BCN2BRNO: ASR System Fusion for Albayzin 2022 Speech to Text Challenge

Jahnavi Umesh[1,*], *Martin Kocour*[1,*], *Martin Karafiát*[1,*], *Ján Švec*[1,*], *Fernando López*[2,*], *Karel Beneš*[1,*], *Mireia Diez*[1], *Igor Szöke*[1], *Jordi Luque*[2], *Karel Veselý*[1], *Lukáš Burget*[1], *Jan Černocký*[1]

[1]Brno University of Technology, BUT Speech@FIT and IT4I Centre of Excellence, Czechia
[2]Telefónica I+D, Research

ikocour@fit.vutbr.cz

Abstract

This paper describes the joint effort of BUT and Telefónica Research on the development of Automatic Speech Recognition systems for the Albayzin 2022 Challenge. We train and evaluate both hybrid systems and those based on end-to-end models. We also investigate the use of self-supervised learning speech representations from pre-trained models and their impact on ASR performance (as opposed to training models directly from scratch). Additionally, we also apply the Whisper model in a zero-shot fashion, postprocessing its output to fit the required transcription format. On top of tuning the model architectures and overall training schemes, we improve the robustness of our models by augmenting the training data with noises extracted from the target domain. Moreover, we apply rescoring with an external LM on top of N-best hypotheses to adjust each sentence score and pick the single best hypothesis. All these efforts lead to a significant WER reduction. Our single best system and the fusion of selected systems achieved 16.3 % and 13.7 % WER respectively on RTVE2020 test partition, i.e. the official evaluation partition from the previous Albayzin challenge.

Index Terms: ASR fusion, end-to-end model, self-supervised learning, automatic speech recognition.

1. Introduction

This paper describes the BCN2BRNO team's Automatic Speech Recognition (ASR) system for the Albayzin 2022 Speech to Text Transcription (STT) Challenge. We present the detailed description of the datasets, as well as technical details for the development of subsystems and the fusion. This effort is a collaboration between BUT Speech@FIT research group and Telefónica Research (TID). Our first goal in the challenge was to compare recent deep learning architectures to classical hybrid ASR systems, e.g. by integrating novel research developments into our conventional hybrid Spanish speech recognition system. Our second goal was, to significantly improve the baseline performance of our previous Albayzin systems [1].

Our primary system is a word-level ROVER fusion of five individual models. It achieves 13.7 % WER on our development set, that is, the official evaluation dataset in the previous Albayzin STT 2020 Challenge. The LM rescoring is carried out separately for some of the individual ASR subsystems. The rescored N-best lists from the subsystems are then aligned into a single confusion network using the SRILM N-best Rover tool [2]. The three additionally submitted contrastive systems are: (1) a fusion of fewer individual ASR systems, (2) the best individual system, and (3) the recent Whisper system, vanilla except for our custom output normalization.

*Equal contribution.

2. Data

In Albayzin 2022 Speech to Text Challenge, we are provided with the databases from the previous evaluations: RTVE2018 and RTVE2020, together with RTVE2022 [3] created freshly for this year's challenge. RTVE2018 is a collection of shows from public Spanish Television (RTVE) during the years 2015 to 2018. It contains 569 hours of audio, from which 468 hours are provided with subtitles and the rest contains human-revised transcripts. RTVE2020 consists of TV shows of different genres, broadcast by the RTVE from 2018 to 2019. RTVE2022 is a collection of diverse audio materials from the 1960's to the present. All RTVE databases together contain around 768 hours of audio content.

For this year's challenge, we split these databases into training, development, and validation sets. We followed the original data splits and use the whole RTVE2018 database (train, dev1, dev2, test partition) together with the RTVE2022's train partition for training, which resulted in 738 hours of training data. We used the RTVE2022's dev partition with 2.5 hours of audio for development. This dev partition was originally designed for Albayzin 2020 Text and Speech Alignment Evaluation (TaSA), where the transcripts were generated from respoken recordings [4]. The RTVE2020's test partition with 39 hours of recordings was used for cross-validation. This partitioning allows us to compare the performance of our models with the performance of systems provided by participants in the previous challenge.

Since our training and development data contains inaccurate transcripts, we needed to filter them out. We followed the same transcript retrieval process as in [1, Section 2.1]. This resulted in 512 hours of clean training data (out of initial 738 hours) and 41 minutes of clean development data (out of initial 2.5 hours).

