Direct Construction of Compact Context-Dependency Transducers From Data

Author: David Rybach, Michael Riley

Report: 黄予宏
Professor: 陳嘉平 教授

Abstract

This paper describes a new method for building compact context-dependency transducers for finite-state transducer-based ASR decoders. Instead of the conventional phonetic decisiontree growing followed by FST compilation, this approach incorporates the phonetic context splitting directly into the transducer construction. The objective function of the split optimization is augmented with a regularization term that measures the number of transducer states introduced by a split. We give results on a large spoken-query task for various n-phone orders and other phonetic features that show this method can greatly reduce the size of the resulting context-dependency transducer with no significant impact on recognition accuracy. This permits using context sizes and features that might otherwise be unmanageable.

Index Terms: WFST, LVCS

Introduction

轉換器主要由C。L。G三部分組成

C： 上下文獨立(CI)-上下文相關(CD)轉換器

L：根據發音字典的 文字-音素 轉換器

G：語言模型

本文專注在C的建構上

Introduction

最直接建造轉換器的方法：

使用p^{n-1}個狀態與p^{n}個轉換來表達

=>對於較大文字量或加入其他關係時會顯得笨拙

如：考慮word boundary、母音、重音等

若訓練資料不足，常使用發音決策樹來tie一些發音

產生緊密的轉換器。

Introduction

有很多文獻都提出有效率的方法來建造轉換器

決策樹是用來建立FST最常見的方法

但只用決策樹的方法還是很容易造成狀態數過多

本文改變了決策樹建立的部分

修改了tie的演算法來控制狀態數

並加了參數α 來權衡精確度及轉換器大小

Decision-Tree Construction

決策樹常用來把一些音tie在一起

一般決策樹分群法是用音的特性或者位置來建樹

找最相似的音素分在一群

本文也用上述方法，並加上了分裂時的評比分數

依據此分數來決定要分裂哪個狀態

算法如下

Decision-Tree Construction

$L(t) = G(t) - \alpha \cdot S(t)$

G(t): 分裂對acoustic likelihood的增加值

S(t): 分裂需要增加多少狀態來區分差異

α : 用於權衡的變數(0~∞)

其中α = 0 表示與普通決策樹方法無異

α = ∞表示忽略模型的聲學特徵

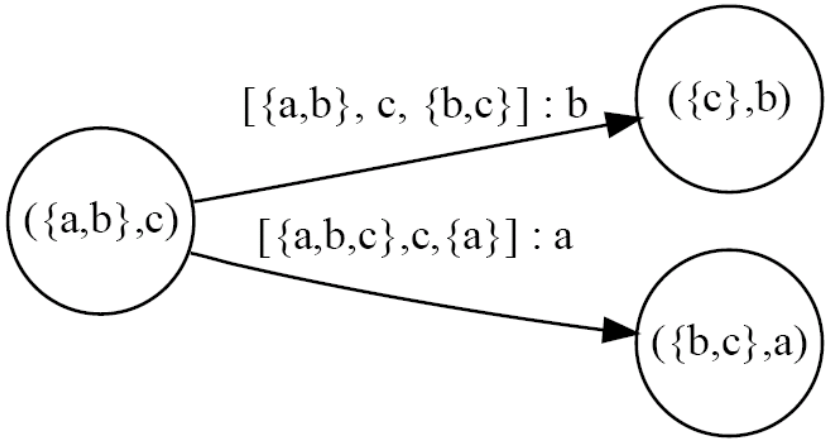
Transducer Construction

本文使用音素 π 向左L、向右一音素的L+2音素模型

下圖是以音素 c 而言L = 1的三音素模型

狀態內代表到此狀態時的字尾集合

線上代表input label(CD模型)與output label(CI音素)



Transducer Construction

符號定義

T = (A, B, Q, I, F, E)

T: 有限狀態轉換器

A/B: 輸入/輸出字母

Q/I/F: 有限/初始/結束狀態集合

E: 轉換路徑

E[q]表示離開狀態q的轉換路徑集合

I[q]表示進入狀態q的轉換路徑集合

p[e]/n[e]代表轉換路徑e的原本/下個狀態

i[e]/o[e]代表轉換路徑e的輸入/輸出標籤

E(a)代表一個轉換路徑輸入標籤爲a，定義成E(a) = {e : i[e] = a}

前個狀態Q(a) = {p[e] : e屬於E(a)}

Transducer Construction

以三音素模型爲例，當要分裂一個音素模型時

需要找出受影響的狀態，對狀態做改變

若是分裂右音素時

只需要更新出此狀態之轉換線的input label

$$i[e] \leftarrow \begin{cases} m_1 & \text{if } i[e] = m, o[e] \in C_1' \\ m_2 & \text{if } i[e] = m, o[e] \in C_1'' \\ i[e] & \text{if } i[e] \neq m \end{cases}$$

