The 2019 Conference on Computational Linguistics and Speech Processing
ROCLING 2019, pp. 35-46

室內遠距離語音辨識實驗

Experiments on In-House Far-Field Speech Recognition

邱炫盛 Hsuan-Sheng Chiu, 楊智合 Jyh-Her Yang

中華電信研究院
Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taiwan

{samhschiu, houseyang0204}@cht.com.tw

摘要

近年來，語音辨識的應用推出各種遠距離操作的系統，例如車用語音助理、智慧音箱等，在這些系統中遠距離語音辨識扮演著關鍵的角色。本文主要提出我們在智慧音箱裝置上的遠距離語音辨識相關實驗及成果，我們利用資料擴充方式、模擬遠距離語音及基於類神經網路聲學模型來降低字元錯誤率。在實驗部分，本文利用智慧音箱錄製了三種距離的平行測試語料，其中 50cm 情境語料可從 13.31% 降至 8.41%，相對改善 36.8%，而 80cm 情境語料從 19.20%降至 10.89%，相對改善 43.2%。

Abstract

In recent years, speech recognition applications have introduced a variety of remote operating systems, such as car voice assistants, smart speakers, etc. In these systems, far-field speech recognition plays a key role. This paper mainly presents our experiments and results on far-field speech recognition on smart speaker devices. We use data augmentation methods, simulated far-field speech and neural network-based acoustic models to reduce the character error rate (CER). In the experimental part, this paper recorded the parallel test corpus of three distances using the smart speaker. The 50cm situation corpus can be reduced from 13.31% to 8.41%, the relative improvement is 36.8%, and the 80cm situation corpus is reduced from 19.20% to 10.89% with relative improvement of 43.2%.

關鍵詞：遠距離語音辨識，智慧音箱，類神經網路聲學模型，資料擴充

Keywords: far-field, smart speaker, neural network-based acoustic models, data augmentation

一、緒論

近年來，在引進深度類神經網絡(Deep Neural Network, DNN)於聲學模型訓練之後，自動語音辨識方面就取得了重大進展[1,2,3]，對於任一語音辨識領域，當提供足夠且具代表性的訓練語料時，DNN 就可以學習其聲學本質上的變異性，如：語者、性別、頻寬、環境等差異。然而，在一個真實室內的空間內，可以想像到遠距離語音辨識能夠讓我們的生活更佳便利，但是遠距離語音辨識仍然是一個具有挑戰性的問題[4]。

目前已經提出許多技術[4,5,6]來處理遠距離語音辨識問題，其中最有效的作法就是資料擴充(Data Augmentation)，它讓 DNN 有機會可以學習到真實環境中可能遭遇到的情況，只要 DNN 能夠學習到真實環境的聲學特徵，在訓練及測試條件匹配之下就會有不錯的效果。而收集大量現實各種樣式的環境語料很耗費大量人力及金錢，因此，利用模擬方法產生訓練語料是一種可行的選擇，藉由模擬各種樣式環境語料進行資料擴充，對於強健聲學模型上能得到非常顯著的效果，可從 IARPA-ASPIRE 遠距離辨識競賽[7,8]中看到使用資料擴充方法得到最大相對 33%的改善效果。

本文主要分享利用前述作法於室內智慧音箱上的遠距離語音辨識實驗，使用模擬的空間脈衝響應(Room Impulse Response, RIR) 捲積於語音信號中來表示一個空間的殘響(Reverberation)，藉由調整不同參數可以改善智慧音箱上的遠距離語音辨識問題。