2.1. Noise data augmentation

In addition to the RTVE data, we also use Spanish CommonVoice dataset, which comprising around 400 hours of read speech validated by volunteers [5]. Instead of using this dataset directly, we corrupt it with noise to better match the target domain, i.e. RTVE2022 dev and RTVE2020 test sets.

The noise data augmentation is a 3-step process. First, we extract non-speech segments longer than 2 seconds from the RTVE dataset. Then, these noises (with the addition of *restaurant*, *street*, *home*, *workshop*, and *fan* noises [1]) are used to degrade the CommonVoice data. Finally, we reverberated [6] 50 % of the data and transcoded one fifth of the data by various codecs (i.e. *AMR-xx*, *G.7xx* and *GSM-xx*). Target SNR was randomly chosen in the range of 6 dB to 20 dB. The resulting augmented data partition is referred to as "CV_aug" in the rest of the text.

[1]Mentioned noises were downloaded from `freesound.org`.

3. Automatic Speech Recognition

Our automatic speech recognition pipeline consists of 4 conceptual blocks: a) voice activity detection, b) end-to-end and hybrid ASR models, c) RNN-LM rescoring, and d) shallow fusion. First, we split each audio recording into smaller segments with our custom voice activity detection. The segmented speech is then processed by the ASR models, where each model produces a N-best list of hypotheses and scores. The scores are later re-calibrated using RNN-based language model, and the 1-best hypothesis is generated for each ASR system based on the adjusted score. Finally, the 1-best transcripts from all ASR models are united into a single output using shallow fusion. In the following sections, we describe each block in more detail.

3.1. Voice activity detection

Voice activity detection (VAD) is applied to both development and evaluation data in order to segment long audio recordings into smaller chunks containing speech. Our VAD is based on a simple feed-forward neural network with two outputs (i.e., speech and non-speech) and four layers with 400 neurons each. For the input, 15-dimensional standard Mel-filter bank features are combined with 3 additional Kaldi pitch coefficients [7], and cepstral mean normalization is applied. The VAD is trained on RTVE2018 dev1, dev2, and test data (101 hours). Pre-softmax outputs are converted to logit posteriors and smoothed by averaging over 31 consecutive frames. Speech segments are extracted by thresholding the smoothed logit posteriors at the value of -0.5.

3.2. ASR models

We experimented with 5 different ASR models. Three models are using Encoder-Decoder Transformer architecture [8] (XLS-R Conformer, XLSR-53-CTC and Whisper model [9]). The fourth is RNN Transducer architecture [10], and the fifth is the hybrid DNN-HMM model [11].

XLS-R Conformer Inspired by the recent success of pre-trained models based on wav2vec2.0 [12], we decided to examine these models more closely in this challenge. We first built a conformer model from scratch using Mel-filter bank features and then we investigated the use of XLS-R (wav2vec2.0) embeddings as features.

The Conformer model was trained from scratch in ESPnet2 framework [13]. The input features are 80-dim Mel-filter bank outputs. We use SpecAugment data augmentation [14] with time-warp of 5 windows, with the frequency mask applied twice in range of 0 to 30 Mel-filter channels. The time mask is also applied twice in range of 0 to 40 frames. This system is based on the Conformer architecture [15] and is composed of 12 encoder layers and 6 decoder layers. The conformer encoder layer incorporates, in addition to a self-attention module, a convolutional layer in between two feed-forward modules. The decoder was built using masked self-attention as well as cross-attention between the encoder embeddings and the decoder. Each encoder and decoder layer outputs 512 dimensional embeddings; attention is done with 8 parallel heads and the feed-forward module expands the data into 2048 dimensions. We use the standard ESPNnet2 training recipe with 25k warm-up steps and the learning rate $8 \cdot 10^{-4}$. We use byte-pair encodings (BPE) [16] as target output units and empirically found 1500 BPEs gave the best performance on the dev set. The models are trained with the joint CTC/Attention loss with the CTC weight of 0.3. This model gave a performance of 18.9 % WER without the use of any external language model.

