

TTS block diagram with clear I/O & processing of each block

Sian-Yi Chen

Advisors : Tay-Jyi Lin and Chingwei Yeh

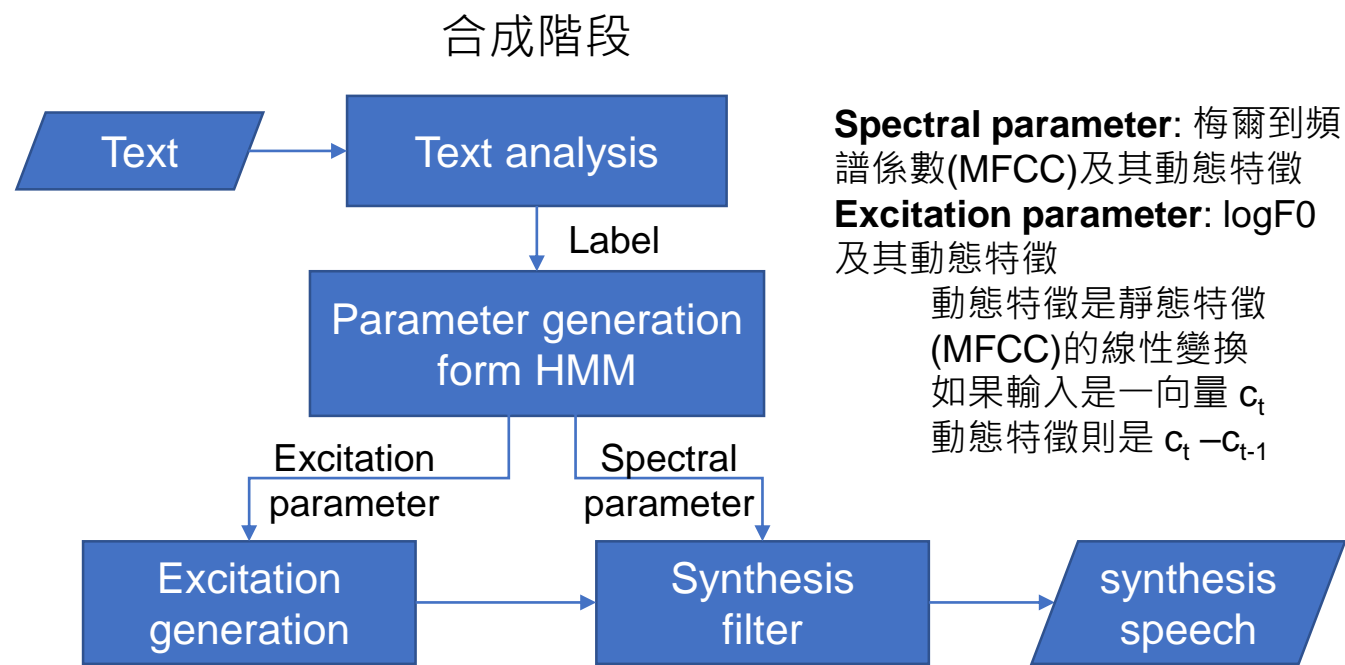
Outline

傳統 TTS : statistical parametric synthesis 統計參數合成 [1]

其中可以分為訓練及合成部分，在合成階段中，首先，輸入文字，將給定的單詞轉換成上下文相關的標籤序列，然後透過分類與回歸樹 (CART) 演算法挑選對應的 HMM 模型，接著根據這個 HMM 模型使用語音參數生成演算法生成頻譜參數還有激勵訊號參數，最後透過梅爾對數頻譜近似濾波器 (Mel Log Spectrum Approximation filter, MLSA filter) 生成語音訊號。

合成語音流程

- 從文字如何轉換成上下文標籤
- 標籤如何與 frame 對齊
- 使用 CART 挑選 HMM 模型
- 生成頻譜 (MFCC) 與激勵 (F0) 參數
- 使用 MLSA 生成波形



Context-dependent label

上下文標籤：輸入一段文字，解析其中音素、音節、單詞、短語以及發聲的數量之間的關係。

- 音素：之前、現在、之後的音素、位置以及子母音類別等
- 音節：之前、現在、之後的音節個數、重音、位置等
- 單詞：之前、現在、之後的詞性、位置、個數等

以 “Author of the danger trail.” 這段話作為舉例，它會將此句子拆分到音素，如右下圖所示，標籤就會依照多少音素生成多少標籤。

其餘資訊，像是位置，子母音為何則緊接在後方，詳細格式可參考附錄

第一個標籤：ao th er

第二個標籤：th er ah

第三個標籤：er ah v

.