本文將在接下來的第二節會介紹本實驗所使用的方法，在第三節將會描述本次實驗在真實音箱於不同距離下的結果，最後結論會放在第四節作說明。

二、實驗方法

（一）模擬空間調整

在 Ko[6]的論文中，主要透過鏡像法產生了 3 種 RIR，其產生方式為先依照主要距離限制，例如長寬 1~10 公尺內，高 2~5 公尺內，亂數產生 200 組空間參數，這些參數包含了空間的長寬高、接收端位置與空間的吸收係數，再限制聲音來源端與接收端距離為 0 到 5 公尺，亂數產生 100 組來源端位置。最後每一種 RIR 各有 20000 組模擬結果。另外，Ko 也整理了實際空間錄製的 RIR，共 325 筆來進行相關實驗，所以總共有 4 種類型的 RIR，此資料集稱為 Room Impulse Response and Noise Database（以下稱 RIRND），

表一、RIRND 資料集

參數	設定值
RIR_small	空間長寬 1~10 公尺，高度 2~5 公尺，吸收係數 0.2~0.8，聲音來源與接收位置距離為 5 公尺內，亂數產生共 20000 組
RIR_medium	空間長寬 10~30 公尺，高度 2~5 公尺，吸收係數 0.2~0.8，聲音來源與接收位置距離為 5 公尺內，亂數產生共 20000 組
RIR_large	空間長寬 30~50 公尺，高度 2~5 公尺，吸收係數 0.2~0.8，聲音來源與接收位置距離為 5 公尺內，亂數產生共 20000 組
RIR_real	由三套資料集組成： RWCP sound scene database: 1 個實際空間，長度為 6.66 公尺，寬度為 4.18 公尺，聲音來源與接收距離為 2 公尺，響應時間為 0.3~1.3 秒，共 182 組 REVERB challenge database: 3 個模擬空間，響應時間分別為 0.25 秒、0.5 秒與 0.7 秒，2 種聲音來源接收位置距離分別為 0.5 公尺與 2 公尺；1 個實際空間，響應時間 0.7 秒，2 種聲音來源接收位置距離分別為 1 公尺與 2.5 公尺，共 36 組 Aachen impulse response database: 4 個實際空間，空間分別為 3 x 1.8 x 0.5、5 x 6.4 x 2.9、8 x 5 x 3.1、10.8 x 10.9 x 3.15 公尺，聲音來源接收距離分別為 0.5, 1, 1.5 公尺、1, 2, 3 公尺、1.45, 1.7, 1.9, 2.25, 2.8 公尺、4, 5.56, 7.1, 8.68, 10.2 公尺，共 107 組

詳細資料如表一所示。然而，我們主要的應用是在家裡的客廳內使用智慧音箱，使用 RIRND 資料集未必完全符合我們的需求，所以我們嘗試設計以客廳為空間大小的參數以符合使用情境。我們先依照客廳的可能範圍亂數產生空間大小 200 組，再亂數產生 100 組聲音來源端（即說話者）及接收端（即音箱）位置與殘響時間等參數，同樣總共有 20000 組結果（以下稱 RIR_exp）。說話者與音箱距離的產生方式是先將空間平面劃分成 5 x 5 的區塊，由左上至右下編號為 0 至 24 號，並假設音箱可能放在電視旁邊（編號 1、3）、客廳中間桌子（編號 12）或是空間四邊角落（編號 0、4、20、24），其他編號為可能的說話者位置。至於其他參數如反射次數， RIRND 設定為 10，我們則不限制；收音方式我們與 RIRND 相同假設為全指向性；殘響時間參數部分，RIRND 是使用吸收係數，且假設空間平面皆相同，其殘響時間則可透過 Sabine 公式換算得出，簡單地說，吸收係數越高，殘響時間越短，我們則是參考 REVERB 2014 設定為 0.25~0.7，我們設定為 0.2~0.6。表二是我們的參數詳細設定，我們亦進一步統計，以我們的方法產生出的模擬結果，其說話者與音箱的距離大部分會分佈在 2~3 公尺的距離，其次則是 1~2 公尺及 3~4 公尺，如表三所示。