Next, we explore the use of XLS-R wav2vec2.0 embeddings as features instead of Mel-filter bank features. The large XLS-R-128 model was trained on 436,000 hours of unlabelled speech data from 128 languages [17]. We use the 0.3 billion parameter model of XLS-R. This model was imported into ESPnet-2 using the S3PRL framework and is used only as the feature extractor with multi-layer feature aggregation. The 1024-dimensional embeddings of the XLS-R are subsequently used as the input features to the conformer model. This setup is exactly identical to the previous one, except the mel-filter-bank features replaced by the XLS-R embeddings. During the training of this model, the XLS-R parameters are not updated. For training, we have increased the learning rate to $2.5 \cdot 10^{-3}$ and the number of warm-up steps to 40k. We found the use of the XLS-R embeddings as features provided 1.8 % abs. improvement of WER over the conventional end-to-end conformer.

XLSR CTC We also investigated the use of the smaller original XLSR-53 model, pre-trained with 56k hours of audio in 53 languages [18], and then fine-tuned to the Spanish Common-Voice. We added two linear layers randomly initialized on top of the Wav2Vec2.0 architecture. The resulting model has more than 300M trainable parameters and outputs 38 distinct characters: unaccented letters a – z, accented vowels á, é, í, ó, ú, and the diaeresis on the vowel u(ü). The transcription is obtained using simple greedy decoding from the frame-wise character posteriors that the model produces.

The model training has been based on the CTC recipe from the SpeechBrain toolkit [19]. We only use SpecAugment [14] as an augmentation method. The model was trained using the CTC loss only and the learning rates are updated using the NewBob scheduler [20]. Additionally, we manually restarted the LR at some points in training. The ASR was trained using a batch size of 3, setting the starting LRs for the linear layers and Wav2Vec2.0 to 1.0 and 10^{-5}, respectively. Finally, the best checkpoint in terms of WER is stored; this purely acoustic end-to-end ASR model achieved 24.6 % WER on RTVE2020.

Whisper We ran the vanilla Whisper [9] model ("large", 1550M parameters) on our VAD segmentation. The raw transcripts contained several mistakes, so we further applied three-step transcription filtering. First, we omitted all the punctuation and converted numbers to textual form. This gave us 1 % abs. reduction of WER. Then, we removed 1397 segments with transcript "i" which indicates non-sense transcription (caused by acoustic mismatch, music, etc.), yielding 0.1 % abs. Finally, we observed that in difficult acoustic conditions such as music, the Whisper decoder tends to get stuck and produce long strings of repetitive symbols, e.g. "lalalalala...", "tadadadada...". We used `zlib` to compress the segment transcripts and filtered out segments with compression factor higher than 2, resulting in deletion of 566 very well-compressed segments and an absolute improvement of 0.9 % WER.

RNN-T We also investigated the performance of RNN-T model. We chose the SpeechBrain implementation. This recipe uses CRDNN architecture [21] for the RNN-T's transcription network and an RNN with gated recurrent units (GRU) [22] for the prediction network. The joiner simply sums the transcription network output and the prediction network output and applies LeakyReLU non-linearity on top. The entire RNN-T model has more than 135M trainable parameters. The input consists of 80 filter bank features and the model predicts the posterior probabilities over a vocabulary of 1000 tokens. These tokens are a mixture of words, BPEs, and characters. We used the "Unigram" algorithm from SentencePiece [16] to generate the token vocabulary. The tokenizer was trained on cleaned

RTVE training transcripts. The model was trained from scratch on cleaned RTVE training data using the RNN-T objective function. We applied SpecAugment technique on features only to mask frequency bins.

Hybrid CNN-TDNN-HMM Hybrid DNN-HMM ASR was trained with the Kaldi toolkit. Factorized Time Delay NN (TDNNf) architecture [11] with convolutional layers (CNN) was selected as it was showing similar performance to more complicated recurrent NN types including those based on Long-Short-Term Memory (LSTM) cells. Our CNN-TDNNf architecture contains 6 CNN layers (with 64, 64, 128, 128, 256, 256 filters) followed by 19 TDNNf layers each with 1536 neurons, and bottle-neck factorization to 160 dimensions with stride 3.

Feature augmentation adaptation method was used in our system to deal with various acoustic condition in target data. It incorporates a compact representation of speaker or noise information into a fixed-dimensional vector appended to the input features. Two approaches were used in this work:

- **i-vectors** [23] – nowadays commonly used adaptation technique. Online 100 dimensional i-vectors [24] were estimated on same features as for acoustic models (40-dim MFCCs).
- **x-vectors** [25] – popular in speaker recognition field and showing also slight gain in ASR [26, 27]. Further analysis [28], used in this work, showed significant gain over i-vectors for x-vector extractor trained on a significant amount of speaker identification data.

