Survey of disentangled representation online resource w/ illustrating examples

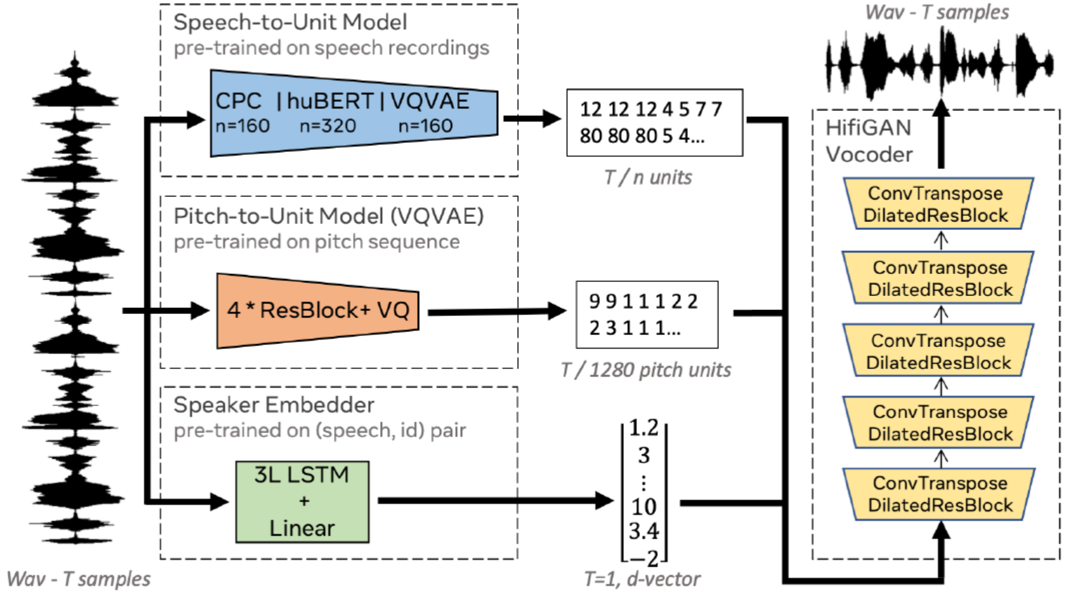
Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

Outline

- Action item
  - □ 看看 paper 有沒有提供 open source code，並重建 github project

- Status report
  - ① 論文並沒有提供原始碼，只有提供訓練前後的語音樣本
    - Audio samples link：https://speechbot.github.io/resynthesis/
  - ② 在圖一語音架構中，分別用了三個編碼器，和一個解碼器，其中除了 HuBERT 與 Speaker verification 以外皆有對應論文的 open source code
    - 編碼器
      1) 語音內容，由三個神經網路構成
         a. CPC [1]
         b. HuBERT [2]，找到同為 HuBERT 但為不同論文的 code
         c. VQ-VAE [3]
      2) 基本頻率 (YAAPY 演算法 [4] + VQ-VAE [3])
      3) 說話者 (Speaker verification [5])，為改進 [5] 的 code
    - 解碼器
      1) HiFi-GAN [6]
  - ③ CPC GitHub source code
    - 已成功解決大部分環境問題，但還在找訓練到一半中斷的問題，猜測是 C 碟容量不足導致



(圖一) Speech resynthesis architecture

6 個神經網路模型分別引用的論文：

[1] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” arXiv preprint arXiv:1807.03748, 2018.
[2] W.-N. Hsu et al., “Hubert: How much can a bad teacher benefit ASR pre-training?” in *NeurIPS* Workshop on Self-Supervised Learning for Speech and Audio Processing Workshop, 2020.
[3] A. van den Oord et al., “Neural discrete representation learning,” in *NeurIPS*, 2017.
[4] K. Kasi and S. A. Zahorian, “Yet another algorithm for pitch tracking,” in *ICASSP*, 2002.
[5] G. Heigold et al., “End-to-end text-dependent speaker verification,” in I*CASSP*, 2016.
[6] J. Kong et al., “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *NeurIPS*, 2020.