.

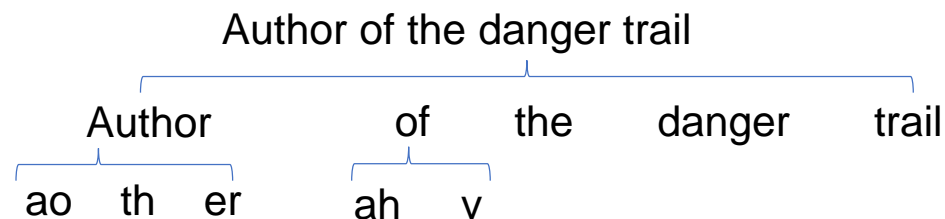
.

上下文音素標籤

句子

單詞

音素



句子拆分至音素

問題集

QS代表問題集，雙引號中間為問題名稱，大括弧內容則為問題的內容。

問題中包含前後聲韻母為何？韻律？位置？詞性？聲調？位置？特徵劃分等等
對每一個標籤詢問此問題集的每一題，因此每一個標籤就會有一個416維的向量。

問題一：QS "C-Vowel" {-aa+}

詢問第一個標籤是否有此元音

label ao th er 其餘資訊



[0 ...]

1*416 的向量

```
questions-radio_dnn_416.hed (~/.Merlin/merlin/misc/questions) - gedit
Open Save
QS "C-Vowel" {-aa+,-ae+,-ah+,-ao+,-aw+,-ax+,-axr+,-ay+,-eh+,-el+,-em+,-en+,-er+,-ey+,-ih+,-ix+,-iy+,-ow+,-oy+,-uh+,-uw+}
QS "C-Consonant" {-b+,-ch+,-d+,-dh+,-dx+,-f+,-g+,-hh+,-hv+,-jh+,-k+,-l+,-m+,-n+,-nx+,-ng+,-p+,-r+,-s+,-sh+,-t+,-th+,-v+,-w+,-y+,-z+,-zh+}
QS "C-Stop" {-b+,-d+,-dx+,-g+,-k+,-p+,-t+}
QS "C-Fricative" {-ch+,-dh+,-f+,-hh+,-hv+,-s+,-sh+,-th+,-v+,-z+,-zh+}
QS "C-Liquid" {-el+,-hh+,-l+,-r+,-w+,-y+}
QS "C-Front" {-ae+,-b+,-eh+,-em+,-f+,-ih+,-ix+,-iy+,-m+,-p+,-v+,-w+}
QS "C-Central" {-ah+,-ao+,-axr+,-d+,-dh+,-dx+,-el+,-en+,-er+,-l+,-n+,-r+,-s+,-t+,-th+,-z+,-zh+}
QS "C-Back" {-aa+,-ax+,-ch+,-g+,-hh+,-jh+,-k+,-ng+,-ow+,-sh+,-uh+,-uw+,-y+}
QS "C-Front_Vowel" {-ae+,-eh+,-ey+,-ih+,-iy+}
QS "C-Central_Vowel" {-aa+,-ah+,-ao+,-axr+,-er+}
QS "C-Back_Vowel" {-ax+,-ow+,-uh+,-uw+}
QS "C-Long_Vowel" {-ao+,-aw+,-el+,-em+,-en+,-ent+,-iy+,-ow+,-uw+}
QS "C-Short_Vowel" {-aa+,-ah+,-ax+,-ay+,-eh+,-ey+,-ih+,-ix+,-oy+,-uh+}
QS "C-Diphthong_Vowel" {-aw+,-axr+,-ay+,-el+,-em+,-en+,-er+,-ey+,-oy+}
QS "C-Front_Start_Vowel" {-aw+,-axr+,-er+,-ey+}
QS "C-Fronting_Vowel" {-ay+,-ey+,-oy+}
QS "C-High_Vowel" {-ih+,-ix+,-iy+,-uh+,-uw+}
QS "C-Medium_Vowel" {-ae+,-ah+,-ax+,-axr+,-eh+,-el+,-em+,-en+,-er+,-ey+,-ow+}
QS "C-Low_Vowel" {-aa+,-ae+,-ah+,-ao+,-aw+,-ay+,-oy+}
QS "C-Rounded_Vowel" {-ao+,-ow+,-oy+,-uh+,-uw+,-w+}
QS "C-Unrounded_Vowel" {-aa+,-ae+,-ah+,-aw+,-ax+,-axr+,-ay+,-eh+,-el+,-em+,-en+,-er+,-ey+,-hh+,-ih+,-ix+,-iy+,-l+,-r+,-y+}
QS "C-Reduced_Vowel" {-ax+,-axr+,-ix+}
QS "C-IVowel" {-ih+,-ix+,-iy+}
QS "C-EVowel" {-eh+,-ey+}
QS "C-AVowel" {-aa+,-ae+,-aw+,-axr+,-ay+,-er+}
```