表二、空間脈衝響應模擬參數設定

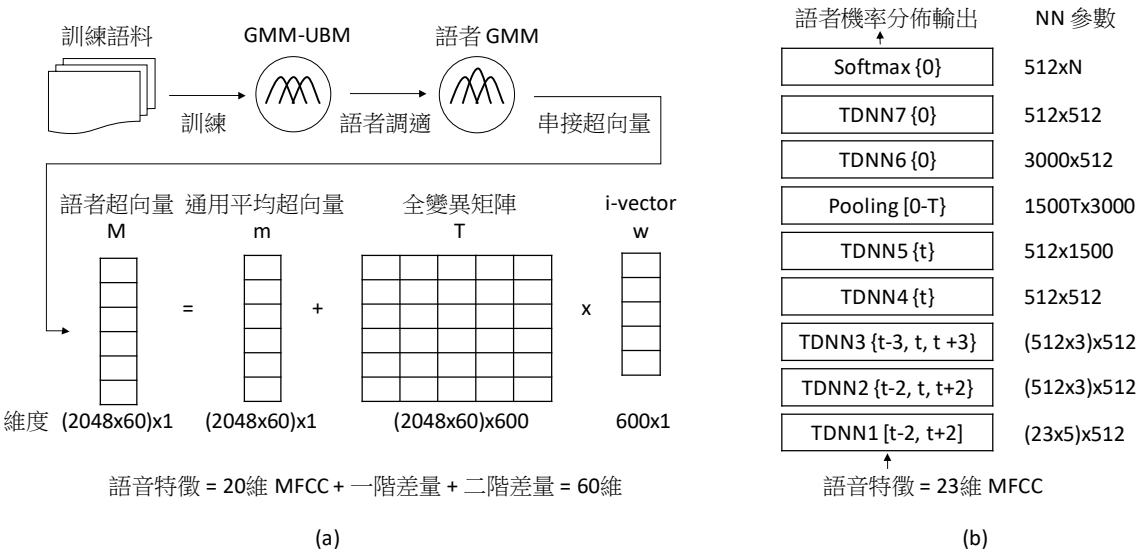
參數	設定值
空間長度 (公尺)	[3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0]
空間寬度 (公尺)	[3.0, 3.4, 3.8, 4.2, 4.6, 5.0]
空間高度 (公尺)	[2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.2]
說話者高度 (公尺)	[0.9, 1.1, 1.3, 1.5, 1.7]
音箱高度 (公尺)	[0.4, 0.6, 0.8, 1.0, 1.2, 1.4]
音速 (公尺/秒)	340
接收端指向性	全指向
響應時間 (T60)	[0.2, 0.3, 0.4, 0.5,0.6]
反射次數	無限制
說話者位置編號	[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
音箱位置編號	[0, 1, 3, 4, 12, 20, 24]

表三、說話者與音箱距離分佈

說話者與音箱距離	筆數	分佈 (%)
0~1 公尺	1871	9.355
1~2 公尺	5869	29.345
2~3 公尺	6774	33.870
3~4 公尺	4347	21.735
4~5 公尺	1139	5.695

