IberSPEECH 2022
14-16 November 2022, Granada, Spain

TID Spanish ASR system for the Albayzin 2022 Speech-to-Text Transcription Challenge

Fernando López[1,2], *Jordi Luque*[1]

[1]Telefónica I+D
[2]Universidad Autónoma de Madrid

wiliam.lopezgavilanez@telefonica.com, jordi.luque@telefonica.com

Abstract

This paper describes Telefónica I+D's participation in the IberSPEECH-RTVE 2022 Speech-to-Text Transcription Challenge. We built an acoustic end-to-end Automatic Speech Recognition (ASR) based on the large XLS-R architecture. We first trained it with already aligned data from CommonVoice. After we adapted it to the TV broadcasting domain with a self-supervised method. For that purpose, we used an iterative pseudo-forced alignment algorithm fed with frame-wise character posteriors produced by our ASR. This allowed us to recover up to 166 hours from RTVE2018 and RTVE2022 databases. We additionally explored using a transformer-based seq2seq translator system as a Language Model (LM) to correct the transcripts of the acoustic ASR. Our best system achieved 24.27% WER in the test split of RTVE2020.

Index Terms: end-to-end model, pseudo-forced alignment, domain adaptation, automatic speech recognition

1. Introduction

This paper describes Telefónica I+D's participation in the IberSPEECH-RTVE 2022 Speech-to-Text Transcription Challenge. We explain the technical details of the datasets and subsystems that have been used for our submission.

When it comes to specific domains or low-resource tasks, end-to-end systems tend to underperform conventional approaches [1]. Given that domain-specific datasets are scarce, costly, and human-time-consuming, techniques to tackle the lack of high-quality data have been explored. A common approach is to retrieve audio-to-text alignments from available audio data that have low-quality text references [2, 3, 4, 5, 6]. Some of these systems use a post-filtering process based on a confidence score to filter-out mistaken references. Moreover, there is recent work that uses massive datasets by relaxing the human-labeled requirement (weak supervision), achieving impressive zero-shot behaviors [7]. Another line of action is the use of unlabeled audio. On the one hand, with semi-supervised methods by pseudo-labeling of unlabeled audio [8]. On the other hand, unsupervised pre-training methods by learning speech representations that are subsequently used in a concrete task (supervised). Requiring less quantity of labeled data to achieve state-of-the-art results [9, 10, 11].

In the same spirit, trying to overcome the scarcity of datasets that pair audio and text, there is work using external elements that works directly at the text level. Concretely, in [12, 13] the use of machine translation models to correct ASR outputs has been proven to improve the performance in terms of Word Error Rate (WER). Similarly, distilling knowledge from the language representation model BERT, during training, also improves the ASR performance [14].

Considering these ideas, we built an acoustic end-to-end ASR model based on the XLSR-53 architecture, which was pre-trained with 56k hours of unlabeled audio [10]. We fine-tuned this model to the Spanish language with the utterance-level aligned data from CommonVoice. Then, we used the self-supervised method presented in [6] to adapt the ASR to the TV broadcast domain. It consists of using the trained end-to-end acoustic ASR to retrieve data from RTVE2018 and RTVE2022. This process is repeated several times, as more data is aligned when the model is better adapted to the target domain. Additionally, we fine-tuned a machine translation model from Catalan to Spanish to correct ASR output (LM).

Using the same ASR, the difference between our four submissions lies in the use of different audio segments and post-processing of the ASR's hypothesis with the LM. We used both, segments coming from a Voice Activity Detector (VAD) model (c1_VAD) and segmentation with sliding window without overlap (c3_W10). Moreover, we analysed the impact of the LM in both cases (c2_W10LM and p_VADLM).

