Article

The XMUSPEECH System for Accented English Automatic Speech Recognition

Fuchuan Tong ¹ , Tao Li ^{2,†} , Dexin Liao ², Shipeng Xia ², Song Li ¹, Qingyang Hong ^{2,*} and Lin Li ^{1,*}

¹ School of Electronic Science and Engineering, Xiamen University, Xiamen 361005, China; tongfuchuan@stu.xmu.edu.cn (F.T.); songli@stu.xmu.edu.cn (S.L.)

² School of Informatics, Xiamen University, Xiamen 361005, China; 31520211154017@stu.xmu.edu.cn (T.L.); 31520191153363@stu.xmu.edu.cn (D.L.); xiashipeng@stu.xmu.edu.cn (S.X.)

* Correspondence: qyhong@xmu.edu.cn (Q.H.); lilin@xmu.edu.cn (L.L.)

† Co-first author: Tao Li.

Abstract: In this paper, we present the XMUSPEECH systems for Track 2 of the Interspeech 2020 Accented English Speech Recognition Challenge (AESRC2020). Track 2 is an Automatic Speech Recognition (ASR) task where the non-native English speakers have various accents, which reduces the accuracy of the ASR system. To solve this problem, we experimented with acoustic models and input features. Furthermore, we trained a TDNN-LSTM language model for lattice rescoring to obtain better results. Compared with our baseline system, we achieved relative word error rate (WER) improvements of 40.7% and 35.7% on the development set and evaluation set, respectively.

Keywords: AESRC2020; i-vector; x-vector; multistream CNN

Citation: Tong, F.; Li, T.; Liao, D.; Xia, S.; Li, S.; Hong, Q.; Li, L. The XMUSPEECH System for Accented English Automatic Speech Recognition. *Appl. Sci.* **2022**, *12*, 1478. <https://doi.org/10.3390/app12031478>

Academic Editor: Yoshinobu Kajikawa

Received: 31 August 2021

Accepted: 27 January 2022

Published: 29 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The standard English ASR system has been able to obtain a high recognition accuracy and meet the commercial requirements of certain scenarios. However, numerous studies have shown that accent affects the accuracy of ASR to a large extent, Feng et al. [1] quantified the bias of a SOTA ASR system, and the result showed the accuracy of the system for native speakers in different regions varies greatly owing to accent. Vergyri et al. [2] showed a well-trained ASR model performed badly on other accents corpus. Due to the inconsistency of the accent itself, the variability of speech speed and phoneme pronunciation, as well as the scarcity of accented speech data, accented English recognition is still a challenging subject. The Interspeech 2020 Accented English Speech Recognition Challenge (AESRC2020) (<https://www.datatang.com/INTERSPEECH2020> (accessed on 20 August 2021)) includes two tracks, Accent Recognition (AR) Track and Accented English ASR Track. This paper describes the XMUSPEECH systems for the Accented English ASR Track, which is a task to train an ASR model while the data is composed of diverse English accents.

In the field of multi-accent ASR, the mainstream approaches are the DNN-HMM model and the end-to-end model. The end-to-end model is a hot topic of current research [3–8]; these studies have made significant contributions to end-to-end pretraining, data augmentation, adversarial learning, unsupervised training, etc. The end-to-end architecture adopts a single network structure, which is simple compared with the traditional hybrid model and can avoid the bias caused by multi-module training, thus ensuring the global optimum to the greatest extent. However, end-to-end training requires a large amount of data, and this single network structure also leads to relatively low interpretability of the model. The DNN-HMM model follows the traditional speech recognition framework, including an acoustic model, language model, and lexicon, which constitute the decoder, and the input acoustic features are converted into text by the decoder. The acoustic model acts to align phonemes and acoustic features, it accounts for most of the computational overhead of the whole system and determines the performance of speech recognition systems. There