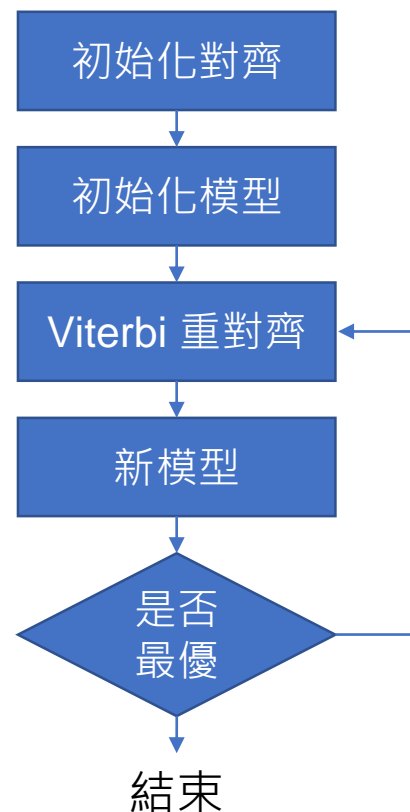
問題集，問題題數就是向量化的維度，此檔案包含416個問題

透過HMM將音素與 frame 對齊

最一開始不知道一段語音的哪些幀對應哪些狀態，因此進行**初始化**，也就是平均分配，假設 **author** 這個詞發聲 1.5 秒，若一個 **frame** 長 25ms，一次移動 25ms，則可以得到 60 個**frame**，也就是“ao”、“th”、“er” 每個音素各對應至 20 個 **frame**，每個音素又由 3 個狀態所組成，因此每個狀態分配到 6 或 7 個 **frame**。



音素對齊frame示意圖



HMM 訓練流程

HMM 初始化模型

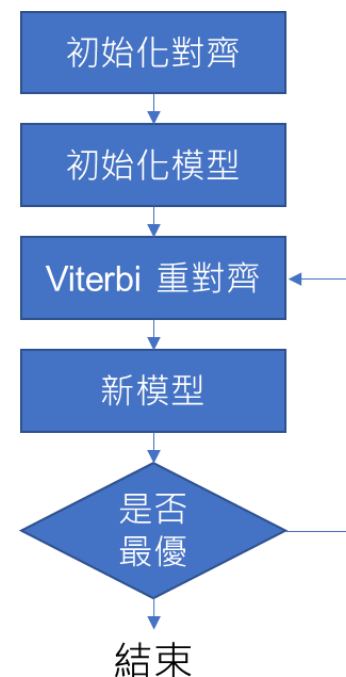
輸入：觀察序列 $O=[o_1, o_2, \dots, o_x]$ (X frame 的 MFCC 特徵)

輸出：通過模型計算每一 frame 對於 “ao”、“th”、“er” 這 3 個音素的某一狀態 (3狀態) 的機率

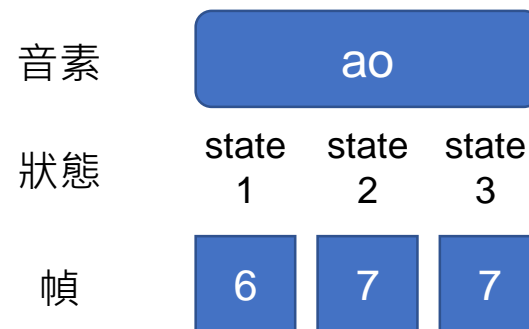
HMM模型 $\lambda=(A,B,\Pi)$

- 其中A是隱藏狀態轉移機率的矩陣
- B是觀測狀態生成機率的矩陣
- Π 是隱藏狀態的初始機率分佈

- 初始化完就可得到轉移機率 A，計算轉移次數 (狀態1->狀態1，狀態1->狀態2)，轉移次數/總轉移次數 = 轉移機率
- 初始機率分佈 Π ：HMM 模型是從左到右的模型，一開始在狀態1的機率為100%，所以此參數可忽略
- 狀態生成機率 B：一個狀態對應一個gmm模型，一個狀態又對應好幾個frame，所以好幾個frame對應一個gmm模型，初始化後，可得知狀態1對應6個frame，因此可以透過此計算狀態1的gmm模型 (單高斯模型)，求得平均值和變異數。



