The 33rd Conference on Computational Linguistics and Speech Processing (ROCLING 2021)
Taoyuan, Taiwan, October 15-16, 2021. The Association for Computational Linguistics and Chinese Language Processing

利用少量語碼轉換資料之中英語音辨識系統
Exploiting Low-Resource Code-Switching Data to Mandarin-English Speech Recognition Systems

林厚安 **Hou-An Lin**, 陳嘉平 **Chia-Ping Chen**
國立中山大學資訊工程學系
National Sun Yat-sen University
Department of Computer Science and Engineering
m093040066@nsysu.edu.tw, cpchen@cse.nsysu.edu.tw

摘要

本篇論文中，我們探討如何使用少量的語碼轉換 **(Code-Switching)** 資料來實現語碼轉換語音辨識系統。我們以 **Transformer** 端到端模型開發語碼轉換語音辨識系統，並使用中文資料集加上少量中英混合的課程語音資料集來訓練，作爲本篇論文的基準 **(Baseline)**，接著比較加入多任務學習 **(Multi-task learning)**、遷移學習 **(Transfer learning)** 對於系統效能的差異。實驗結果的部分，以字元錯誤率 **(Character Error Rate, CER)** 作爲評斷系統的標準，最後我們將三個系統分別結合了語言模型 **(Language model,LM)**，最終相比 **baseline** 的 **28.7%** 我們的最好的結果下降到了 **23.9%**。

Abstract

In this paper, we investigate how to use limited code-switching data to implement a code-switching speech recognition system. We utilize the Transformer end-to-end model to develop our code switching speech recognition system, which is trained with the Mandarin dataset and a small amount of Mandarin-English code switching dataset, as the baseline of this paper. Next, we compare the performance of systems after adding multi-task learn ing and transfer learning. Character Error Rate(CER) is adopted as the criterion for the system. Finally, we combined the three systems with the language model, respectively, our best result dropped to 23.9% compared with the baseline of 28.7%.

關鍵字：語碼轉換、語音辨識、語言識別、語言模型、遷移學習、少量資源

Keywords: code-switching, speech recognition, language model, language identification, transfer learning, low-resource

1 緒論

由於 2019 年新型冠狀病毒 (Covid-19) 在世界各地爆發，對各行各業造成衝擊，教育界也是其中之一。在這個大環境之下，各級學校開始採用了遠距離教學的策略，讓學生可以在家學習，而提供課程影片給學生觀看就是一種很受歡迎的遠距離教學的做法。

爲了增加學生觀看影片學習的效率，我們想爲課程影片加上字幕，所以先使用我們的中文語音辨識系統 (Automatic Speech Recognition, ASR) 來辨識影片中語音來生成課程內容的文本，但課程中時常會有少量的英文專有名詞，使得課程很容易出現語碼轉換 (Code-Switching) 的內容，因此本篇論文針對語碼轉換的情況下開發一個中英文語音辨識系統。

在本篇論文中，我們使用 Transformer 端到端 (End-to-End, E2E) 的架構 (Vaswani et al., 2017)(Karita et al., 2019a)，在訓練時使用論文 (Tsunoo et al., 2019) 提出的 contextual block processing 的 Transformer 編碼器，來開發我們的辨識系統。在 inference 時解碼器會使用 Blockwise Synchronous Beam Search 的方法 (Tsunoo et al., 2020)。

實驗的部份，我們以原先用於中文語音辨識系統的資料加入我們現有的少量語碼轉換資料訓練系統，同時以此作爲實驗的比較標準，並嘗試幾種方法來改善我們的系統。由於不同語言的發音方式有變異，因此我們參考論文 (Zeng et al., 2019)(Li et al., 2019)，在訓練階段加入 LID (Language identification) 分類器聯合訓練系統。同時我們擁有的語碼轉換資料有限，所以我們嘗試使用遷移學習來克服這個問題，遷移學習是一種在資料量不足時很有效的做法，我們將原先的中文語音辨識系統當作預訓練模型，再以現有的少量的語碼轉換資料訓練系統。