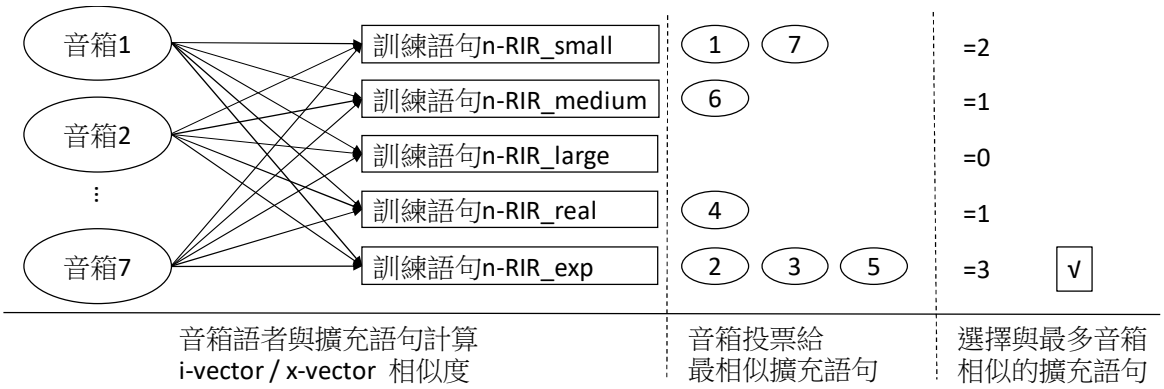
（二）訓練資料選擇

我們已經透過模擬 RIR 方式來產生遠距離的訊號殘響，然而對於我們目標的客廳環境音箱辨識仍不一定一致，尤其原始資料大多來自於手機或麥克風。如我們所知，如果選擇後的訓練資料與測試環境或設備來源一致，與使用全部資料相比，不僅能加快訓練速度，甚至可能得到相同或更好的結果。聲音資料選擇的方法有許多種，大致上可分成依據語言文字相關性，或是依據聲學特性相關性來決定[9]。於本論文中，我們初步嘗試使用 i-vector 及 x-vector 進行資料選擇。I-vector 是近幾年內語者辨識技術上標準的先進方法，其主要是透過以 GMM-UBM (Gaussian Mixture Model-Universal Background Model) 建立訓練語者的超向量(Supervector)，再進一步訓練全變異(Total Variability)矩陣。不同於 JFA (Joint Factor Analysis)分別對語者及通道建模，這個矩陣會同時包含語者與通道的資訊，並使用此矩陣建立目標語者與測試語句的 i-vector，最後再透過 LDA (Linear Discriminant Analysis)降維與 PLDA (Probabilistic Linear Discriminant Analysis)計算相似



圖一、(a) i-vector (b) x-vector 示意圖

度。X-vector 則是最近被提出用在語者辨識上有更佳的效果[10]。X-vector 主要是在 DNN 裡加入時間池化(Temporal Pooling)層，讓音框(Frame)層次資訊轉換成音段(Segment)層次資訊，訓練完成後，輸入語音直接使用池化層的下一層網路輸出當作 x-vector，並可以沿用 LDA 與 PLDA 進行相似度計算；x-vector 也可再透過模型架構調整引用更多資訊，例如音素資訊[11]。此外，i-vector 與 x-vector 皆可以透過資料擴充方式來提升準確度。圖一是 i-vector 與 x-vector 的模型示意圖，i-vector 主要是以 GMM-UBM、特徵空間與 EM 演算法進行訓練，而 x-vector 則是以 NN 模型為訓練架構。除了語者辨識，Siohan [12]也將 i-vector 用來作資料選擇。Siohan 把目標資料集的 i-vector 分佈當作其特性，並透過一句或多句批次的方式，分別計算批次加入前與加入後與目標資料集分佈的 KL 距離，如果距離縮小，則加入，反之則不加入。而我們初步的作法則是以分類與投票方式，先使用原始資料訓練 i-vector 及 x-vector，再準備一部分音箱語料當作發展集，並對發展集與所有擴充資料抽取 i-vector 或 x-vector 特徵後，計算其相似度，再針對每一句訓練語句，選擇與發展集語者有最多且最大相似度的擴充版本加入。訓練資料選擇流程如圖二所示，假設音箱為 7 個，訓練語句 n-RIR_xxx 為第 n 句訓練語句經過 xxx 方式處理的擴充版本。i-vector 與 x-vector 兩者處理語者的聲學特徵方式截然不同，參數量也有很大的差異，雖然主要都是用來區分語者，但是以我們的資料選擇方式來說，對於同一句訓練語句使用不同的擴充方式都是相同語者，主要是希望透過這種方式，嘗試選出與音箱特性最相似的擴充語句。



