Decision Tree State Clustering With Word and Syllable Features

By Hank Liao, Chris Alberti, Michiel Bacchiani, and Olivier Siohan / September, 2010

reporter：許妙鸞
Professor：陳嘉平

Abstract

- In large vocabulary continuous speech recognition, decision trees are widely used to cluster triphone states.
- In addition to commonly used phonetically based questions, others have proposed additional questions such as phone position within word or syllable.
- This paper examines using the word or syllable context itself as a feature in the decision tree, providing an elegant way of introducing word- or syllable-specific models into the system.
- Positive results are reported on two state-of-the-art systems: voicemail transcription and a search by voice tasks across a variety of acoustic model and training set sizes.

- **Index Terms:** decision tree state clustering, large vocabulary continuous speech recognition, tagged clustering.

Introduction

- 最先進的連續語音辨識詞彙系統，使用決策樹去聚集上下文相依的HMM 狀態
- 上下文相依的模型依條件分成左右兩邊的phone，稱之爲 triphone
- Triphone 的數量相當大，不是所有的triphone都可以在訓練資料時被遵守，這導致資料稀疏的議題

Decision Tree State Clustering

- 決策樹常被用在大量詞彙的連續自動語音辨識，把聚集到的大量CD分成小的集合，其分佈可以被評估
- Context 的小資料被結合直到資料足夠使用
- 標準分詞的聲學單位是triphone
- 如:

trees → t+r t-r+iy r-iy+z iy-z

決策樹使用語音問題分類

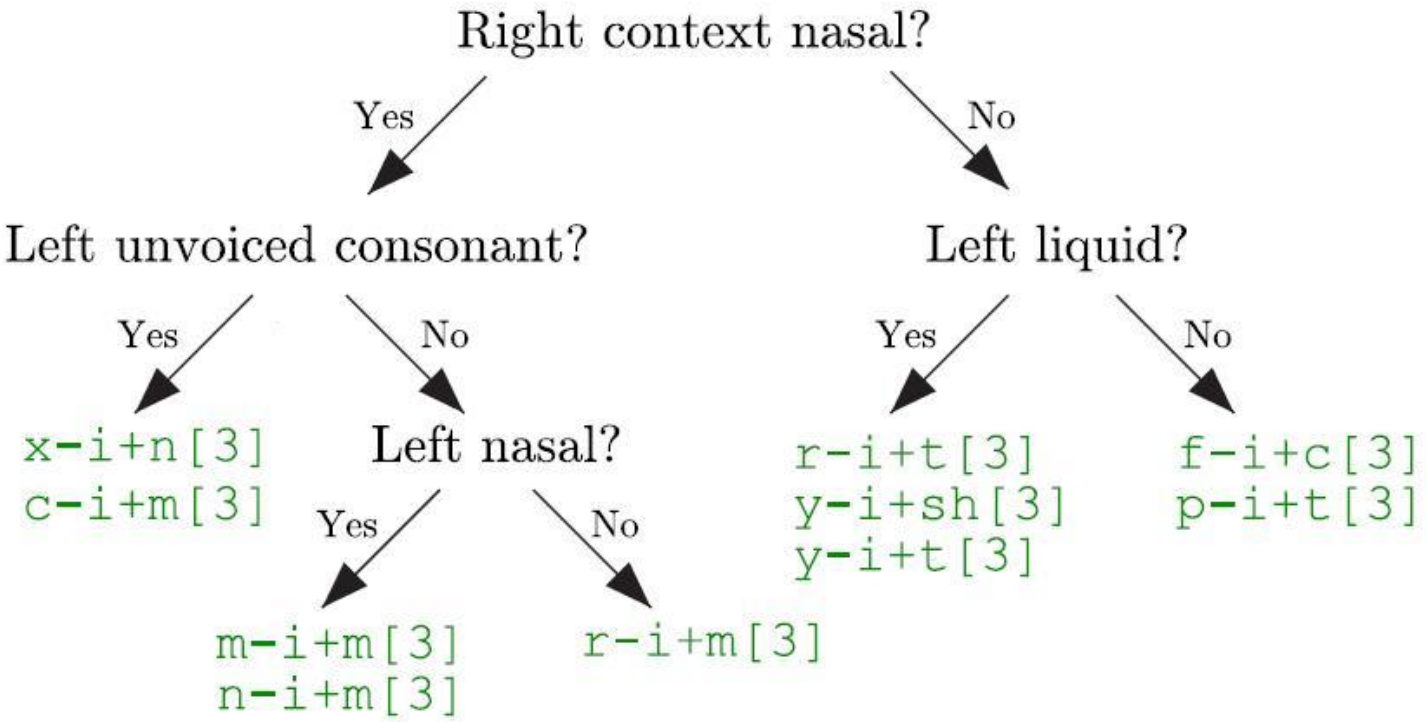

Figure 1: Decision tree clustering state 3 of phone i. 10 contexts are clustered into 5 leaves/states.