The ASR feature extraction is based on 40-dim MFCCs, where inverse cosine transform is applied preceding the input of the NN. It re-creates the Mel-filter bank outputs more suitable for further CNN processing. The adaptation vectors are transformed by affine transform to 200 dimensions. Both feature streams are concatenated and serve as CNN input. NNs are trained with the Lattice Free Maximum Mutual Information (LF-MMI) objective and bi-phone targets as suggested in [29].

3.3. RNN-LM rescoring

Our ASR systems produce N-best lists[2], that are rescored by an external language model. The LM is a custom LSTM trained using BrnoLM[3]. The LM consists of two layers of LSTM with 1500 units each and operates on an independent vocabulary of 20k BPE units. We have pretrained the language model on a collection of Spanish newspaper texts (around 440M tokens) and then fine-tuned it to the transcripts of the training data.

To obtain actual improvements from the LM rescoring, we tuned the weight of the LM score (with optimal values in 0.25 – 0.3, compared to 1 as the weight of the original ASR score) and the word insertion bonus (with optimal values in 5.5 – 6.5) on the development data. We did not attempt to subtract the internal LM of the decoder.

3.4. System fusion

To facilitate effective fusion of outputs of the different systems, we first compact each resulting N-best list into a confusion network. This is done by iteratively aligning the individual hypotheses to the currently best scored path in the confusion network. Then, the best path through the confusion network is taken and the respective bin-posterior probability is assigned as confidences to each word.

To reconcile the differences in ASR system outputs, these best paths are taken from each ASR system and united into a single transcript by system fusion based on voting process. We used NIST ROVER [2], where the voting is done according to the word frequency and maximum confidence. We tuned the α parameter, which is a trade-off between frequency of word occurrence and maximum word confidence. We also tuned the null word confidence (also known as blank symbol confidence). We set α to 0.7 and null word confidence to 0.9. We did not use the time information during fusion.

4. Results

The overall results are presented in Table 1. The initial part of the table shows performance of individual ASR systems, while the latter part contains results obtained from fusion of ASR systems. The best single system is *XLSR-128-Conformer (c2)* with 17.1 % WER, where we used XLS-R-128 model to produce feature embeddings and trained a Conformer model on top of them. Without the pre-trained model, the Conformer WER is 18.9 %.

Next, the results show that rescoring of N-best lists by external language model provides significant benefits. It improves WER of each system by 0.7 % absolute on average. Our best single-model ASR system achieved 16.3 % WER with RNN-LM rescoring and 17.1 % WER without rescoring.

Table 1: *Word error rates of our models on RTVE2020 test set with and without external RNN-LM rescoring.* CV_aug *indicates the model was trained both on RTVE and augmented CommonVoice datasets. Note that the hybrid ASR model contains n-gram LM by design, marked with* †.

	Model	Test [% WER] w/o LM	with LM
1	XLSR-128-Conformer (c2)	**17.1**	**16.3**
1*	Conformer (no pre-training)*	18.9	–
2*	XLSR-128-Conformer*	17.5	17.0
2	+ *CV_aug*	17.7	16.9
3	Whisper (c3)	19.1	–
4	CNN-TDNN-HMM + i-vectors	22.2†	–
5	CNN-TDNN-HMM + x-vectors	21.6†	–
6	XLSR-53-CTC	24.6	–
7	RNN-T	24.7	–
8	+ *CV_aug*	21.5	20.9
9	Fusion 1 + 2	17.1	16.0
10	Fusion 1 + 2 + 8	16.4	15.7
11	Fusion 1 + 2 + 5 (c1)	16.0	–
12	Fusion 1 + 2 + 5 + 8	15.7	–
13	Fusion 1 + 2 + 5 + 6 + 8	15.2	14.8
14	+ Whisper w/o LM (p)	**14.8**	**13.7**

We mixed the CommonVoice recordings with noises extracted from the Albayzin training data this time, resulting in the *CV_aug* set. Appending this set into the training (System 2 in Table1) shows slight 0.2 % degradation when no RNN-LM was used, but 0.1% improvement when RNN-LM rescoring is used. We assume that *CV_aug* data badly influenced the model decoder, which was later corrected by using RNN-LM. Unfortunately, the baseline system, 2*, has a small bug in the setup leading to worse performance than the corrected system 1.