圖二、訓練語料選擇流程示意圖

（三）實驗設定

我們的訓練語料來自商業採購、專案語料以及網路上各類開源語料的集合，共有 2461 小時（以下稱 STT2461），其中來源設備包含了麥克風、手機與廣播等裝置；錄音樣式包含了腳本錄音與口語對話；情境包含了新聞、各類命令、各類書籍語句念稿與主題式聊天等方式；語言則包含了中文、英文與台語三種語言，其中以中文為主，約佔 76.8%，其餘為 18.8%及 4.4%。我們另外從訓練語料中取出特定資料集，共 507 小時進行實驗（以下稱 K360）。測試語料則是主要來自不同廠商的音箱設備，並包含在辦公室環境錄製的腳本錄音以及用戶實際環境體驗測試。錄製腳本的應用領域包含了各種資訊查詢，如電視頻道、節目、電影、歌曲、廣播、有聲書、股票、店家、天氣、路況等，並搭配多種前綴語或後綴語，如「我要看 OO 台」、「我想聽 OO 的 OO」、「OO 股價多少」、「最近的 OO 在哪裡」、「OO 有沒有下雨」等；或是語音命令，如設定鬧鐘、行事曆、訂車票、設備控制，如「設定 O 點 O 分的鬧鐘」、「打開客廳冷氣」等。其他資訊如表四所示，其中除了 IBPH3 是包含麥克風與不同距離音箱對齊的語料外，其他則都是不同廠商的音箱錄音語料。

我們使用 Kaldi 工具[13]進行相關實驗，採用的聲學模型架構為 TDNN-F[14]，網路層數為 11 層，每一層維度為 1280 維，SVD 分解維度為 256 維，模型架構主要參考 kaldi/egs/swbd/s5c/local/chain/tuning/run_tdnn_7n.sh，語音特徵使用 40 維 MFCC、3 維 PITCH 及 100 維 i-vector 進行訓練。進行實驗時，原始模型的 epoch 設為 4，加入擴充資料時則設為 2。進行訓練資料選擇實驗也是使用 Kaldi 抽取 i-vector 與 x-vector，我們

表四、發展與測試語料集

編號	IBPH3	IBD00	IBQC5K	IBDEV	IB0515
類型	測試	測試	測試	發展	測試
句數	2056	15546	5047	2960	7011
小時數	1.21	11.44	4.56	2.77	6.46
平均秒數	2.13	2.65	3.25	3.38	3.32
收音裝置	麥克風/音箱	音箱	音箱	音箱	音箱
環境	辦公室 20/50/80 公分	辦公室 80 公分	實際環境	實際環境	實際環境
類型	腳本錄音	腳本錄音	腳本錄音	實際用戶	實際用戶
音箱廠商	A	B	B: 1512 句 C: 3535 句	B	B: 1623 句 C: 3226 句 D: 2162 句

參考 kaldi/egs/sre16/v1 與 kaldi/egs/sre16/v2 設定，i-vector 為 600 維，x-vector 為 512 維，但皆不做資料擴充步驟。方法實驗的聲學模型語料使用的是 K360 訓練集與 IBDEV 發展集。語言模型部分，我們使用以中文網頁跟新聞為主的 3-gram 背景模型，再將各種應用情境訓練語料整合訓練出一個應用導向的 5-gram 語言模型，最後透過內插法將兩個模型合併，初步設定背景模型比重為 0.3。此外，詞典大小為 102 萬詞，腳本測試集語言複雜度(Perplexity)約為 1087，實際用戶測試集約為 1388。然而，由於詞彙量大，測試句之間的複雜度差異也很大，從數十到上萬皆有。