父節點的平均值和變異數

$$\boldsymbol{\mu}^p = \frac{1}{N^p} \sum_{c \in p} \boldsymbol{m}^c \qquad \boldsymbol{\Sigma}^p = \frac{1}{N^p} \sum_{c \in p} \boldsymbol{S}^c - \boldsymbol{\mu}^p \boldsymbol{\mu}^{p\mathsf{T}}$$

$$N^p = \sum_{c \in p} N^c$$

N^c : total state occupancy

p:a parent node

c:context

μ^p :mean

Σ^p: 變異數

m^c : 一階的context c 統計

S^c : 二階的context c 統計

每一個節點的變化

$$\mathrm{LL}_{\mathrm{gain}} = \log\left[\frac{L(X_{\in y}|\boldsymbol{\mu}^y, \boldsymbol{\Sigma}^y;)L(X_{\in n}|\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n)}{L(X_{\in p}|\boldsymbol{\mu}^p, \boldsymbol{\Sigma}^p)}\right]$$
$$= -\tfrac{1}{2}N^y \log(|\boldsymbol{\Sigma}^y|) - \tfrac{1}{2}N^n \log(|\boldsymbol{\Sigma}^n|)$$
$$+\tfrac{1}{2}N^p \log(|\boldsymbol{\Sigma}^p|)$$

y : yes child node

n : no child node

$X_{\in i}$: 訓練的資料和node i 相關

$X_{\in p} = X_{\in y} \cup X_{\in n}$

Word and Syllable Context

- 這篇paper提出以真實的單字或音節的上下文爲條件的phone model
- 在一個FST爲基礎的ASR系統，一個加權FST被用來表示統計的語言模型G,一個語音的詞彙L,上下文相依的轉換器 C
- 最佳化的decoding graph是使用FST做minimization, determinization and composition

Decoding Graph

$$\min(C \circ \det(L \circ G))$$

$$\min(\det(CL)) \circ G$$

det : determinization

G :統計的語言模型(weighted word acceptor)

L :一個語音的詞彙

(context independent phone to word transducer)

C:上下文相依的轉換器

(CD phone model to context independent phones)

CL

```
1   2                    3                      4
0  1   t@word=trees#wb=true    trees
1  2   r@word=trees#wb=false
2  3  iy@word=trees#wb=false
3  4   z@word=trees#wb=true
```

1: from state
2: to state
3: input label
4: output label
wb: word boundary
@、#: 分開phone和特徵關鍵值的配對

CL with left context

```
0 1   t@word=trees#wb=true#left=sil trees
1 2   r@word=trees#wb=false#left=t
2 3  iy@word=trees#wb=false#left=r
3 4   z@word=trees#wb=true#left=iy
```

Chou Partitioning

- Chou's partitioning algorithm(CPA) 被用來尋找在決策樹的節點最佳分割的資料
- 可被想成是K-means clustering 的應用
- 表示可能的分割的兩個群集被初始化，藉由分裂高斯的父節點
- 意指一些分數的變異,± 0.2被用在這項工作中，且k-means的一些相互作用會執行到收斂爲止

實驗

- 分成兩個工作
  - Voicemail transcription
  - A search by voice task a.k.a
- 在這篇paper所有的系統使用大量的定向搜索，使得搜尋錯誤在實驗中不是一個因素
- $M = \beta N^{0.4}$
  - M:高斯的數量
  - N: 觀察的數量

Voicemail Transcription

- 語音信箱轉錄系統訓練425小時的資料，大約50k voicemails
- 兩個測試集合總數大約有35k的單字或超過3小時的語音
- 語言模型是 Kneser-Ney smoothed, entropy pruned , trigram language model,從各種資源,包括轉錄他們本身(voicemail)和廣播新聞加入字詞

Phonetic classes V.S Chou partitioning

Question type	Number of States		
	7000	9000	12000
	Avg. Cost per Frame		
Phonetic classes	3.06	3.03	3.00
Chou partitioning	3.07	3.03	2.99
	% WER		
Phonetic classes	27.5	27.4	27.4
Chou partitioning	27.9	27.3	27.3

Table 1: Comparing triphone systems using hand-crafted phonetic classes with automatic CPA questions. Average cost per frame (smaller indicates better fit) and word error rate are reported for increased number of leaves/states.

上下文特徵和語音類別問題的結合

Phonetic		Word		Syllable
Left	Right	ID	Boundary	ID
50.9%	49.1%			
41.4%	40.6%			18.1%
47.6%	46.6%		5.7%	
40.8%	41.5%	17.7%		
39.4%	39.8%	16.0%	4.8%	

Table 2: Percentage of splits by context feature for different 9000 leaf trees. Each row is a mix of different context features.