在剩餘的章節中，章節二會介紹我們所使用的系統架構以及訓練方法，在章節三中我們會介紹我們的實驗設置以及所使用的資料集，在

章節四會描述我們的實驗結果，最後一章會對於整個實驗做結論。

2 實驗方法

首先，我們使用中文資料集並加上課程內容的資料訓練一個語音辨識系統當作實驗的比較基準，同時也嘗試在訓練階段採用多任務學習的方法加入 LID 分類器作爲一個輔助的任務，並用相同的資料集訓練另一個語音辨識系統。

另外，我們以中文資料集先訓練出一個中文語音辨識系統，再應用遷移式學習的技術，加上包含語碼轉換內容的課程資料去微調一個辨識系統。最後對所有系統進行比較以及分析。

2.1 端到端模型

我們的端到端語音辨識模型 (E2E ASR model) 是使用論文 (Tsunoo et al., 2019) 提出使用 contextual embedding 的 Transformer 的架構如圖 1所示。同時以此 Transformer 架構並加入 LID 分類器，架構如圖 2所示，其中 LID 分類器的架構爲圖 2中左半邊的部分。

2.1.1 Transformer 編碼器

首先輸入的音檔會被表示爲一個 80 維的梅爾頻譜圖 (mel-spectrogram) 序列。接著我們會對特徵做降採樣 (Subsampling)，其中降採樣模塊 (Subsampling module) 是由兩層的卷積神網路 (Convolutional neural network, CNN) 組成，其中 kernel size 爲 3，stride 爲 2 還有 256 個 channel 以及 ReLU 的激活函數 (activation function)。我們使用加入 contextual embedding 的 Transformer 編碼器作爲我們的架構，自注意力機制 (self-attention) 能夠使輸入序列的每個位置都能關注到其他任意位置的資訊，以獲取輸入序列的全局信息，架構與原始 Transformer 一樣，架構在圖 1的左半部。

2.1.2 Transformer 解碼器

當 Transformer 的解碼器接收到編碼器的輸出 X_e 跟先前序列的 IDs $Y[1:u] = Y[1], ..., Y[u]$，最後解碼器會計算出輸出序列 $p_{\text{s2s}}(Y|X_e)$ 的後驗機率如下：

$$[p_{\text{s2s}}(Y[2]|Y[1], X_e), ..., p_{\text{s2s}}(Y[u+1]|Y[1:u], X_e)] = \text{softmax}(Z_d W_{\text{att}} + b_{\text{att}})$$

$$p_{\text{s2s}}(Y|X_e) = \prod_u p_{\text{s2s}}(Y[u+1]|Y[1:u], X_e)$$

其中 Z_d 是解碼器的輸出，$W_{\text{att}} \in \mathbb{R}^{d_{\text{att}} \times d_{\text{char}}}, b_{\text{att}} \in \mathbb{R}^{d_{\text{char}}}$ 是可學習的參數，d_{char} 爲字元的數量。Transformer 解碼器的架構爲圖 1的右半部。

2.1.3 語言辨識分類器

由於不同語言在發音方式會有差異，因此我們參考論文 (Zeng et al., 2019)(Li et al., 2019) 的方法，利用 LID 分類器當作一個輔助的任務，幫助我們提昇辨識系統的效能。分別分類輸入爲中文、英文還是語碼轉換的句子。而將此分類器接上原本的 Transformer 架構並使用多任務學習方法訓練系統，在 LID 分類器的損失函數 (Loss function) 是使用 cross entropy 損失函數，LID 分類器的架構如圖 2左半邊所示。

2.1.4 訓練方法

在訓練時我們使用論文 (Tsunoo et al., 2019) 提出的 contextual block processing 的方法來進行訓練，在解碼器的部分採取與原始 Transforemr 解碼器相同的批次 (batch) 訓練。