三、實驗結果

（一）模擬空間調整結果

我們首先測試加入不同 RIR 進行訓練的效果，其中實驗結果皆以字錯誤率(Character Error Rate, CER)為評估標準，其結果如表五、表六所示。由於商業因素，實際用戶的測試結果會用正規化後的錯誤率表示，即是將錯誤率除以基準錯誤率。首先，顯而易見地，當距離越遠，其辨識效果明顯降低，如 IBPH3 未加入擴充資料的 CER 從 5.69%增加至 19.20%；而如果比較未加入擴充資料與加入後的效果，皆能有所提升，且遠距離的改善更明顯，例如 IBPH3-50cm 可從 13.31% 降至 8.41%，相對改善 36.8%，IBPH3-80cm 可

表五、空間模擬測試 CER 結果 1

K360 加入不同 RIR	IBPH3-mic	IBPH3-20cm	IBPH3-50cm	IBPH3-80cm
NO_RIR	5.69	7.49	13.31	19.20
RIR_small	4.65	6.45	8.50	10.98
RIR_medium	4.78	6.24	8.83	11.79
RIR_large	4.99	6.29	9.64	12.27
RIR_real	5.02	5.99	8.78	11.67
RIR_exp	4.92	5.87	8.41	10.89

表六、空間模擬測試 CER 結果 2

K360 加入不同 RIR	IBD00	IBQC5K	IBDEV	IB0515
NO_RIR	10.53	11.34	1.000	1.000
RIR_small	8.57	8.44	0.863	0.982
RIR_medium	8.53	8.43	0.903	0.974
RIR_large	8.91	8.89	0.909	0.976
RIR_real	8.65	9.05	0.919	0.961
RIR_exp	8.52	8.49	0.928	0.990

從 19.20%降至 10.89%，相對改善 43.2%。而如果我們觀察自行模擬的 RIR_exp，在大部分的測試上都能有改善，但令人意外的是實際環境加上實際用戶的發展與測試集，雖然有改善，但是改善幅度並不大。然而，使用其他類型的 RIR 也是類似情況，觀察結果其主要原因是實際環境除了有遠距離收音問題外，還有背景雜訊干擾如電視聲的問題以及語言模型涵蓋率不夠，如新歌曲、新電影等 OOV(Out-of-Vocabulary)的影響。

（二）訓練資料選擇結果

接著我們呈現訓練資料選擇後的實驗結果，如表七、表八所示。可以觀察到，不管是使用 i-vector 或是 x-vector，效果大部分都會比亂數選擇來得好；而使用 x-vector 與 i-vector 則是互有高低，但差異不大。然而，如果與表五、表六使用單一 RIR 的情況相比，資料選擇後只有在 IBPH3-50cm 的情況下有改善，其他則是沒有改善。我們嘗試統計資料選擇後的 RIR 分佈，如表九所示，亂數是平均分佈，i-vector 以 RIR_exp 與 RIR_real 的比例為最多，x-vector 則是以 RIR_exp 與 RIR_small 為最多。如果 i-vector 與 x-vector 能反應出語音的語者或聲學特性，根據這樣的分佈來看，表示以 RIR_exp 擴充的語料應該是與實際環境最相似；若以 IBDEV 發展集與 IB0515 測試集的單一 RIR 結果來看，的確以 RIR_small 與 RIR_real 擴充的效果最好。資料選擇的效果不如預期的原因或許是因為我們初步資料選擇後僅是讓不同 RIR 之間的混合比例不同，雖然這樣可以確保與單一

表七、訓練資料選擇 CER 結果 1

K360 加入不同 RIR	IBPH3-mic	IBPH3-20cm	IBPH3-50cm	IBPH3-80cm
NO_RIR	5.69	7.49	13.31	19.20
RIR_rand	5.25	6.15	8.67	11.35
RIR_ivector	4.90	6.12	8.41	11.32
RIR_xvector	4.99	6.10	7.95	11.28

表八、訓練資料選擇 CER 結果 2

K360 加入不同 RIR	IBD00	IBQC5K	IBDEV	IB0515
NO_RIR	10.53	11.34	1.000	1.000
RIR_rand	8.54	8.83	0.926	0.967
RIR_ivector	8.55	8.44	0.907	0.983
RIR_xvector	8.60	8.46	0.902	0.978