2. Databases

The Albayzin 2022 Speech-to-Text Transcription Challenge provided the databases from previous evaluations: RTVE2018 [15] and RTVE2020 [16]; together with newly released data in RTVE2022 [17]. RTVE2018 is a collection of shows from public Spanish Television (RTVE) broadcasted during the years 2015 to 2018. It has 569 hours of unaligned audio, partitioned into 4 different subsets: train, dev1, dev2, and test. The train split consists of 460 hours of audio with closed captions from TV shows. The dev1, dev2, and test splits contain 57, 15, and 41 hours of human-revised transcripts, respectively. Whereas RTVE2020 is a collection of shows from RTVE during the years 2019 to 2020. It includes a test split with 55 hours of human-transcribed audio. RTVE2022 is a collection of diverse audio materials from the 60's to the present. It has 223 hours of audio split into two partitions: train and test. The train partition has 168 hours of automatically transcribed audio, and is automatically aligned. The test partition consists of 55 hours of diverse audio material.

We additionally used for this work the Spanish CommonVoice [18] database, wich comprises more than 200 hours of reading speech, which is utterance-level aligned and validated by volunteers. Table 1 presents the amount of data of the two versions used in this work.

2.1. Splits

We created three splits from the RTVE2018 database. We named them train, dev1 and dev2; and they are used to train the ASR and the LM, validate ASR training, and validate

10.21437/IberSPEECH.2022-55

Table 1: *Number of hours and sentences for the different splits of the Spanish CommonVoice datasets.*

Split	version 6.1		version 7.0	
	hours	samples	hours	samples
train	236.9	161.8k	291.2	196.0k
dev	25.2	15.1k	25.8	15.3k
test	25.9	15.1k	26.4	15.3k
total	288	192k	343.4	226.6k

and test the LM training, respectively. Table 2 presents how we constructed our splits. We used show-based criteria to avoid speaker repetition between splits. Similarly, the show type and duration were also considered. Additionally, the full RTVE2022 database has been used for training and RTVE2020 was used to compare our results with the previous evaluation.

Table 2: *Simplified information about the shows and splits from RTVE2018 database.*

Show	Type	Hours	Split
LA24H	news	16	dev1
EC	reports	13	train
LT24HTer	interview	27	train
AFI	documentary	11	dev2
LM	news	228	train
SG	contest	29	train
AV	contest	6	train
DH	contest	10	train
LT24HEnt	interviews	5	dev1
Millenium	debate	19	dev1
LN24H	interviews	33	train
20H	news	41	train
CA	news	17	train
AP	news	70	train
AG	news	38	train
LT24HEco	economy	4	dev2
LT24HTiempo	weather	2	dev1

2.2. Alignments

After exploring the provided databases, we noticed that even human-revised transcripts present differences with the spoken content. This is accentuated in the train split of RTVE2018 where the text references provided are subtitles. There are several reasons that justify this. First, the standards used in Europe follow some restrictions related, for instance, to the reading speed, time on the screen of transcriptions or the leading-in time due to the natural eye movements [19]. Second, the kind of show affects the speech and transcription mismatch, such as in sports shows, where speaking velocity may substantially differ from subtitles velocity. Finally, spontaneous speech as in interviews or discussions may present interruptions, repetitions, revisions, and/or restarts [20]. Of course, the mechanism used to generate transcriptions also affects the result.

We decided to tackle audio-to-text mismatches by using the iterative pseudo-forced alignment algorithm presented in [6], which uses the Connectionist Temporal Classification (CTC) paths produced by an end-to-end ASR to get alignments. In this method, several combinations of audio and text are aligned until finding the best possible match. It is an anchor-based approach, so only the last aligned utterance within the analysis window is taken in to account to accept/reject alignments. Thus, mismatched audio and text alignments between anchors can be accepted, not affecting the remaining file alignment. The algorithm produces a confidence score for each aligned utterance, we used it to filter out the alignments by selecting a threshold (-1.0 in the log space). Table 3 presents the amount of data recovered by this method along iterations. As the ASR is better adapted to the acoustic domain, it is capable of recovering more data domain data.

Table 3: *Data recovery, in hours, from RTVE databases.*

Iteration	RTVE2018				RTVE2022
	train	dev1	dev2	test	train
Original	460	55	15	39	168
1st-pass	-	18	5	-	-
2nd-pass	74	30	9	21	32
Recovered	16%	55%	60%	54%	19%

3. Automatic Speech Recognition

3.1. Voice Activity Detection

A VAD model is used in evaluation data to filter non-speech segments. It consists of a dense neural network of four layers with 400 neurons each and two outputs. The network is fed with 15-dimensional Mel-filter bank features augmented with 3 additional Kaldi pitch coefficients [21].