2.1.5 聯合訓練 CTC 與 Transformer

連續時序性分類 (Connectionist temporal classification, CTC)(Graves et al., 2006) 學習語音特徵與每個字元的對齊，CTC 聯合訓練也有效的加快了學習的速度，可以讓模型快速的收斂 (Kim et al., 2017; Karita et al., 2019b)。在訓練階段，我們採用多任務損失函數 (Multi-task loss)，損失函數結合了來自解碼器和 CTC 的負對數機率 (Kim et al., 2017; Karita et al., 2019b,a; Tsunoo et al., 2020)，損失函數如下所示：

$$L_{\text{mtl}} = -\alpha \log p_{\text{s2s}}(Y|X_e) - (1-\alpha) \log p_{\text{ctc}}(Y|X_e)$$

p_{ctc} 是 CTC 預測的後驗機率，α 是一個超參數，用於調整 CTC 和 S2S 模型之間的比例。

2.1.6 聯合訓練 CTC、LID 分類器與 Transformer

訓練中有新增 LID 分類器架構的系統，損失函數會加上 LID 的負對數機率，損失函數如下：

$$L_{\text{mtl}} = -\alpha \log p_{\text{s2s}}(Y|X_e) - (1-\alpha) \log p_{\text{ctc}}(Y|X_e) - \log p_{\text{lid}}(L|X_e)$$

其中 L 是該筆輸入的語言類別，p_{ctc} 是 CTC 預測的後驗機率，p_{lid} 是 LID 分類器的後驗機率，α 是一個超參數，用於調整 CTC 和 S2S 模型之間的比例。

2.1.7 聯合解碼 CTC、LM

在解碼 (Decoding) 階段，我們簡單的將 S2S、CTC 以及語言模型的機率各取對數後合併起

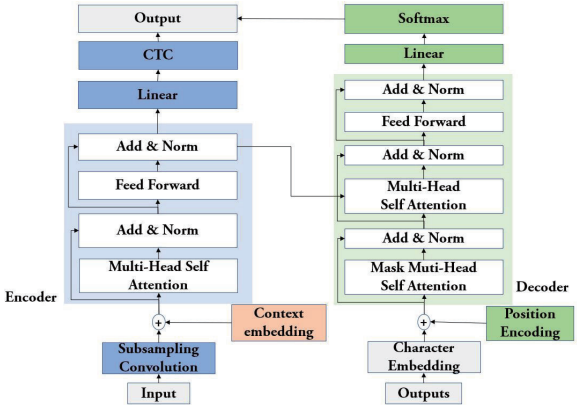

圖 1. 加入 contextual embedding 的 Transformer 架構圖

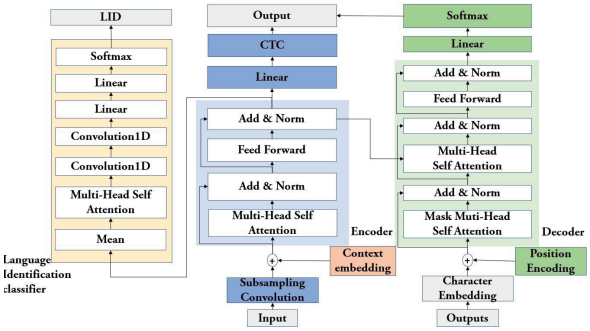

圖 2. 聯合訓練 LID 分類器且加入有 contextual embedding 的 Transformer 架構圖

來，並以 (Tsunoo et al., 2020) 提出的 Block Boundary Detection (BBD) 技術以及 blockwise synchronous beam search algorithm 取代原先的 beam search algorithm，其中 blockwise synchronous beam search algorithm 能夠在使用一定的的 block 數下，就能達到近似一般 beam search 的效果。由於 attention-based 的解碼器經常會出現過早預測出 <eos> 或者預測重複的 token，BBD 主要目的是判斷當前 block 新預測出來的假設是否爲可靠的 (reliable)，若判斷爲不可靠，就會讓解碼器使用編碼器下一個 block 的輸出以繼續解碼。其中 (Tsunoo et al., 2020) 提出的 Block Boundary Detection (BBD) 以及 blockwise synchronous beam search algorithm。我們自表 1的測試集中選取一筆音檔並觀察其搜尋過程，如圖3所示。圖中紅字的部份即爲預測出重複的 token，因此判斷此 block 在這一次新預測出的假設爲不可靠的，進而使解碼器使用編碼器下一個 block 的輸出繼續進行解碼。

資料集	音檔數	總時長 (小時)
Course-train	827	3.03
Course-val	90	0.31
Course-test	389	0.92

表 1. 課程資料集

$$\hat{Y} = \arg\max_{Y \in y^*} \{\lambda \log p_{\text{s2s}}(Y|X_e) + (1-\lambda) \log p_{\text{ctc}}(Y|X_e) + \gamma \log p_{\text{lm}}(Y)\}$$