表九、訓練資料選擇比例分佈

混合 RIR	small	medium	large	real	exp
random	20.01%	19.93%	20.07%	19.92%	20.07%
i-vector	20.16%	12.10%	19.66%	22.43%	25.65%
x-vector	29.87%	10.62%	2.98%	11.51%	45.02%

RIR 的語料數量大小一致，但並沒有使用相關性分數來選擇出最相似或是篩選掉不相似的資料，反而不如使用單一 RIR 的效果，未來可嘗試調整成將所有擴充資料依分數排序後再選擇出不同數量進行實驗。

（三）增加訓練語料結果

最後我們呈現使用全部語料 STT2461 進行訓練的結果，如表十、表十一所示。在這個結果中，訓練的 epoch 設為 6，其他的模型設定與方法實驗相同。除了使用更多語料外，我們也加入了加減速以及加雜訊兩種語料擴充方法，加速與減速分別為 1.1 倍與 0.9 倍，雜訊則使用 MUSAN 資料集[15]，包含一般噪音、音樂及人聲，SNR 分別為 0~15dB、5~15dB 及 13~20dB，同時我們也呈現了 Google 辨識結果。首先，我們以經過加減速與雜訊的結果為基準，累積加入不同 RIR 的訓練，雖然訓練時間增加很多，但在每一個測試集都有改善，讓模型更強健。我們的模型效果在腳本錄音的表現比 Google 好，例如 IBD00 與 IBQC5K 的結果，主要是因為較多特別領域詞彙，例如頻道名稱、節目名稱或歌曲名稱等，透過語言模型的調整，準確率更高。但是在實際環境與實際用戶下，例如 IBDEV 與 IB0515 的結果，Google 表現較好，除了實際用戶比較少說出腳本中的特殊詞彙之外，我們也觀察到主要是我們的模型較無法正確地拒絕背景人聲干擾，導致插入錯

表十、增加語料測試 CER 結果 1

STT2461	IBPH3-mic	IBPH3-20cm	IBPH3-50cm	IBPH3-80cm
Base + sp + noise	4.85	8.46	11.00	17.82
+RIR_small + RIR_medium	4.60	6.79	8.85	13.45
+RIR_large + RIR_real	4.65	6.79	8.41	11.67
+RIR_exp	4.62	6.82	8.11	11.51
GOOGLE	12.62	15.95	19.55	24.24

表十一、增加語料測試 CER 結果 2

STT2461	IBD00	IBQC5K	IBDEV	IB0515
Base + sp + noise	9.04	7.76	0.833	0.884
+RIR_small + RIR_medium	8.02	7.87	0.813	0.871
+RIR_large + RIR_real	7.92	7.65	0.777	0.847
+RIR_exp	7.85	7.26	0.762	0.855
GOOGLE	12.16	8.28	0.667	0.782

表十二、不同廠商音箱測試 CER 結果

IB0515	廠商 B	廠商 C	廠商 D
Base + sp + noise	0.882	0.907	0.835
+RIR_small + RIR_medium	0.824	0.916	0.804
+RIR_large + RIR_real	0.800	0.887	0.801
+RIR_exp	0.832	0.898	0.775
GOOGLE	0.939	0.692	0.877

表十三、測試 CER 錯誤分佈

測試-模型	插入	刪除	取代
IBD00-STT2461+RIR_exp	11.95%	22.08%	65.97%
IBD00-GOOGLE	3.21%	65.01%	31.78%
IB0515-STT2461+RIR_exp	25.67%	25.15%	49.18%
IB0515-GOOGLE	1.70%	86.14%	12.16%

