102
  - CPU：16 AMD Ryzen 7 1700 Eight-Core Processor
  - CUDA：11.4
  - GPU：geforce gtx 1080ti

- C 實驗環境
  - 作業系統：Windows 10
  - 編譯器：mingw x86_64
  - CPU：Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz

- Baseline
  - 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 e-prints, 2020

- 參數設置
  - 指定spk、pretrained_model_dir、voc、stage、stop_stage

實驗預訓練模型設置

TTS pre-trained model

1. training / decoding config file：編解碼器參數設定檔，ex：架構用 pytorch、使用 GPU、要訓練多少次…等等神經網路超參數
2. cmvn file： Cepstral Mean and Variance Normalization、提取聲音的特徵以後，將聲學特徵從一個空間轉換成另一個空間，使這個空間下的特徵參數更符合某種概率分布，壓縮特徵參數值域的範圍，減少訓練和測試環境不匹配等。
3. e2e file：Transformer 模型的參數， model.loss.best 檔
4. dict file：在微調前，將文本使用此字典標記索引，並記錄成 JSON 檔

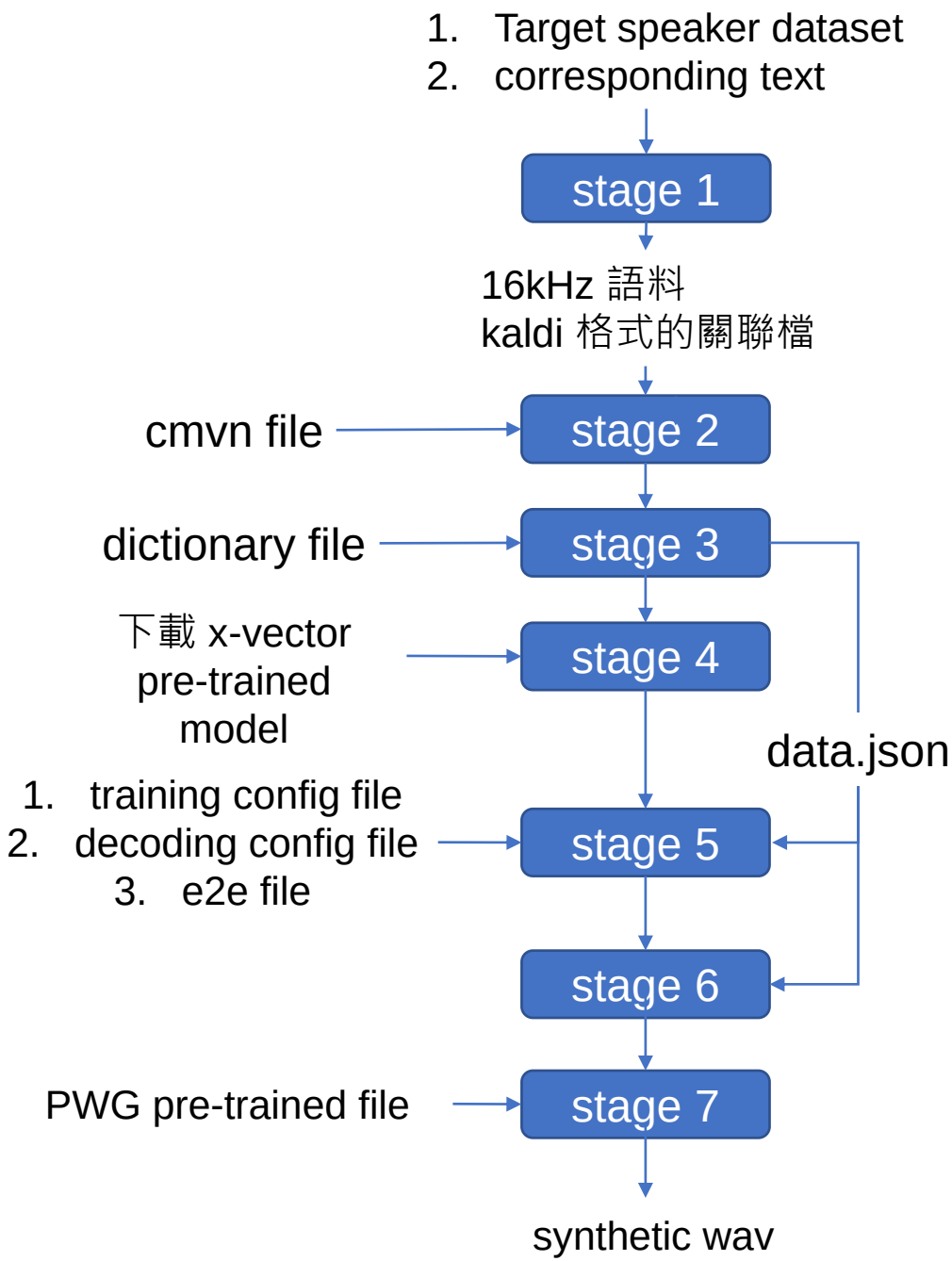
PWG pre-trained model checkpoint h5、pkl 保存模型和參數