其中 $p_{\text{lm}}(Y)$ 是Y 的語言模型機率，λ 和 γ 爲超參數，用來調整他們各自所佔的比重，y^* 是一個輸出假設 (output hypotheses) 的集合。

2.1.8 遷移學習

遷移學習 (Transfer learning)(Wang and Zheng, 2015) 是一種被廣泛應用的技術，遷移學習可以將已經學習過的預訓練模型繼承到其他領域來訓練模型，可以省去重新從頭訓練所需要的工作，還可以解決我們在語碼轉換資料不足的問題。由於我們的語碼轉換資料較少，直接訓練可能會造成過度擬合 (Overfitting) 的現象。因此我們使用遷移學習，將先前訓練好的中文語音辨識系統當作預訓練模型，再加上表 1中少量的語碼轉換資料來微調 (fine-tune) 出中英語碼轉換語音辨識系統。

3 實驗設置與資料集

3.1 資料集

本篇論文中在語音辨識模型所使用的訓練集分爲三種，底下逐一說明，並以內容爲含有語碼轉換資料的課程資料集表 1中的 Course-test 資料集來作爲本篇論文的測試集。

3.1.1 課程資料集

此數據集是來自課程影片的資料 [1]，課程爲機率學的內容，其中包含語碼轉換的句子。全部共有 10 堂課程，我們將其中 2 堂課程內容作爲測試集而剩餘 8 堂課程以 90%、10% 的方式分爲訓練集以及驗證集。在表 1顯示了此數據集的資訊。

3.1.2 中文資料集

中文資料集由四個資料集組成，各別資料集的詳細資訊以表 2所示。

(1) NER-Trs-Vol：由國立教育廣播電台提供，其文本爲談話性節目以及新聞報導

[1] https://www.youtube.com/playlist?list=PL_Ks_ZHSKSQ5T2w4gEDCftEmbGNDBj48z

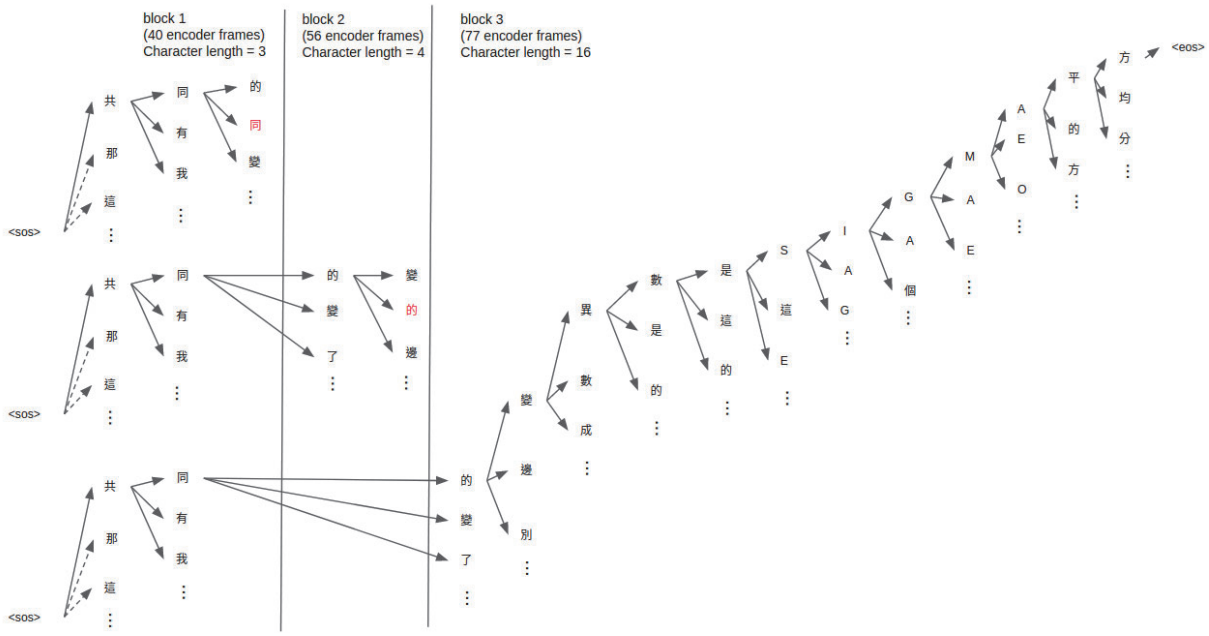

圖 3. blockwise synchronous beam search 演算法下搜尋" 共同的變異數是 SIGMA 平方" 其中 beam size 爲 10