3.2. End-to-end acoustic ASR

We constructed an end-to-end acoustic model based on the large XLS-R architecture. It consists of a convolutional feature encoder, followed by a transformer with 24 blocks with an inner dimension of 4096 and a model dimension of 1024. Concretely, we used the XLSR-53 model, which was pre-trained with 56k hours of audio in 53 languages [10], and then fine-tuned in the Spanish CommonVoice dataset (version 6.1). We added two linear layers randomly initialized on top of the Wav2Vec2.0 architecture. The resultant model counts for more than 300M trainable parameters and classifies among 38 characters, including unaccented letters between a-z, the accented vowels á, é, í, ó, ú, and the diaeresis on the vowel u(ü). Finally, the transcription is obtained using simple greedy decoding from the frame-wise character posteriors that the model produces.

3.2.1. Training

Based on the SpeechBrain's CTC recipe [22], utterance audio length is limited to up to 10 seconds. Moreover, signals are ordered by length, using shorter clips in the first batches of the epoch, and longer ones at the end. Regarding data augmentation, the unique technique used is SpecAugment [23]. Furthermore, the model is optimized minimizing only a CTC loss, and the learning rates (LR) are updated using the NewBob scheduler [24]. Additionally, we manually restarted the LR at some points in training. Finally, the best checkpoint in terms of WER is stored. The ASR has been trained using a batch size of 3, setting the starting LRs for the linear layers and Wav2Vec2.0 of 1.0 and 10^{-5}, respectively.

The model was first trained during 150 epochs with CommonVoice 6.1, after, during 15 epochs with CommonVoice 7.0. Then, we started the iterative self-supervised process of aligning data of RTVE databases with the same ASR and using that data to continue training it. We repeated this process several times, as when the model is better adapted to the broadcast TV domain it is capable of aligning more data from RTVE2018 and RTVE2022.

3.3. Machine translation LM

Additionally, we explored the use of a neural machine translation system as an external LM to correct our acoustic-only ASR hypothesis. This has been previously explored in [12, 13], given that the machine translation model is a strong candidate to learn the type of errors the ASR produces and correct them. Concretely, we used a transformer-based encoder-decoder model that translates from Catalan to Spanish [1]. We did so because the model knows how to produce correct Spanish sentences at the output. It was originally trained using the MarianMT framework [25] with the Open Parallel Corpus (OPUS) [26] by the Language Technology Research Group at the University of Helsinki.

3.3.1. Data generation

In order to fine-tune the machine translation model for the downstream task (ASR correction) we need to generate a parallel corpus with source and target sentences. We explored to ways of generating that data. The first one consist on manually generating errors in the reference text. At the word-level, we modeled deletions with the probability of our ASR to delete words in the RTVE2020 test: $^{deletions}/_{words} = 0.12$. At character-level, we produced three types of errors:

1. **typos**: simple remove, insert, substitute any character. Similarly, we modelled character swapping.
2. **misspelling**: homphonic mistakes as the replacement of *"que"* by *"ke"*, *"v"* by *"b"* or *"o"* by *"ho"*.
3. **repetitions**: simple repeat a character.

Some examples of the manually generated mistakes are provided in Table 4.

Table 4: *Manual mistakes produced to fine-tune the machine translation model.*

Source	Target
*ya **justifia** el **aswnto***	*ya justifica el asunto*
[deletion] *por veinte euros no era*	*abono por veinte euros no era*
*el **scetor** se ha **marcdao piara** dos mil **veinticiqnco***	*el sector se ha marcado para dos mil veinticinco*

The second way of generating source and target sentences was the usage of checkpoints at different points during ASR training. We used four checkpoints to process all aligned utterances and get ASR hypothesis. This was used as source data and text reference was used as target data. In Table 5 we show some examples this data. Of course, the ASR also produces the correct transcription for many utterances. This was also used to teach the model not to change the text when it is correct.