is much interest in developing the acoustic model [9–11], specifically, Ahmed et al. [11] proposed a convolutional neural network (CNN)-based architecture, which had variable filter sizes along the frequency band of the audio utterances, and the overall accuracy for accent speech recognition surpassed all of the prior work. Shi et al. [9] used TDNN [12] as an acoustic model for accented English speech recognition and achieved a relatively low average WER. The TDNN [12] is a 1-D convolution neural network, it performs well on temporal tasks such as ASR. On the basis of TDNN [12], Povey et al. proposed [13] TDNN-F, which has fewer parameters and can be trained with a deeper level. Our system adopts TDNN-F [13] as an acoustic model. We have two considerations for adopting this structure. Firstly, the features of the hidden layer of TDNN-F are not only related to the input at the current moment but also with the input at the past and future moments, definitely, the context information is fully modeled. Secondly, the TDNN-F is a 1-D convolution with time translation invariance, for different accents, the same phoneme is pronounced with sounds of different lengths, therefore, the TDNN-F could capture the common features between different accents. We further developed it with multistream CNN and attention mechanism. The idea of multistream CNN [14] comes from the multistream self-attention architecture [15] but without the multi-headed self-attention layers, it processes the input speech frames in multiple streams in parallel, which allows the system to have better robustness. Inspired by speaker recognition, an accent-dependent method is introduced to speech recognition [16–18]. Karafiát et al. analyzed the suitability of x-vectors for ASR adaptation in detail. Furthermore, Turan et al. [17] combined accent embeddings with semi-supervised LF-MMI training to train an acoustic model and achieved almost the same results as the fully supervised approach. In this paper, we extracted four types of representations (i.e., spk-ivector, accent-ivector, spk-xvector, and accent-xvector) and compared their efficiency in the accented ASR system. We found that x-vector embedding performed better than i-vector embedding. Finally, language model rescoring was employed in order to obtain a lower WER. Our best system achieved relative WER improvements of 40.6% on the development set and 35.6% on the evaluation set compared with the baseline.

The rest of this paper is structured as follows. In Section 2, we describe the details of our systems. In Section 3, we provide the experimental setups and discuss the results from various approaches. Finally, we conclude our work in Section 4.

2. System Structure

2.1. Acoustic Modeling

We followed the conventional steps to train hybrid GMM-HMM acoustic models referring to the Kaldi [19] recipe for CHIME6 (https://github.com/kaldi-asr/kaldi/tree/master/egs/chime6/s5_track1 (accessed on 24 August 2021)). It has been shown that sequence-level training criteria such as lattice-free maximum mutual information (LF-MMI) performs better than frame-level criteria for ASR [20]. Our systems are based on the TDNN-F [13] acoustic model using the LF-MMI training criterion. We experimented with various network structures. All of our experiments are based on the Kaldi toolkit.

- **TDNN-F:** The TDNN-F model is the first 11 layers of TDNN-F in the recipe for CHIME6 of the Kaldi (egs/chime6/s5_track1/local/chain/tuning/run_tdnf_1b.sh).
- **CNN-TDNNF-Attention:** The CNN-TDNNF-Attention model consists of one CNN layer followed by 11 time-delay layers and a time-restricted self-attention layer [21], and we applied a SpecAugment [22] layer on top of the architecture to make it more robust. The CNN layer has a kernel size of 3×3 and a filter size of 64. The 11-layer TDNN-F shares the same configuration as the previously illustrated TDNN-F, except it substitutes the first TDNN layer with a TDNN-F layer, which has 1536 nodes, 256 bottleneck nodes, and no time stride. The attention block has eight heads, the value-dim and key-dim are set to 128 and 64, respectively, the context-width is 10 with the same number of left and right inputs, and the time stride is 3.
- **Multistream CNN:** We positioned a 5-layer CNN to better accommodate the top SpecAugment layer, followed by an 11-layer multistream CNN [14].

2.2. Multistream CNN Architecture

Multistream CNNs have shown their superiority in robust speech recognition since their diversity in temporal resolutions across multiple parallel streams would achieve stronger robustness.