資料集	音檔數	總時長 (小時)
NER-Trs-Vol1	21,089	126.65
AISHELL-1	20,000	24.82
AISHELL-2	20,000	19.87
科技大擂台	24102	50.50
total	85,191	221.84

表 2. 中文資料集

資料集	音檔數	總時長 (小時)
NER-Trs-Vol1	21,089	126.65
AISHELL-1	20,000	24.82
AISHELL-2	20,000	19.87
科技大擂台	24,102	50.50
Course-train	827	3.03
total	86,018	224.87

表 3. 中文與課程資料集

的朗讀式語音，其中總時長爲 126.8 小時，共 21,089 筆數據。

(2) AISHELL-1、AISHELL-2：由 AISHELL 公司提供 (Bu et al., 2017)，分別由 400、1991 位來自中國不同區域的人所錄製，其文本內容包含智能家居、無人駕駛等領域，我們同時將文本內容由簡體中文轉換爲繁體中文，並在兩個資料集中各隨機選取其中 20,000 筆資料加入訓練集。

(3) 科技大擂台 (Formosa Grand Challenge)：由國研院科技政策研究與資訊中心提供，內容爲華語文能力測驗，分爲文章、題目及選項。總時數約爲 400 小時，我們使用問題及選項中的部分資料，總時長約爲 50.5 小時，共 24,102 筆數據。

3.1.3 中文與課程資料集

此訓練集資料我們簡單的將課程資料集表 1中的 Course-train 資料集與中文資料集表 2合併，其資料集資訊如表 3所示。

3.1.4 語言模型資料集

在語言模型所使用的資料集爲內容含有中英文語碼轉換的課程資料集，其資料集資訊如表1所示。文本內容爲每一筆音檔所對應的文字，訓練集、驗證集以及測試集分別有 827、90 以及 389 筆文本，訓練集中 character token 總數爲 39796。

3.2 實驗設置

由於我們的資料量較少，因此使用了速度擾動 (speed perturbation) (Ko et al., 2015) 以及 SpecAugment(Park et al., 2019) 來對資料進行增強。其中，速度擾動會將訓練資料經過 0.9、1.0、1.1 三個參數來產生出三種不同聲音速度的訓練資料來增加訓練資料量，在 SpecAugment 的部分，會直接對頻譜圖進行三種變形，第一是時間扭曲 (Time Warping) 他會在時間方向上進行平移的動作，而其他兩個分別是在時間 (time) 跟頻率 (frequency) 方向上做遮罩 (mask)。礙於硬體資源限制，有

端到端模型	語言模型	CER(%)
Transformer	-	28.7
Transformer + LID	-	27.8
Transformer	√	27.3
Transformer + LID	√	26.5

表 4. 比較 baseline 與新增 LID 分類器架構系統的實驗結果

端到端模型	語言模型	CER(%)
Transformer	-	24.1
Transformer	√	23.9

表 5. 遷移訓練的實驗結果

些許音檔經過速度擾動後使得音檔變長，在訓練階段造成顯示卡 (GPU) 記憶體不足的問題，我們將經過速度擾動後的資料中超過 47 秒的音檔刪除。其中在中文資料集表2以及中文與課程資料集表3在經過速度擾動後都分別刪除了 228 筆音檔，分別佔其資料集中的 0.089% 以及 0.088%。