1. 保存整个模型（.h5） => model.save (filepath) 将 Keras 模型和权重保存在一个 HDF5 文件中，該文件包含：模型的结构、模型的权重、训练配置（损失函数，优化器，准确率等）、优化器的状态
2. 保存模型权重（.h5）
3. 保存和加载整个模型（.pkl） => pytorch 保存資料的格式

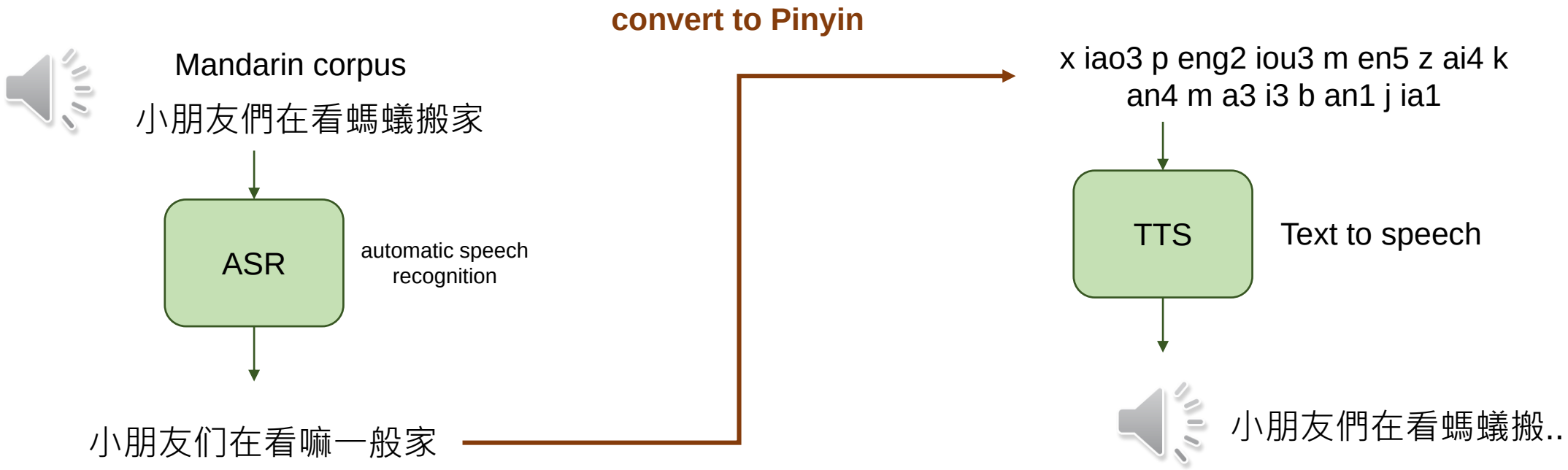
Training processes

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



轉換流程


convert to Pinyin
Mandarin corpus
小朋友們在看螞蟻搬家
ASR
automatic speech recognition
小朋友们在看嘛一般家
x iao3 p eng2 iou3 m en5 z ai4 k an4 m a3 i3 b an1 j ia1
TTS
Text to speech
小朋友們在看螞蟻搬..

實驗結果

Transformer 學習音色對象：震威
輸出語句：小朋友們在看螞 (蟻搬家) ，320句語料中的第301句

震威原音色

python 版本輸出結果

C 版本輸出結果

Outline

1. Introduction
   - 研究背景、研究動機、研究貢獻
2. TTS Related work
   - 傳統TTS
   - 現今TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - 架構
   - 前處理
   - Transformer
4. Implementation of Transformer-based TTS
   - 環境設置
   - 訓練流程
   - 實驗結果
5. Conclusion

結論

- 本論文使用基於VCC2020提供的baseline（Seq-to-Seq based on Cascade ASR + TTS w/ ESPnet）並將其從英文轉英文的模型使用網路上提供的預訓練模型替換成成中文轉中文的模型，並且在中間增加了中文轉拼音，最後生成指定對象聲音，並且音質、音色良好

- 另外，為了使該系統能應用在嵌入式系統上，使用C語言實現了Transformer-based的TTS，並且在baseline中，女生合成的品質良好，但男生的結果會有明顯的雜音，而C版本Transformer的男生合成結果音質更好，沒有雜音

參考文獻

[1] T. Hayashi, et al., ”ESPNET-TTS: unified, reproducible, and integratable open source end-to-end text-to-speech toolkit,” *arXiv preprint arXiv:1910.10909* , 2019

[2] 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

[3] A. Vaswani, et al., “Attention Is All You Need”, *arXiv preprint arXiv:1706.03762*, 2017.

[4] T. Yoshimura, “Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems,” *Ph.D thesis, Nagoya Institute of Technology*, Jan. 2002.

[5] H. Zen, K. Tokuda and A. W. Black, “Statistical parametric speech synthesis,” *speech communication*, Vol. 51, pp.1039-1064, 2009.