HMM 訓練流程



音素對應至frame

HMM 新模型

重新對齊

初始化結束需要重新對齊，使用的是viterbi演算法，根據初始化模型 $\lambda=(A,B,\Pi)$ 來計算，記錄每個時刻的每個可能狀態最優路徑概率，同時記錄最優路徑的前一個狀態，不斷向後反覆運算，找出最後一個時間點的最大概率值對應的狀態。

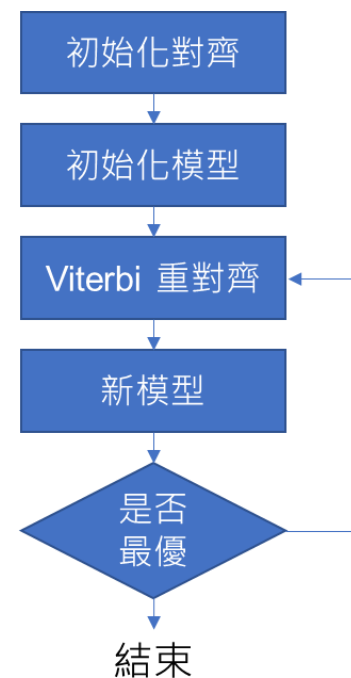
Viterbi是一種動態規劃演算法。它用於尋找最有可能產生觀測事件序列的維特比路徑(隱含狀態序列)

反覆運算

新的A和新的B又可以進行下一次的Viterbi演算法，尋找新的最優路徑，得到新的對齊，新的對齊繼續改變著參數A、B。如此迴圈反覆運算直到收斂，則GMM-HMM模型訓練完成。

反覆運算何時是個頭？

一般是設置固定輪數，也可以看一下對齊之後似然的變化，如果變化不大了，基本就是收斂了。)



HMM 訓練流程

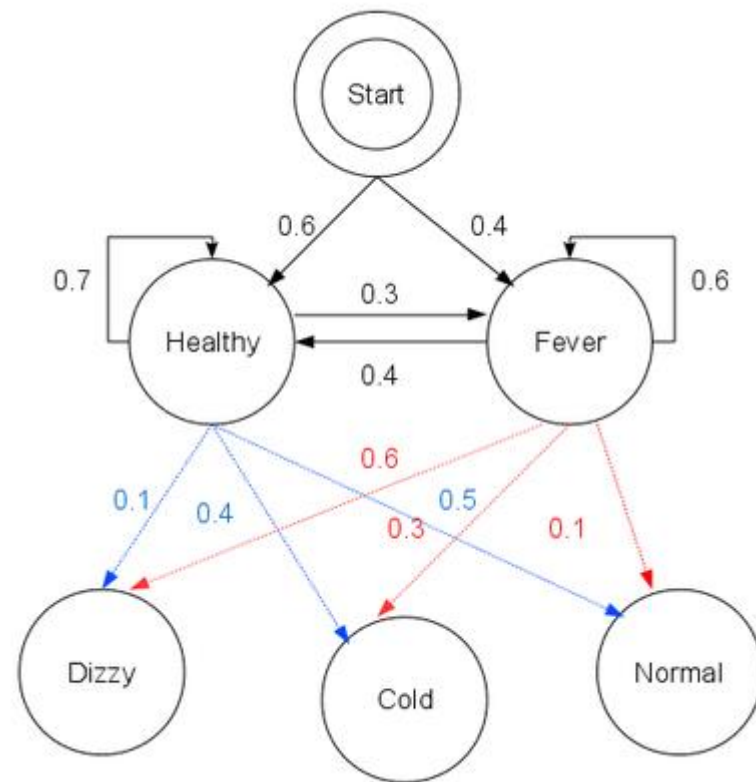
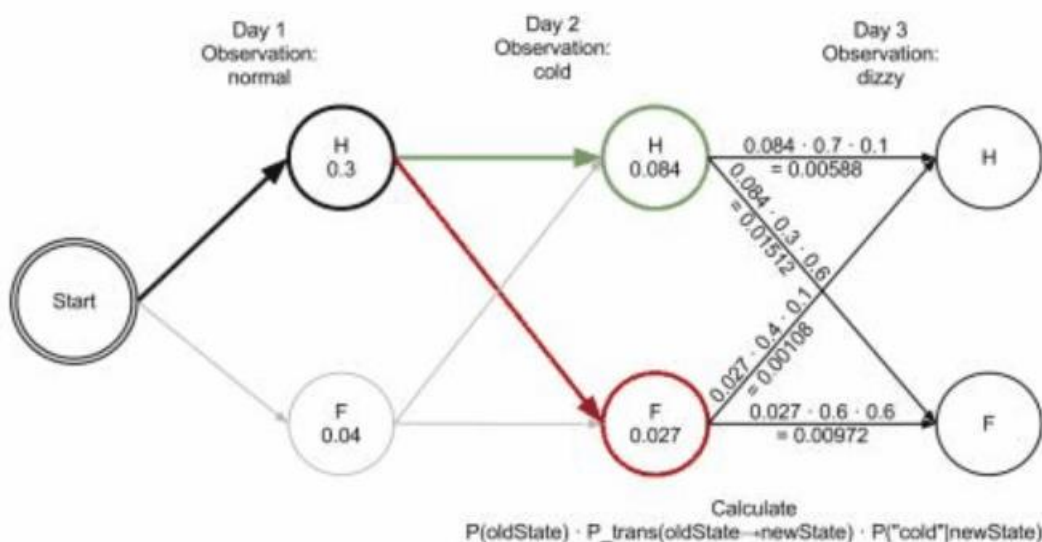
Viterbi演算法概念