[2]With the notable exception of Kaldi, whose lattices can be easily reduced to N-best lists.

[3]https://github.com/BUTSpeechFIT/BrnoLM

The performance of Whisper, 19.1 % WER, is not bad at all, given that it is a generic multilingual ASR system that was not adapted to Albayzin data. The Kaldi systems 4 and 5 are about 4.5 % absolute behind the best single system and the XLSR-53-CTC and RNN-T systems are about 7 % WER abs. behind the best single system. Interestingly, adding the augmented CommonVoice data *CV_aug* helped the RNN-T system dramatically, reducing the WER by 3.2 % abs.

Another large WER reduction comes from the fusion of ASR systems. Thanks to a diverse mix of ASR systems (wav2vec2.0 based Conformer models, hybrid model, RNN-T model and Whisper Transformer model), we were able to further decrease WER by 2.3 % absolute. The systems have various architectures, as well as objective functions and pre-training. Note that the fusion of XLSR Conformers with hybrid ASR (System 11) achieved a solid 1.1 % absolute WER reduction w.r.t. the fusion of just XLSR Conformer models (System 9). Finally, our overall best result 13.7% WER was achieved by fusing 6 ASR systems in combination with RNN-LM rescoring (System 14).

Note also that in all our fusions, the wav2vec2.0 model occurred twice. This is to give higher emphasis on the words from the two best systems in the majority voting done by ROVER. We use this simple trick, since our word-level confidences were computed just from aligned N-best lists from each single system, e.g. if the same incorrect word occurred in many single model hypotheses, it obtained high confidence score. Calibrating the confidences on a held-out set should solve this issue.

4.1. Submitted systems

The primary system (marked '(p)' in Table 1) is a fusion of rescored hypotheses from XLSR-128-Conformer models (Systems 1 and 2), rescored hypotheses from RNN-T model (System 8) and, without rescoring, the transcripts from Whisper, hybrid CNN-TDNN-HMM and XLSR-53-CTC models (Systems 3, 5, and 6). We submitted also 3 contrastive systems ('(c1)', '(c2)' and '(c3)'): '(c1)' is a fusion of rescored hypotheses from the two XLSR-128-Conformer models (Systems 1 and 2) and non-rescored hypotheses from our hybrid model (System 5); '(c2)' is our best single-model ASR (System 1) with rescored hypotheses and '(c3)' are Whisper transcripts without LM rescoring (System 3).

Table 2 shows that performance differences between RTVE2020 and RTVE2022 test sets are quite small, which is a sign of good generalization. Also, the ranking of the systems is the same on both test sets.

In Table 3, we see that inside the RTVE2022 test set, there are huge WER differences across the TV shows spanning from 5.89 % to 49.85 %. The worst performance was achieved on APB TV show. We find, during the post-eval analysis, that this particular TV show is just partially transcribed and sometimes the "reference" transcripts contain sentences, which speaker did not say verbatim. We also notice that some of our substitutions were actually correct words.

Table 2: *Word error rate comparison of submitted systems on RTVE2020 test and RTVE2022 test data partitions.*

	System	RTVE2020	RTVE2022
(p)	2x Conformer, TDNN, RNN-T, CTC, Whisper	**13.7**	**14.4**
(c1)	2x Conformer, TDNN	16.0	15.2
(c2)	Conformer	16.3	17.2
(c3)	Whisper	19.1	18.7

Table 3: *Word error rate decomposition on individual TV shows from RTVE2022 test.*

		System			
TV Show	#words	p	c1	c2	c3
ED	40317	14.38	15.28	16.96	23.22
AG	39853	5.89	6.28	7.27	7.16
CA	36186	17.71	19.10	22.05	19.28
SYG	34356	10.05	10.67	11.76	19.94
EC	30174	13.61	14.68	16.68	16.33
CO	29110	8.55	9.25	10.83	15.59
AT	28817	11.16	12.04	13.92	18.63
3x4	23548	13.11	14.01	14.97	14.23
EE	23398	23.55	24.61	26.83	26.85
NN	19027	9.33	10.10	11.55	10.63
CCA	18026	10.31	11.00	12.13	13.18
ERA	15369	22.08	22.75	24.62	25.40
ATE	12149	9.47	10.41	12.55	10.05
JYS	9464	11.79	11.95	14.64	18.43
IU	8947	20.36	21.22	24.65	24.81
GR	8694	26.63	29.00	32.34	27.33
APB	7439	49.85	51.27	61.12	59.66
RD	6943	20.60	22.41	25.38	20.50
CPE	6871	13.87	14.86	17.25	15.01
TO	6380	23.98	24.73	25.82	26.69
YR	4213	29.74	30.52	35.22	32.33
Total	409 281	**14.35**	15.24	17.22	18.65