我們的端到端架構爲 Transformer 的架構是由 12 層編碼器，6 層解碼器組成。在論文 (Tsunoo et al., 2019)(Tsunoo et al., 2020) 中有提出使所有輸入的 frame 有一半重疊的方法。同時會將每個 block 中所有的 frame 分爲三個部份，包括過去 (Past) 已經看過的 frame 、當前 (Current) 使用的 frame 以及未來 (Future) 的 frame，其中過去和未來這兩部份的 frame 是提供給當前的 frame 上下文資訊，而這三個部份的 frame 數分別以 $\{N_l, N_c, N_r\}$ 表示，這部份我們的設置爲 $\{8, 16, 16\}$，而在論文 (Tsunoo et al., 2019) 提出的 contextual embedding，我們初始此 contextual embedding 的方式是將每個 block 中所有的 frame 取平均來作爲初始值，同時使用 position encoding 來幫助分辨 blocks 的序列。其中在多任務學習方法 (multitask learning) 的超參數 α 爲 0.3，decoding 階段的超參數 λ 和 γ 分別爲 0.5 以及 0.3，beam size 大小爲 10。

而 LID 分類器是由一層的 multi-head self attention 以及 2 層的 1D-convolution ，其中 kernel size 爲 3，stride 以及 padding 都是爲 1，最後再通過兩層 linear 所組成。

在語言模型的部份，我們使用 2 層 1024 個神經元的長短期記憶 (Long Short-Term Memory, LSTM) 來建立遞迴神經網路，並以表 1的訓練資料進行訓練。

我們的所有實驗都基於 ASR 工具 ESPnet2(Watanabe et al., 2018) 來開發。

4 實驗結果

由於課程內容多以中文爲主，英文專有名詞爲輔，因此我們以字元錯誤率作爲評估模型的標準。首先使用中文與課程資料集表 3當作訓練集，比較 Transformer 架構以及有加入 LID 分類器後兩者之間的結果，實驗結果如表 4所示。在未加入語言模型的情況下可以看到在加入 LID 分類器後字元錯誤率由 28.7% 下降到了 27.8%，在結果上可以看到加入 LID 分類器對於整體系統效能確實有幫助，再加入我們訓練的語言模型, 兩個實驗的結果皆有明顯的下降, 分別降至 27.3%、26.5%。

接著我們以中文資料集表 2先訓練一個中文語音辨識系統當作預訓練模型，再使用遷移訓練的技術使用課程資料集表 1來進行微調，結果也下降至 24.1%，加入語言模型後, 結果也下降至 23.9%, 實驗結果如表 4所示。

5 結論

在這些實驗中，我們使用多任務學習方法加入了 LID 分類器來提昇系統效能，由於不同語言對於發音方式也不一樣，因此加入此分類器來判別當前語音爲哪種語言或是語碼轉換的聲音訊號，對於系統確實是有幫助的。同時我們使用了遷移學習的技術，基於原先用了較多訓練資料訓練出來的系統當成預訓練模型並只用了極少的語碼轉換資源微調語音辨識系統，實驗結果也有明顯的改善。

在未來，我們也將探討如何改善我們的系統，由於中英語碼轉換的語音資料不易取得。同時以實驗結果來看，語言模型對於系統的確是有幫助，因此我們會先針對語言模型系統進行改進，像是新增更多的語碼轉換的文本資料，且由於我們是以辨識課程爲目的，我們也會加入不同領域常用的英文專有名詞來對語言模型進行改善，以提升我們整體系統效能。

References

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *Oriental COCOSDA 2017*, page Submitted.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, and et al. 2019a. A comparative study on transformer vs rnn in speech applications. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2019b. Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration. In *Proc. Interspeech 2019*, pages 1408–1412.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and S. Khudanpur. 2015. Audio augmentation for speech recognition. In *INTERSPEECH*.

Ke Li, Jinyu Li, Guoli Ye, and Yifan Gong. 2019. Towards code-switching asr for end-to-end ctc models. pages 6076–6080.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*.

Emiru Tsunoo, Yosuke Kashiwagi, Toshiyuki Kumakura, and Shinji Watanabe. 2019. Transformer asr with contextual block processing.

Emiru Tsunoo, Yosuke Kashiwagi, and Shinji Watanabe. 2020. Streaming transformer asr with blockwise synchronous beam search.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Dong Wang and Thomas Fang Zheng. 2015. Transfer learning for speech and language processing.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

Zhiping Zeng, Yerbolat Khassanov, Van Tung Pham, Haihua Xu, Eng Siong Chng, and Haizhou Li. 2019. On the end-to-end solution to mandarin-english code-switching speech recognition.