今天有一位醫生要判斷病人是健康的還是發燒，病人只會回答正常、頭暈、冷，醫生要從這3個答案中判斷是否發燒，是否發燒就是隱藏狀態(無法直接觀察到)
右圖為病人各狀態的機率：

- 當天健康的病人隔天只會有30%的機率會發燒
- 如果病人是健康的會有50%的機率覺得正常
- 如果病人發燒了會有60%的機率覺得頭暈

病人連續看醫生3天，得以下結果：[正常、冷、頭暈]

根據viterbi演算法可以計算出3天的狀態分別是：[健康、健康、發燒]



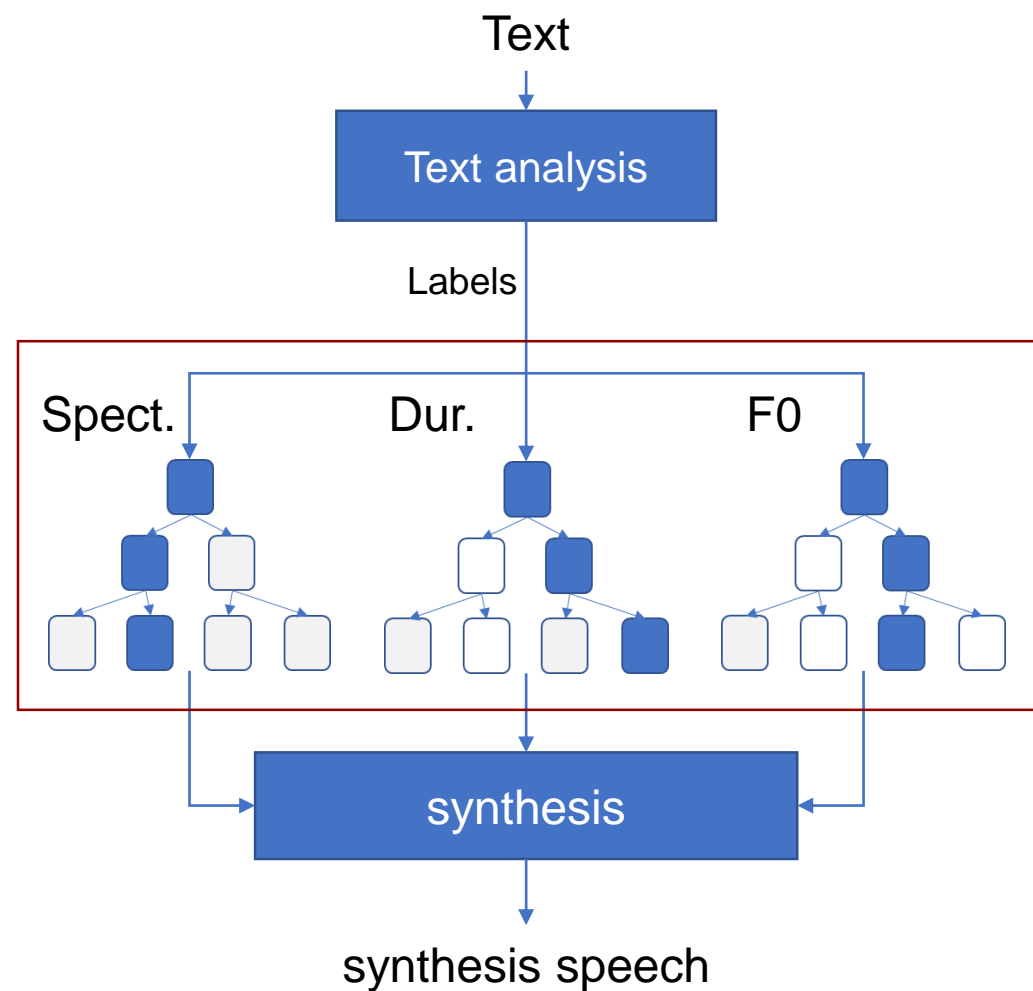
透過CART分類和迴歸樹選擇適合的HMM模型

CART：為決策樹的一種，在條件下輸出的條件概率分佈的學習方法。

CART是二元樹，每個節點取值方式為判斷“是”與“否”，左邊為“是”的分支，右邊則為“否”。

1) 樹的生成：基於訓練數據集生成決策樹，生成的決策樹要儘量大。

2) 樹的剪枝：用驗證數據集對已生成的樹進行剪枝並選擇最優子樹，這時損失函數最小作為剪枝的標準。



statistical parametric synthesis

附錄

■ 上下文標籤格式

完整上下文標籤：

sil^sil-ao+th=er@1_2/A:0_0_0/B:1-1-2@1-2&1-7#1-4\$1-3!0-2;0-

4|ao/C:0+0+1/D:0_0/E:content+2@1+5&1+2#0+3/F:in_1/G:0_0/H:7=5@1=2|L-L%/I:7=3/J:14+8-2

所有特殊位元皆為固定格式的連接用符號，無意義

可以看到除了 p1~p7 總共分為 9 種關係，各類所代表的物理意義可參照下一頁每一種顏色對應其關係格式：