As shown in Figure 1, the input speech frames were first processed by a few initial single stream CNN layers, which could be TDNN-F or 2D-CNN, and then entered multiple specific branches that were stacked by TDNN-F layers. To achieve the diversity of temporal resolution, every branched stream has a unique dilation rate, which corresponds to the time stride in the TDNNs. Each dilation rate was chosen from the default subsampling rate (three frames) to make the TDNN-Fs better streamlined with the training and decoding process when given input speech frames are subsampled. In our multistream CNN network, the log-mel spectrogram was first randomly masked in both frequency and time by a SpecAugment layer, then five layers of 2D-CNN were positioned to better accommodate the features. We used 3×3 kernels for the 2D-CNN layers; the filter size of the first two layers was 64, the third and the fourth were 128, and the last was 256. For every other 2D-CNN layer, we applied frequency band subsampling with a rate of two. In the multi-stream part, in each branch, we stacked 11 layers of TDNN-F with 512 nodes and 128 bottleneck nodes. We employed three streams with the 6-9-12 dilation rate configuration, which means the TDNN-Fs of each stream had 6, 9, and 12 time-stride, respectively. The output embeddings of multiple streams were then concatenated followed by ReLU, batch normalization, and a dropout layer.

Figure 1. Schematic diagram of the multistream CNN acoustic model architecture.

2.3. Accent/Speaker Embeddings

I-vector is a popular technology in the field of speaker recognition. It was motivated by the success of the Joint Factor Analysis (JFA) [23]. JFA is used to construct the subspace of speaker and channel separately, proposing powerful tools to model the inter-speaker variability and to compensate for channel/session variability in the context of GMM. However, Dehak et al. [24] proved that channel factors estimated using JFA, which are supposed to model only channel effects, also contain information about speakers. Thus, i-vector methods construct a low-dimensional subspace, termed the total variability space. This space contains factors of both speaker and channel variability. In this way, the i-vector models both speaker and channel information and characterizes most of the useful speaker-specific information in a fixed and low dimensional feature.

Due to the powerful representation capabilities of DNNs, x-vector is now a powerful representation for speaker recognition. On one hand, the TDNN-based architecture models the short-term context; on the other hand, the statistics pooling layers process all the information across the time dimension, so that subsequent layers operate on the entire segment. The DNN architecture used to extract x-vectors is outlined in Table 1. The splicing parameters for the five TDNN layers were: $\{t - 4, t - 3, t - 2, t - 1, t, t + 1, t + 2, t + 3, t + 4\}$, $\{t - 2, t, t + 2\}$, $\{t - 3, t, t + 3\}$, $\{t\}$, and $\{t\}$. The statistics pooling layer was used to calculate the mean and standard deviation on all frames of the input segment, followed by upper layers at the segment level with a softmax output layer. Segment-level embeddings

were extracted from the 512-dimensional affine component of the first fully connected layer. Speaker embeddings, which capture both speaker and environment specific information, have been shown to be useful for the ASR task. For accented speech, the information of tone and speaking habit of each accent is especially important. We have explored four types of representations as auxiliary inputs in a neural network to further improve the accuracy of the accented ASR system; the procedure is shown in Figure 2. In the procedure, we first extracted embeddings of each utterance and then concatenated them to each frame feature of the utterance as a compensatory input for the neural network.

Figure 2. The overall framework of the system.

Table 1. Accent/spk-xvector architecture.

Layer	Layer Type	Context	Size
Frame1	TDNN	$\{t - 4: t + 4\}$	512
Frame2	TDNN	$\{t - 2, t, t + 2\}$	512
Frame3	TDNN	$\{t - 3, t, t + 3\}$	512
Frame4	TDNN	$\{t\}$	512
Frame5	TDNN	$\{t\}$	1500
Stat pool		$\{0, T\}$	2×1500
Segment6	Affine	$\{0\}$	512
Segment7	Affine	$\{0\}$	512
Softmax		$\{0\}$	Num.accent/spk

2.4. Neural-Network Alignment

The phonetic alignment generated by GMM could be inaccurate, so we trained a DNN model using the frame-level criteria to obtain a better alignment, which substituted the GMM alignment to train acoustic models with the LF-MMI training criterion.

2.5. Language Model Rescoring

During decoding, a 4-gram language model (LM) was used to generate the lattice and score. This model had the problem of data sparsity because we only used the transcription of the training data to train it. To obtain better results, we trained a 4-layer TDNN-LSTM LM for lattice rescoring. It is worth mentioning that n was set to 20 for n -best rescoring. Furthermore, when training TDNN-LSTM LM, we also used Librispeech [25] text in addition to the transcription of the training set.