Transducer Construction

若是分裂左音素時

則可能需要製造新狀態(若原本不存在)

而狀態線需要做移動或增加的動作

原先的入狀態轉換線依它的output label

決定要進入哪一個狀態

Transducer Construction

而出狀態轉換線則要個別做更新或增加(若沒有)

根據分裂的各個狀態做狀態線更改

$$e_j = (q_j, i_j, o[e_j], n[e_j]) \in E[q_j],\ j \in \{1, 2\}$$

$$i_j \leftarrow \begin{cases} m_j & \text{if } i[e_j] = m \\ i[e_j] & \text{if } i[e_j] \neq m \end{cases}$$

自我轉換線在這篇文章不考慮

Transducer Construction

依此方法可以往前推到左邊L個音素

用遞迴的方法，當狀態q做分裂時

其前任狀態p存在一個轉換線連結到q

那p也必須做分裂，一直分裂到開頭爲止

Transducer Construction

雖然此篇文章不證明

向右分裂不增加狀態

向左分裂增加的狀態數爲極小

計算最後狀態數的增加會是極小值

以S(t)代表整個轉換器的特徵

Transducer Construction

還有更多特徵可以加入音素模型中

像是word boundary資訊、音節、語者性別等

word boundary就是標示一個音素出現在字的位置

做法是改變音素的表示方式並修改發音字典

爲了控制音素模型的數量可以用新音素性質表示

如(前端、中端、末端音素)

Experimental Results

以一個大量詞彙連續語音辨識(LVCSR)

的語音查詢任務做實驗

一個訓練2100小時語音查詢的baseline聲學模型

對應每個轉換器用bootstrap model

重新訓練聲學模型

用感知線性預測係數(PLP)做前端處理

Experimental Results

用線性識別分析(LDA)轉換

投影9個連續13維特徵到一個39維特徵向量

tied 隱馬可夫模型狀態模型由最多128高斯密度

per mixture model with semi-tied covariance 構成

total model density in the acoustic model ranges

between 400k~500k

Experimental Results

發音字典包含43個音素

更進一步的訓練步驟省略

實驗用簡單的一通解法方法

With a backoff 3-gram 語言模型包含14M n-gram for a vocabulary of 1M words.

測試資料有14.6K個語段內含46K個字

Experimental Results

Table 1: Construction of n-phone models with different values for α. The table shows the number of HMMs, the number of HMM state models (distributions), the size of the C transducer, and the achieved word error rate. The first row for each n-phone order is for the conventional decision-tree-based construction.

		acoustic model		C	
n	α	HMM	dist.	states	WER
3	-	17,066	6,623	1,849	21.2
	0	17,086	6,623	1,056	21.3
	100	16,953	6,623	1,032	21.1
	10^3	16,280	6,619	938	21.2
	10^5	6,846	6,543	722	21.4
4	-	-	-	79,507	-
	0	51,782	8,273	18,951	21.6
	100	43,890	8,257	9,803	21.6
	10^3	33,124	8,263	6,302	21.6
	10^5	9,681	8,144	5,728	21.7
5	-	-	-	3.4M	-
	10^3	22,857	7,000	1,453	21.4

Experimental Results

由表一可以看出使用了本文的方法

可以大量減少狀態數而不影響到錯字率

在語音查詢中，上下文增加並不會影響錯字率

表二可看出增加word boundary訊息也可減少狀態數

Table 2: Results for models with incorporated word boundary information.

n	α	dist.	states	WER
3	0	7,052	12.3k	20.5
	100	6,146	3.0k	20.5
	1k	7,061	2.9k	20.6

Discussion

這篇文章只考慮向右一音素的音素模型

對於向右γ 個音素用此方法還無法做出極小轉換器

不過這篇文章可以建造一個極小自動機

如何在每個步驟建造極小轉換器

是一個非常有趣的問題