Table 5: *ASR mistakes produced to fine-tune the machine translation model.*

Source	Target
*han puesto nombres y apellidos **rog***	*han puesto nombres y apellidos roig*
*que **itana** tenga que **meterla c** mano en una caja oscura*	*que aitana tenga que meter la mano en una caja oscura*

For the manual mistakes, we used all the text references available from train splits counting a total of 427k sentence pairs. In the case of ASR mistakes, we used aligned data as we have to process the audio with the ASR to get the hypothesis. From RTVE2018 and RTVE2022 train splits, we have more than 146k sentences aligned. As a result of processing the data with four checkpoints, we have a total of 587k sentences. For validation and testing, we divided dev2 into two parts.

3.3.2. Fine-tuning

The model was fine-tuned for a maximum of 20 epochs using early stopping with a patience of 5 epochs. We only store the best model in terms of bilingual evaluation understudy (BLEU) metric in validation set (sub-set from dev2). Additionally, we used a batch size of 60 sentences and a starting learning rate of 10^{-4} scheduled with a cosine annealing scheduler [27].

4. Results

4.1. Audio segments

For generating the results we explored two approaches to segmenting the audio that the ASR processes. First, we used the VAD explained in section 3.1 to generate speech segments. Additionally, we segmented with a sliding window without overlap. In this second case, we explored several window sizes. In Figure 1 we present how the window size affects deletions, insertions, and substitutions. While reducing the window size, we decrease word deletions. Nevertheless, when the window is too small mid-word cuts cause the growth of the insertions and substitutions. Best results are obtained while using a 10 seconds-length audio window.

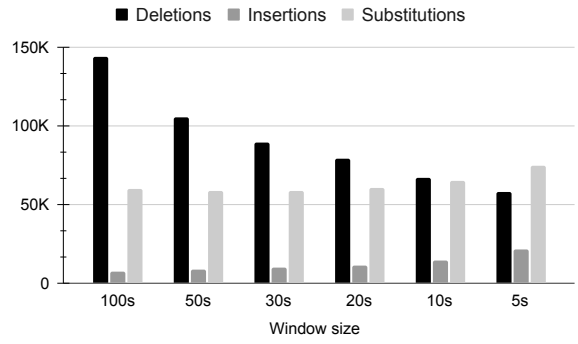

Figure 1: *Deletions, insertions and substitutions across segmentation window size used to get transcriptions.*

[1]Model downloaded from HuggingFace: https://huggingface.co/Helsinki-NLP/opus-mt-ca-es.

4.2. Submitted systems

We submitted four systems, one primary and three contrastive. The primary system (p) uses VAD segments and corrects the output with the LM. The first contrastive system (c1) just uses the VAD segments. On the other hand, the second contrastive system (c2) uses 10 seconds-length window segmentation and corrects the output with the LM. Finally, the third contrastive system (c3) generates the output using a 10 seconds-length segmentation window size.

4.3. Evaluation results

The results are presented in Table 6. As expected, filtering non-speech segments by using a VAD is better than using segmentation with a sliding window, it improves by a 1.8% the WER. Additionally, while using segments generated by a sliding window, the LM always improves the ASR results, as the ASR tends to generate non-sense outputs when processing non-speech audios such as opening/ending music. The best system in RTVE2020 test is VADLM, which uses VAD segments and processes the output with the LM, it achieves 24.27%WER. However, in the RTVE2022 test, the use of the LM slightly degrades the WER by 0.05%, and the best results are achieved using only VAD segments, 23.45% WER.

Table 6: *Resultant word error rate of the submitted systems in RTVE2020 and RTVE2022.*

System	Submission	RTVE2020	RTVE2022
W10	c3	26.68	25.25
W10LM	c2	25.70	24.87
VAD	c1	24.86	**23.45**
VADLM	p	**24.27**	23.50

The results achieved by the LM in our test sub-split from dev2 are presented in 7. While training uniquely with manually generated mistakes or with ASR's mistakes, the resultant model degrades the character error rate (CER) and WER and slightly improves the BLEU. Nevertheless, when training with both manual mistakes and ASR mistakes we obtain improvements in CER, WER, and BLEU. Additionally, to clarify the impact of the LM, in Table 8 we present a challenging example of the ASR caused by the ending music of a show. We can observe that both types of segmentation lead to mistaken outputs. Here the use of the LM provides at least correct Spanish outputs. Nevertheless, on some occasions, the LM is not capable to produce the right results. We believe that the LM can be improved by using more augmented data at the text level, but also by making the ASR face more challenging acoustic conditions while generating hypothesis: mixing the audio with background noise.