3. Experimental Results

3.1. Data Sets and Augmentation

Our experiments were conducted on the accented English data sets (16 kHz) of eight countries provided by Datatang (<https://www.datatang.com> (accessed on 24 August 2021)), including American (US), British (UK), Chinese (CHN), Korean (KR), Japanese

(JPN), Russian (RU), Portuguese (PT), Indian (IND), Spanish (SPA), and Canadian (CA), where the official training set does not include the Spanish and Canadian accents to evaluate the generalization of the ASR system. Each accented speech data were collected from 40 to 110 speakers and recorded by Android devices or iPhones in a quiet house acoustic environment. The speakers were gender-balanced, age 20 to 60. Each country's accented data consisted of about 20 h. The speech content consisted of daily communication and interaction with smart devices. The training set, development set, and evaluation set were about 148 h, 14 h, and 21 h, respectively.

We augmented the training data by changing the speed of the audio signal, producing three versions of the original signal with speed factors of 0.9, 1.0, and 1.1 [26] and then applied volume perturbation. All the systems share the same type of data augmentation techniques. In addition, SpecAugment, which provides masks on the spectrogram of input utterances, was performed during training.

3.2. Effect of Acoustic Model

Firstly, we chose the pure 11-layer TDNN-F to implement a baseline system. The features were 40-dimensional high resolution MFCC computed with a 25ms window and shifted every 10 ms. Table 2 shows that Kaldi pitch features can improve tonal languages for ASR systems. So, we chose the 43-dimensional MFCC with pitch as acoustic features for our systems.

Table 2. Effect of different acoustic models. (The best result are highlighted in bold.)

System	Features	WERs (%) on Dev	WERs (%) on Eval
TDNN-F (baseline)	MFCC	9.12	9.31
TDNN-F	MFCC + Pitch	8.97	9.18
CNN-TDNNF-Attention	MFCC + Pitch	8.92	9.12
Multistream CNN	MFCC + Pitch	8.86	9.08

Secondly, we replaced the network with several models as illustrated in Section 2.1. The results of different network architectures are shown in Table 2. Finally, the multistream CNN was trained and obtained the best performance. As a result, we yielded relative WER improvements of 2.9% and 2.5% on the development set and evaluation set, respectively, compared with the baseline system.

3.3. Effect of Accent/speaker Embeddings

To further improve the performance of the ASR system, we explored various embeddings as auxiliary features while sharing the same acoustic model and training strategy. The WERs achieved by the seven systems are reported in Table 3.

Model M1 is a model trained without any auxiliary embeddings, and its WER was highest in Table 3, so we can conclude that using embeddings as complementary features can significantly improve the performance of an accented English speech recognition system.

We observe that both spk-ivectors model (M2) and accent-ivectors model (M3) can achieve similar WER reduction, and the spk-ivectors + accent-ivectors model (M4) obtained a further improvement on the development set, which means that both speaker-relevant information and accent-relevant information are helpful for accent adaption in accented ASR. These results can also be observed from M5 to M7, which were the models that applied the x-vector embeddings as auxiliary features.

In Table 3, we observe that the x-vector embeddings (M5, M6, and M7) outperformed i-vector embeddings (M2, M3, and M4). Compared with Model M4, which utilized the combination of spk-ivectors and accent-ivectors, Model M6, which just augmented input with accent-xvectors obtained the same 7.02% WER on the development set. This is because the x-vector has a more powerful capability to characterize accent-relevant information.

When combining the accent-xvectors and spk-xvectors in Model M7, we achieved a relative WER reduction of 21.6% on the development set, and 14.8% on the evaluation set, compared with the baseline model M1.

Table 3. WERs (%) achieved by multistream CNN with various input embeddings. (The best result are highlighted in bold.)