1. p1^p2-p3+p4=p5@p6_p7
2. /A:a1_a2
3. /B:b1-b2@b3-b4&b5-b6#b7-b8!b9-b10|b11
4. /C:c1+c2
5. /D:d1_d2
6. /E:e1+e2@e3+e4
7. /F:f1_f2 /G:g1_g2
8. /H:h1=h2@h3=h4
9. /I:i1_i2
10. /J: j1+ j2- j3

■ 生成上下文標籤

範例句子：Author of the danger trail

Author
ao th er

第一個標籤

持續時間 $x^x \cdot \text{sil} + \text{sil} = \text{ao}$ 其餘資訊 狀態一
開頭沒有聲音，sil(靜音)

第二個標籤

持續時間 $x^x \cdot \text{sil} + \text{sil} = \text{ao}$ 其餘資訊 狀態二

⋮

第六個標籤

持續時間 $\text{sil}^x \cdot \text{sil} \cdot \text{ao} + \text{th} = \text{er}$ 其餘資訊
前前音素 前音素 當前音素 下一個音素 下下一個音素

標籤格式(音素、音節、單詞語、短語、句子之間的關係)

2050000 2400000 sil^sil-ao+th=er@1 2/A:0 0 0/B:1-1-2@1-2&1-7#1-4\$1-3!0-2:0-4|ac/C:0+0+1/D:0_0/E:content+2@1+5&1+2#0+3/F:in_1/G:0_0/H:7=5@1=2|L-L%/I:7=3/J:14+8-2