使用不同上下文特徵的結果

Context		Number of States		
Phonetic	Non-phonetic	7000	9000	12000
Triphone	—	27.5	27.4	27.4
	Syllable ID	27.4	27.4	27.0
	Word Boundary	26.9	27.3	26.9
	Word ID	26.9	27.1	27.0
	Word Boundary + ID	26.9	26.9	26.5

Table 3: Performance results when using different context features (% WER) corresponding to the rows in Table 2.

Triphone和系統其他的結合的比較

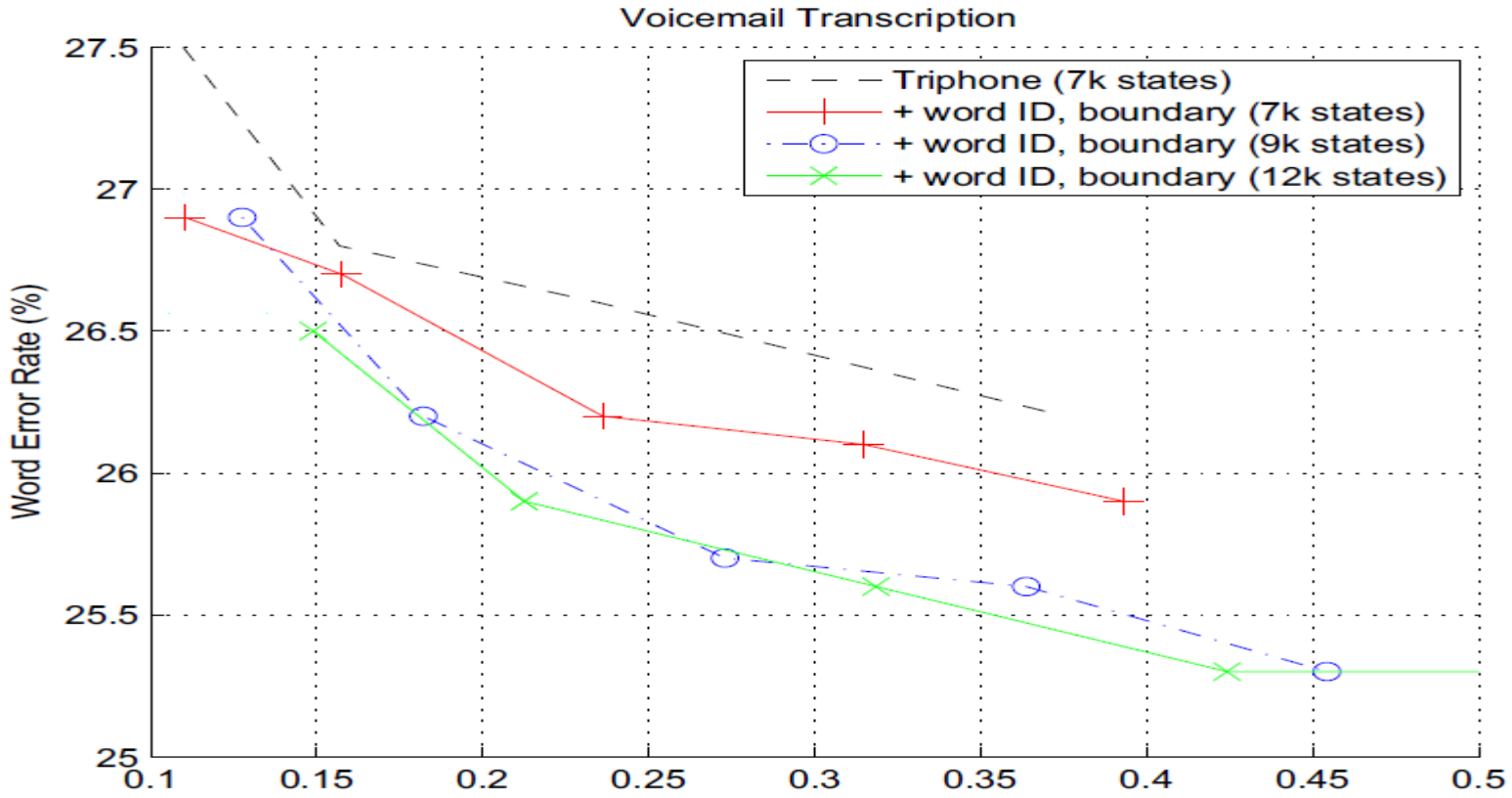

Figure 2: Comparing baseline triphone system with various combined triphone/word/word boundary systems, increasing number of system Gaussians

Search by Voice Task

- The voice search task 請參閱 www.google.com/mobile
- 實驗將額外增加的上下文特徵表現在大量的訓練資料上
- 語言模型是一個backoff trigram model包含14M Ngrams和1M單字
- 測試集合包含14k的單字，每一個語音長度大約3秒左右