Open source code

- 編碼器
  1) 語音內容，使用三個神經網路構成
     a. CPC
        - GitHub - pat-coady/contrast-pred-code: Minimal implementation of Contrastive Predictive Coding for audio.
     b. HuBERT (同為 HuBERT 但為不同篇論文)
        - fairseq/examples/hubert at master · pytorch/fairseq · GitHub
     c. VQ-VAE
        - GitHub - 1Konny/VQ-VAE: Pytorch Implementation of "Neural Discrete Representation Learning“
  2) 基本頻率
     a. YAAPT
        - GitHub - mcraig2/pyaapt: Implementation of the YAAPT (Yet Another Algorithm for Pitch Tracking), an algorithm that determines the fundamental frequency of noisy signals (speech signals, for example).
     b. VQ-VAE (論文表示與語音內容編碼器使用相同的 VQ-VAE)
  3) 說話者
     a. Speaker verification (此為改進作者引用論文的開源碼)
        - GitHub - Janghyun1230/Speaker_Verification: Tensorflow implementation of generalized end-to-end loss for speaker verification
- 解碼器
  a. HiFi-GAN
     - GitHub - jik876/hifi-gan: HiFi-GAN：用於高效和高保真語音合成的生成對抗網絡

CPC (Contrastive Predictive Coding)

CPC [1]，對比預測編碼，目的是使用無監督學習取得高維數據中有用的表示，像是語意或是特徵，在不使用標註資料的情況下辨識資料間的關係。

[1] 提出了 CPC 這種方法，主要思想為：

- 對比：它使用對比方法進行訓練，即主模型必須區分正確和錯誤的數據序列。
- 預測性：模型必須在給定當前上下文的情況下預測未來模式。
- 編碼：模型在潛在空間中執行此預測，將代碼向量轉換為其他代碼向量（與直接預測高維數據相反）。

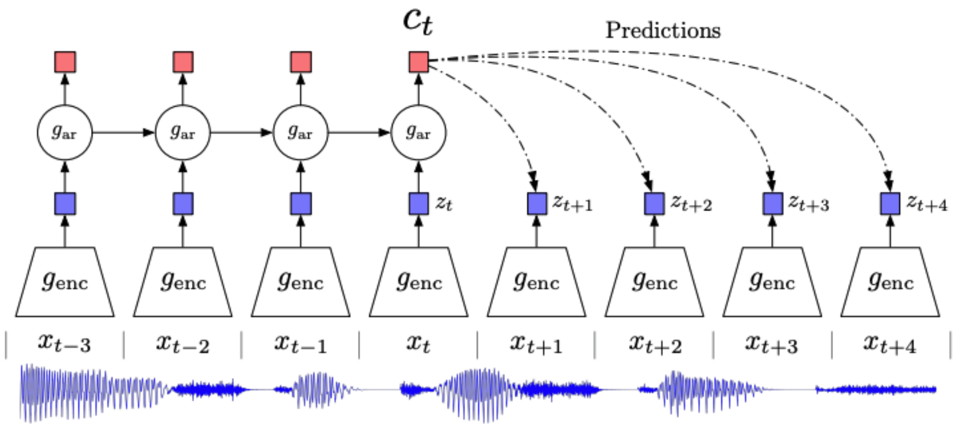
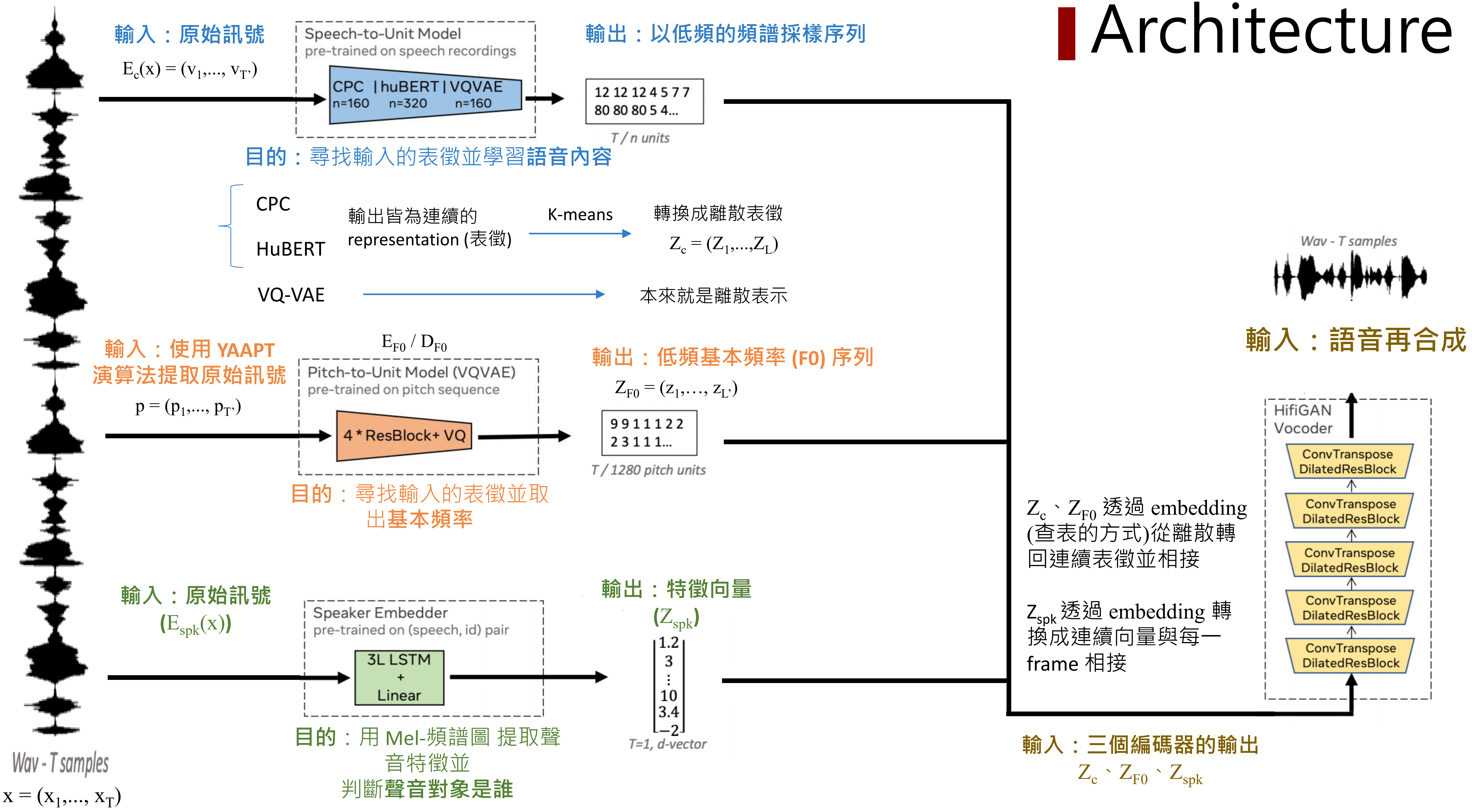