Embeddings	Dev	Eval
[M1] w/o embeddings	8.86	9.08
[M2] Spk-ivectors	7.18	8.01
[M3] Accent-ivectors	7.17	7.95
[M4] + spk-ivectors	7.02	8.02
[M5] Spk-xvectors	7.04	7.76
[M6] Accent-xvectors	7.02	7.89
[M7] + spk-xvectors	6.95	7.74

3.4. Effect of Language Model Rescoring

We selected the best model M7 from Table 3 and applied N-best rescoring using the TDNN-LSTM LM. As shown in Table 4, the best system was taken to be the one that performed best on the development set. The input feature was a 43-dimensional Kaldi MFCC with Pitch appended with spk-xvectors and accent-xvectors, the acoustic model was multistream CNN, and with LM rescoring we obtained 5.41% WER on the development set and 5.99% on the evaluation set. Compared with the baseline system, we achieved relative WER improvements of 40.7% and 35.7% on the development set and evaluation set, respectively.

Table 4. Effect of language model rescoring. (The best result are highlighted in bold.)

System	Features	Embeddings	WER (%) on Dev	WER (%) on Eval
Baseline (TDNN-F)	MFCC	-	9.12	9.31
Multistream CNN	MFCC + Pitch	accent-xvectors +spk-xvectors	6.95	7.74
Multistream CNN +LM rescoring	MFCC + Pitch	accent-xvectors +spk-xvectors	5.41	5.99

3.5. Results of Different Countries' Accents

In addition, we further tested the performance of our system on accents of different countries. As shown in Table 5, our system had the lowest WER on the British accent, but it also had a relatively low WER on Japanese and Korean accents.

Table 5. Results of different countries' accents.

Country	WER (%) on Dev	WER (%) on Eval
China	8.10	9.29
Japan	4.08	4.01
India	6.79	7.24
USA	5.82	4.29
Britain	2.90	3.87
Portugal	4.42	4.57
Russia	6.23	6.83
Korea	4.86	3.99

4. Conclusions

In this paper, we explored various approaches to improve the accuracy of the accented ASR system. For the acoustic model, we tried TDNN-F, CNN-TDNNF-Attention, and

multistream CNN; we found that multistream CNN had the best performance among them. Then we explored various speaker/accent embeddings to further improve the accuracy of accented ASR systems. Experiments showed that using embeddings, which capture the accent/speaker-relevant information as auxiliary inputs can significantly improve the accuracy of accented ASR system. Finally, a language model (LM) was trained for lattice rescoring. We chose the best model from previous attempts to perform LM rescoring and achieved a great improvement in comparison with our baseline.

We will continue to improve the robustness of our acoustic model, including adopting layers with deeper levels and other CNN architectures; then we plan to experiment with End-to-End models as well. Furthermore, Chinese accented ASR is also a great challenge for the current ASR system; we will conduct experiments on Chinese and English accented speech recognition at the same time.

Author Contributions: Conceptualization, F.T., D.L., S.L., Q.H. and L.L.; methodology, F.T., D.L., S.X. and S.L.; software, D.L. and S.X.; validation, S.X. and T.L.; formal analysis, S.L.; investigation, S.X.; resources, Q.H. and L.L.; writing—original draft preparation, D.L., T.L. and F.T.; writing—review and editing, Q.H. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 61876160 and No. 62001405) and in part by the Science and Technology Key Project of Fujian Province, China (Grant No. 2020HZ020005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Feng, S.; Kudina, O.; Halpern, B.M.; Scharenborg, O. Quantifying bias in automatic speech recognition. *arXiv* **2021**, arXiv:2103.15122.
2. Vergyri, D.; Lamel, L.; Gauvain, J.L. Automatic speech recognition of multiple accented English data. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010.
3. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv* **2019**, arXiv:1901.02860.
4. Gao, Q.; Wu, H.; Sun, Y.; Duan, Y. An End-to-End Speech Accent Recognition Method Based on Hybrid CTC/Attention Transformer ASR. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7253–7257. [[CrossRef](#)]
5. Li, S.; Ouyang, B.; Liao, D.; Xia, S.; Li, L.; Hong, Q. End-To-End Multi-Accent Speech Recognition with Unsupervised Accent Modelling. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6418–6422. [[CrossRef](#)]
6. Chen, Y.C.; Yang, Z.; Yeh, C.F.; Jain, M.; Seltzer, M.L. Aipnet: Generative adversarial pre-training of accent-invariant networks for end-to-end speech recognition. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, Barcelona, Spain, 4–8 May 2020; pp. 6979–6983.
7. Na, H.J.; Park, J.S. Accented Speech Recognition Based on End-to-End Domain Adversarial Training of Neural Networks. *Appl. Sci.* **2021**, *11*, 8412. [[CrossRef](#)]
8. Tan, T.; Lu, Y.; Ma, R.; Zhu, S.; Guo, J.; Qian, Y. AISpeech-SJTU ASR System for the Accented English Speech Recognition Challenge. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6413–6417. [[CrossRef](#)]
9. Shi, X.; Yu, F.; Lu, Y.; Liang, Y.; Feng, Q.; Wang, D.; Qian, Y.; Xie, L. The Accented English Speech Recognition Challenge 2020: Open Datasets, Tracks, Baselines, Results and Methods. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6918–6922. [[CrossRef](#)]
10. Najafian, M.; Russell, M. Automatic accent identification as an analytical tool for accent robust automatic speech recognition. *Speech Commun.* **2020**, *122*, 44–55. [[CrossRef](#)]
11. Ahmed, A.; Tangri, P.; Panda, A.; Ramani, D.; Karmakar, S. VFNet: A Convolutional Architecture for Accent Classification. In Proceedings of the 2019 IEEE 16th India Council International Conference (INDICON), Rajkot, India, 13–15 December 2019; pp. 1–4. [[CrossRef](#)]
12. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K.J. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech, Signal Process.* **1989**, *37*, 328–339. [[CrossRef](#)]