每個上下文分裂的數量

Train Set	Context	Number of States		
		7000	9000	12000
420hr	Left phone	44.9%	43.7%	—
	Right phone	39.4%	38.3%	—
	Word ID	10.0%	12.7%	—
	Word boundary	5.7%	5.3%	—
2100hr	Left phone	44.0%	43.4%	42.9%
	Right phone	38.3%	36.9%	36.6%
	Word ID	11.8%	14.3%	16.0%
	Word boundary	5.9%	5.4%	4.4%

Table 4: Percentage of splits of each context feature, varying the number of states and amount of training data.

Triphone和系統其他的結合的比較

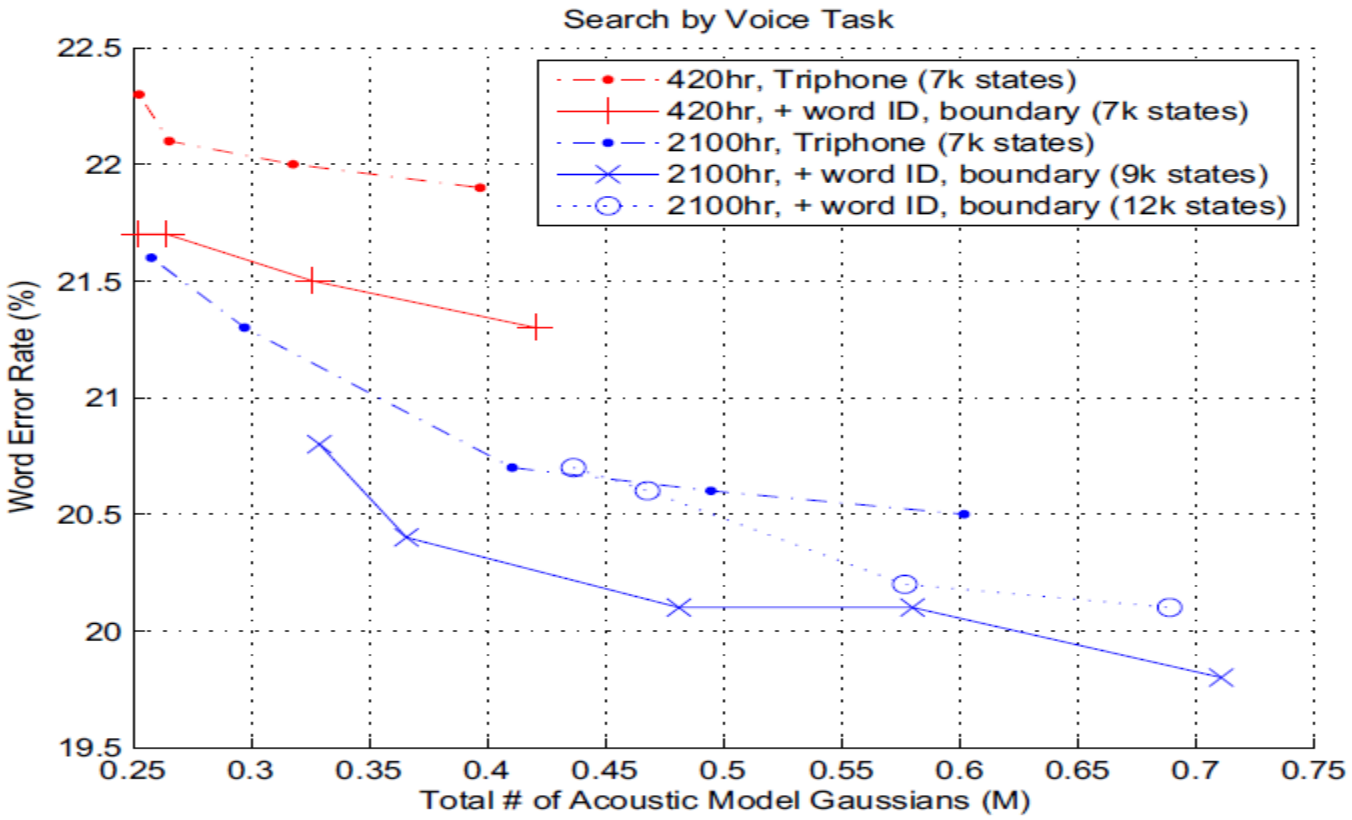

Figure 3: Comparing baseline triphone system with various combined triphone/word/word boundary systems, varying training set size and increasing number of system Gaussians.

結語

- 這篇paper探討新的上下文特徵的使用，以上下文相依的phone model的單字和音節爲基礎，去結合標準的特徵，例如:word boundary 和 triphonic context