前後音素、該音素在音節中的位置

前一個音節是否為重音、音素數量

當前音節重音、音素數量、在單詞中的位置、在短語中的位置...等

下一個音節是否為重音、音素數量

前一個單詞詞性、音節數量

當前單詞詞性、音節數量、在短語中的位置、單詞數量、距離...等

下一個單詞詞性、音節數量

前一個短語中的音節數量、單詞數量

當前短語中的音節數量、單詞數量、在語句中的位置

下一個短語中的音節數量、單詞數量

此話語中的音節、單詞、短語數量

句子

Author of the danger trail

單詞

Author of the danger trail

音素

ao th er ah v

狀態

state1 state2 state3 state4 state5 state6 ... state20

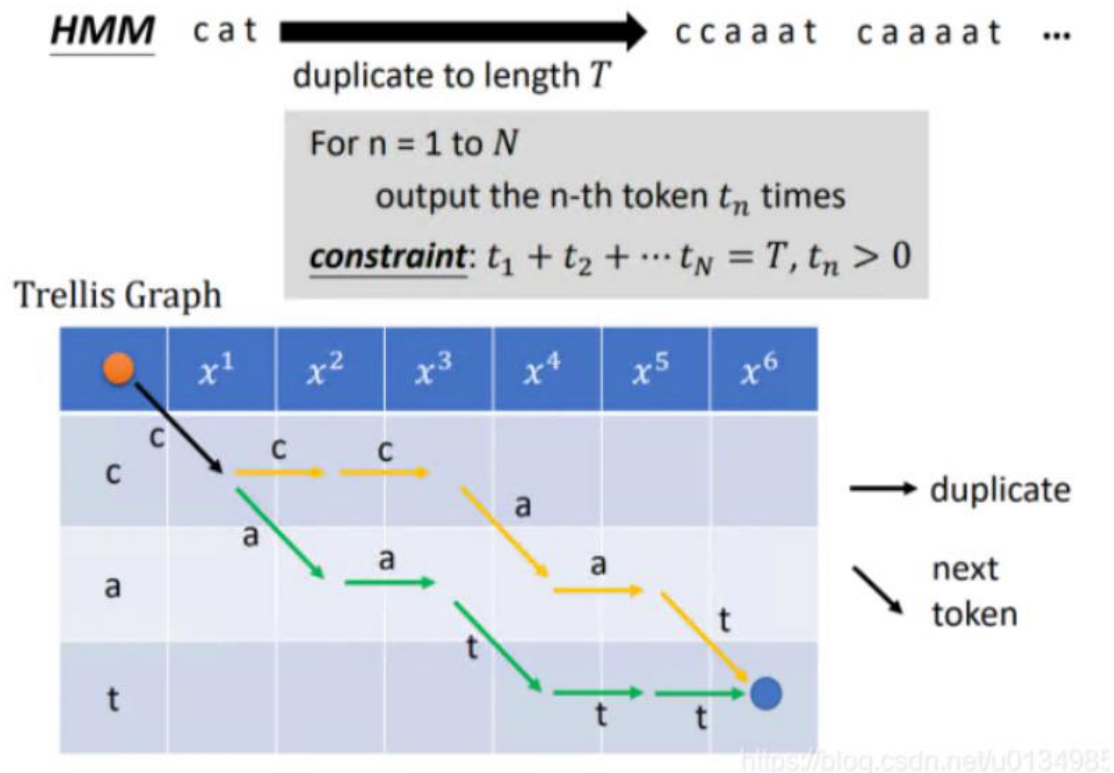
幀



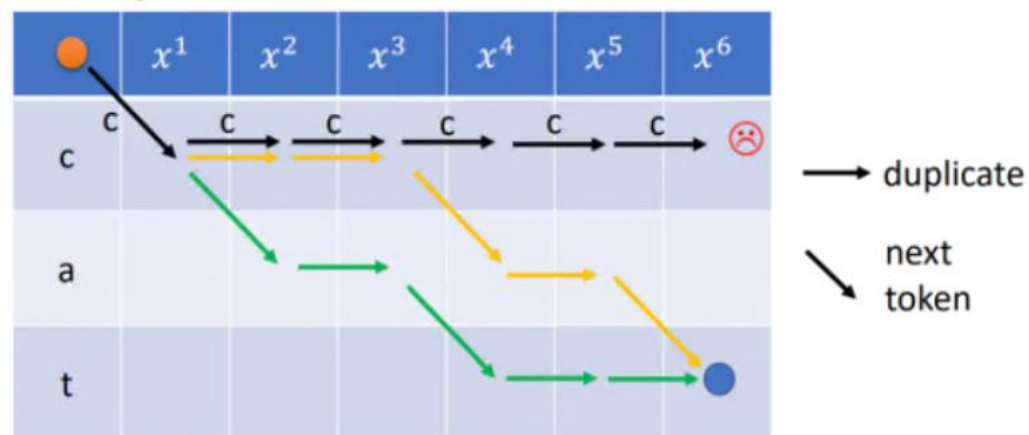
HMM 以5個狀態對齊示意圖

HMM alignment

HMM (Hidden Markov Model)