13. Povey, D.; Cheng, G.; Wang, Y.; Li, K.; Xu, H.; Yarmohammadi, M.; Khudanpur, S. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In Proceedings of the Interspeech 2018, Hyderabad, India, 2–6 September 2018; pp. 3743–3747.
14. Han, K.J.; Pan, J.; Tadala, V.K.N.; Ma, T.; Povey, D. Multistream CNN for robust acoustic modeling. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 6873–6877.
15. Han, K.J.; Prieto, R.; Wu, K.; Ma, T. State-of-the-Art Speech Recognition Using Multi-Stream Self-Attention With Dilated 1D Convolutions. *arXiv* **2019**, arXiv:1910.00716.
16. Chen, M.; Yang, Z.; Liang, J.; Li, Y.; Liu, W. Improving deep neural networks based multi-accent Mandarin speech recognition using i-vectors and accent-specific top layer. In Proceedings of the INTERSPEECH 2015, Dresden, Germany, 6–10 September 2015; pp. 3620–3624.
17. Turan, M.A.T.; Vincent, E.; Juvet, D. Achieving multi-accent ASR via unsupervised acoustic model adaptation. In Proceedings of the INTERSPEECH 2020, Shanghai, China, 25–29 October 2020.
18. Karafiát, M.; Veselý, K.; Černocký, J.H.; Profant, J.; Nytra, J.; Hlaváček, M.; Pavlíček, T. Analysis of X-Vectors for Low-Resource Speech Recognition. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6998–7002. [[CrossRef](#)]
19. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi Speech Recognition Toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Waikoloa, HI, USA, 11–15 December 2011.
20. Povey, D.; Peddinti, V.; Galvez, D.; Ghahremani, P.; Manohar, V.; Na, X.; Wang, Y.; Khudanpur, S. Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In Proceedings of the Interspeech 2016, San Francisco, CA, USA, 8–12 September 2016; pp. 2751–2755.
21. Povey, D.; Hadian, H.; Ghahremani, P.; Li, K.; Khudanpur, S. A Time-Restricted Self-Attention Layer for ASR. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5874–5878.
22. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the Interspeech 2019, Graz, Austria, 15–19 September 2019; pp. 2613–2617.
23. Kenny, P.; Boulianne, G.; Ouellet, P.; Dumouchel, P. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1435–1447. [[CrossRef](#)]
24. Dehak, N. Discriminative and Generative Approaches for Long-and Short-Term Speaker Characteristics Modeling: Application to Speaker Verification. Ph.D. Thesis, École de Technologie Supérieure, Montreal, QC, Canada, 2009.
25. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015.
26. Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M.L.; Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5220–5224.