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

左圖中，編碼器 g_{enc} 取得原始音訊樣本，並輸出向量。透過自回歸網路根據上下文向量 c_t 預測未來的時間步長

[1] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” arXiv preprint arXiv:1807.03748, 2018.

Architecture



輸入：原始訊號

$E_c(x) = (v_1,\dots, v_{T'})$

Speech-to-Unit Model
pre-trained on speech recordings

CPC | huBERT | VQVAE
n=160 n=320 n=160

輸出：以低頻的頻譜採樣序列

12 12 12 4 5 7 7
80 80 80 5 4...

T / n units

目的：尋找輸入的表徵並學習**語音內容**

CPC
HuBERT
輸出皆為連續的 representation (表徵) —K-means→ 轉換成離散表徵 $Z_c = (Z_1,\dots,Z_L)$

VQ-VAE → 本來就是離散表示

輸入：使用 YAAPT 演算法提取原始訊號

$p = (p_1,\dots, p_{T'})$

E_{F0} / D_{F0}

Pitch-to-Unit Model (VQVAE)
pre-trained on pitch sequence

4 * ResBlock+ VQ

輸出：低頻基本頻率 (F0) 序列

$Z_{F0} = (z_1,\dots, z_{L'})$

9 9 1 1 1 2 2
2 3 1 1 1...

T / 1280 pitch units

目的：尋找輸入的表徵並取出**基本頻率**

輸入：原始訊號 $(E_{spk}(x))$

Speaker Embedder
pre-trained on (speech, id) pair

3L LSTM
+
Linear

輸出：特徵向量 (Z_{spk})

$$\begin{bmatrix} 1.2 \\ 3 \\ \vdots \\ 10 \\ 3.4 \\ -2 \end{bmatrix}$$

T=1, d-vector

目的：用 Mel-頻譜圖 提取聲音特徵並判斷**聲音對象是誰**

Wav - T samples

$x = (x_1,\dots, x_T)$

Z_c、Z_{F0} 透過 embedding (查表的方式)從離散轉回連續表徵並相接

Z_{spk} 透過 embedding 轉換成連續向量與每一 frame 相接

HifiGAN Vocoder

ConvTranspose DilatedResBlock

ConvTranspose DilatedResBlock

ConvTranspose DilatedResBlock

ConvTranspose DilatedResBlock

ConvTranspose DilatedResBlock

Wav - T samples

輸入：語音再合成

輸入：三個編碼器的輸出

Z_c、Z_{F0}、Z_{spk}