誤較多。另外，也觀察到 Google 在音箱錄音品質稍差的情況下，容易出現沒辨識結果的情況，刪除的錯誤較多。我們也呈現不同廠商音箱在實際用戶環境的結果，如表十二所示，各種廠商的音箱在模型加入擴充語料後，皆有改善；廠商 B 與 D 的效果已經能跟 Google 相比，而廠商 C 的效果較差，觀察主要是某些使用者的設備環境常出現背景聲造成插入錯誤，並非設備問題，這也是整體 CER 比 Google 略差的主要原因。我們也列出 CER 錯誤在辦公室腳本錄音與實際環境的分佈情況，如表十三所示，可以看出 Google 的錯誤大部分是刪除錯誤，而從辦公室環境到實際環境，我們的模型插入錯誤比例也從 11.95%增加到 25.67%，但因為實驗並未加上信心度估測的機制，相信加入後能有所改善。

四、結論

我們實驗了音箱在不同距離及實際情境下的表現，並嘗試使用 i-vector 與 x-vector 來進行資料選擇，我們亦嘗試加入更多的訓練資料及其他資料擴充方式來進行訓練。實驗結果說明，不同音箱由於硬體不同，錄製的聲音品質也有差異，但使用更多語料及模擬方式，在各種音箱上也能有所改善。由於目前訓練語料仍是以手機及麥克風為主，未來將嘗試使用轉移學習(Transfer Learning)相關方法[16]，將音箱錄製語料與現有語料進行結合，或是當音箱語料收集足夠後，可以直接用來訓練更符合音箱特性的聲學模型。

參考文獻

[1] A. Mohamed, G. Hinton, and G. Penn, “Understanding how deep belief networks perform acoustic modelling”, in Proc. ICASSP, 2012, pp. 4273–4276.

[2] D. Yu., M. L. Seltzer, J. Li, J.-T. Huang , F. Seide, “Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks”, in Proceedings of International Conference on Learning Representations, May 2013.

[3] V. Peddinti, D. Povey, S. Khudanpur. “A time delay neural network architecture for efficient modeling of long temporal contexts”, In: 16th Annual Conference of the International Speech Communication Association (INTERSPEECH). ISCA, Dresden, 205.

[4] K. Kinoshita, M. Delcroix, S. Gannot, E. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, et al. “A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research”, EURASIP Journal on Advances in Signal Processing, 2016.

[5] T. Yoshioka, T. Nakatani, M. Miyoshi, and H.G. Okuno. “Blind separation and dereverberation of speech mixtures by joint optimization”. IEEE Transactions on Audio, Speech, and Language Processing, 19(1):69–84, 2011.

[6] T Ko, V Peddinti, D Povey, ML Seltzer, S Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition”, in Proc. ICASSP, 2017.

[7] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, “Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms,” in Proceedings of 2015 IEEE

Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015, pp. 539–546.

[8] R. Hsiao, J. Ma, W. Hartmann, M. Karafi, I. Sz, J. Honza, S. Watanabe, Z. Chen, S. H. Mallidi, H. Hermansk et al., "Robust speech recognition in unknown reverberant and noisy conditions," in Proceedings of 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015, pp. 533–538.

[9] KARAFIÁT Martin, VESELÝ Karel, ŽMOLÍKOVÁ Kateřina, DELCROIX Marc, WATANABE Shinji, BURGET Lukáš, ČERNOCKÝ Jan and SZŐKE Igor. "Training Data Augmentation and Data Selection," in New Era for Robust Speech Recognition: Exploiting Deep Learning. Heidelberg: Springer International Publishing, 2017, pp. 245-260.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in Proc. ICASSP, 2018.

[11] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker embedding extraction with phonetic information," in Proc. Interspeech 2018, pp. 2247–2251.

[12] Olivier Siohan and Michiel Bacchiani, "iVector-based acoustic data selection," in Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon, France, 2013.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The kaldi speech recognition toolkit," in IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFLCONF-192584. IEEE Signal Processing Society, 2011.

[14] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in Proc. Interspeech, 2018.

[15] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," arXiv, 2015.

[16] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for ASR using LF-MMI trained neural networks," in Proc. ASRU, 2017