5. Conclusions

In this paper, we described our submitted systems for Albayzin 2022 Speech to Text Challenge. The processing pipeline consists of voice activity detection, speech recognition, external language model rescoring, and shallow fusion. We showed that each mentioned part contributes to the performance. Our best single-model ASR system achieved 17.2 % WER in this year's challenge, while our best fusion achieved 14.4 % WER. From these numbers we can conclude that all six systems used in the fusion are in some sense complimentary. Surprisingly, the Whisper model also performs reasonably well; it achieved just 18.7 % WER, while it was not trained on any in-domain data. This suggests that fine-tuning of the model might dramatically improve our system, and thus it is a subject of our future work.

6. Acknowledgements

We would like to thank Ekaterina Egorova for meaningful discussions and help with editing of this article. The work was partly supported by Czech National Science Foundation (GACR) project NEUREM3 No. 19-26934X, Czech Ministry of Education, Youth and Sports from project no. LTAIN19087 "Multi-linguality in speech technologies", Czech Ministry of Interior project No. VJ01010108 "ROZKAZ", and Horizon 2020 Marie Sklodowska-Curie grant ESPERANTO, No. 101007666. Computing on IT4I supercomputer was supported by the Czech Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project "e-Infrastructure CZ – LM2018140".

7. References

[1] M. Kocour, G. Cámbara, J. Luque, D. Bonet, M. Farrús, M. Karafiát, K. Veselý, and J. Černocký, "BCN2BRNO: ASR System Fusion for Albayzin 2020 Speech to Text Challenge," in *Proc. IberSPEECH 2021*, 2021, pp. 113–117.

[2] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, 1997, pp. 347–354.

[3] E. Lleida, A. Ortega, A. Miguel, V. Bazán, C. Pérez, M. Gómez, and A. de Prada, "Rtve 2018, 2020 and 2022 database description." [Online]. Available: http://catedrartve.unizar.es/reto2022/RTVE2022DB.pdf

[4] E. Lleida, A. Ortega, A. Miguel, V. Bazán, C. Pérez, , and A. de Prada, "Albayzin evaluation: Iberspeech-rtve 2022 text and speech alignment challenge: Alignment of respoken subtitles." [Online]. Available: http://catedrartve.unizar.es/reto2022/TaSAC-ST2022_Evalplan.pdf

[5] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[6] I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, 2019.

[7] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE, May 2014.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[9] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022. [Online]. Available: https://cdn.openai.com/papers/whisper.pdf

[10] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012. [Online]. Available: http://arxiv.org/abs/1211.3711

[11] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proceedings of Interspeech*, 09 2018, pp. 3743–3747.

[12] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html

[13] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[14] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[15] A. Gulati, J. Qin, C. C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[16] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018. [Online]. Available: https://arxiv.org/abs/1808.06226

[17] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," *arXiv preprint arXiv:2111.09296*, 2021.

[18] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[19] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[20] S. Renals, N. Morgan, M. Cohen, and H. Franco, "Connectionist probability estimation in the decipher speech recognition system," in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1992, pp. 601–604 vol.1.

[21] T. Parcollet and M. Ravanelli, "The Energy and Carbon Footprint of Training End-to-End Speech Recognizers," Apr. 2021, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03190119

[22] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: http://arxiv.org/abs/1406.1078

[23] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[24] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proceedings of Interspeech*, 2015.

[25] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*, 2018.

[26] M. A. T. Turan, E. Vincent, and D. Jouvet, "Achieving multi-accent ASR via unsupervised acoustic model adaptation," in *Interspeech 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 1286–1290.

[27] J. Rownicka, P. Bell, and S. Renals, "Embeddings for dnn speaker adaptive training," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 479–486.

[28] M. Karafiát, K. Veselý, J. Černocký, J. Profant, J. Nytra, M. Hlaváček, and T. Pavlíček, "Analysis of x-vectors for low-resource speech recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Signal Processing Society, 2021, pp. 6998–7002.

[29] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proceedings of Interspeech*, 09 2016, pp. 2751–2755.