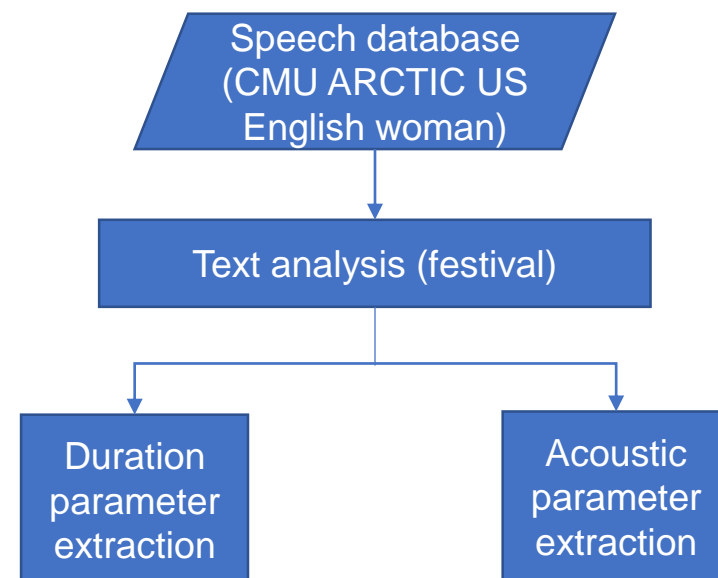
一個token可以重複N次，但是所有token重複的次數和要等於acoustic features的長度T，也就是灰色部分所描述的公式。表中橫軸代表acoustic features，縱軸表示token。從左上角開始走到右下角，每步只能有兩個方向：向右走或是向下走，但終點一定要走到右下角才算是合法的路徑。從起點一直走到終點所有合法的路徑就是所有可能的alignment。



Training part

訓練階段

- 輸入文字和語料庫，輸出文字對齊音檔的向量，輸入兩個模型進行訓練
- 文本分析
 - input：語料庫、語料文本
 - output：訓練用的向量特徵
 - processing
 - 1) 生成HTS(HMM-based 語音合成系統)上下文標籤
 - 2) 透過問題集轉換成向量
- 神經網路模型(持續時間與聲學模型)
 - input：皆為由問題集轉換的向量檔案
 - output
 - 1) Duration model：各個音素的持續時間
 - 2) Acoustic model：音色、音高、頻譜包絡(聲道的形狀)等等聲學特徵



Training flow

WORLD Vocoder 演算法與spectral envelope和MFCC之間的關係

□ 演算法

- 提取F0的演算法(DIO)：通過低通濾波器對原始訊號進行基頻的提取，具體流程是取4個週期計算標準差，並選最低的作為基頻
- 頻譜包絡Spectral Envelope：有三種方法可以取得，LPC、Cepstrum、CheapTrick
 - LPC：一個語音可以用多個語音過去值的加權線性組合來逼近
 - Cepstrum：訊號→FFT→絕對值→對數→相位展開→IFFT→倒頻譜
 - CheapTrick：音高同步分析
 1. F0-adaptive windowing：語音分段不以frame為單位，以f0對應的週期為單位，以保證波形和頻譜的平滑連續，使用hanning window
 2. smoothing of the power spectrum：對時域訊號做FFT，並在三角窗內對訊號進行平滑
 3. liftering in the quefrency domain：將功率頻譜看做是普通訊號，求出訊號的包絡就是找到其低頻
 - 1) 對功率頻譜做IFFT
 - 2) 過濾訊號的到低頻
 - 3) 頻譜恢復：消除先前平滑帶來的變異
 - 4) 取得包絡
- 非週期信號參數：一般的語音都是有周期信號和非週期信號組成，所以，除了以上獲取週期信號的參數，我們還需要得到其中的非週期信號參數，才能完美的合成原始信號。

□ spectral envelope和MFCC之間的關係

- 作者表示WORLD在取得Mel-cepstral與常見方法不同
- 一般會經過FFT再經過三角濾波器，因為MFCC是在頻譜圖上進行，因為未經過平滑處理，所以需要濾波器。
- WORLD是在頻譜包絡上進行，已經有平滑處理，所以流程上看似沒有濾波器但使用效果卻與常見方法想同。

Synthesis voice by WORLD Vocoder

Synthesis part