Table 7: *CER, WER and BLEU of the LM with respect of the data used at training.*

Data	CER%	WER%	BLEU	Sentences
Baseline	3.61	4.13	80.36	-
Manual	3.94	4.50	80.52	427k
Checkpoints	3.64	4.20	82.21	587k
All	**3.55**	**4.10**	**85.36**	1014k

Table 8: *Ending transcription of CN-20181208b file obtained with our four systems compared with text reference.*

System	Output
W10	*delante derecha delante **dtrás cundos** tres **e***
W10LM	*delante derecha delante detrás **cuando estés en** tres **e***
VAD	*delante derecha delante detrás **undos tress***
VADLM	*delante derecha delante detrás un dos tres*
Reference	*delante derecha delante detrás un dos tres*

Finally, it is important to mention that the ASR model has not stopped improving. In Figure 2 the improvement process is depicted, best results are obtained while using the biggest amount of domain data in training, for the moment we only used 166 hours. We strongly believe that there is still a margin for improvement by simply performing more iterations of the self-supervised method for domain adaptation. Or simply by using more complex decoding.

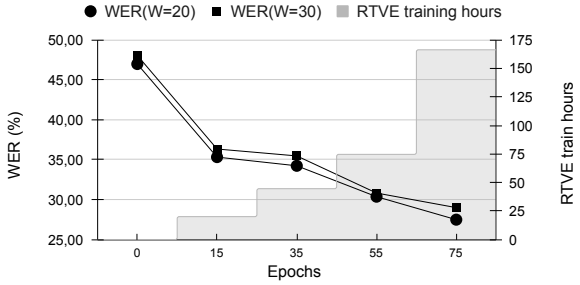

Figure 2: *WER across training epochs. In epochs 35 and 55, LR was restarted to 1.0 for the linear layers and 10^{-5} for Wav2Vec2. W stands for the length in seconds of the segmentation window used to get transcriptions.*

5. Conclusions

In this manuscript, we presented the description of the four systems we submitted in the Albayzin 2022 Speech-to-Text Transcription Challenge. We built an acoustic end-to-end ASR model based on the XLSR-53 architecture, that was pre-trained with unlabeled audio. We first fine-tuned this model to the Spanish language with CommonVoice and after adapted it to the TV broadcast domain with a self-supervised method. This technique uses an iterative pseudo-forced alignment algorithm based on the CTC paths and allowed us to recover 166 hours from RTVE2018 and RTVE2022. In addition, we explored the use of a machine translation model to correct the ASR output (LM). For that purpose, we fine-tuned a model that originally was used to translate from Catalan to Spanish. In our four submissions, we measure the impact of two types of segmentation and the use of the LM. Our best results are 24.27% and 23.45% WER on RTVE2020 and RTVE2022 test sets respectively.

6. Acknowledgements

We want to thank Martin Kocour from the Brno University of Technology, BUT Speech@FIT for sharing with us the VAD segments that were used in this submission.

7. References

[1] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2020, pp. 439–444.

[2] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings," in *Ninth International Conference on Spoken Language Processing*, 2006.

[3] M. Kocour, "Automatic speech recognition system continually improving based on subtitled speech data," Master's thesis, Brno University of Technology, Faculty of Information Technology, 2019. [Online]. Available: https://www.fit.vut.cz/study/thesis/22041/

[4] 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.

[5] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *International Conference on Speech and Computer*. Springer, 2020, pp. 267–278.

[6] F. López and J. Luque, "Iterative pseudo-forced alignment by acoustic ctc loss for self-supervised asr domain adaptation," 2022. [Online]. Available: https://arxiv.org/abs/2210.15226

[7] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," Tech. Rep., Technical report, OpenAI, 2022. URL https://cdn. openai. com . . . , Tech. Rep., 2022.

[8] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *arXiv preprint arXiv:1911.08460*, 2019.

[9] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *CoRR*, vol. abs/2006.11477, 2020. [Online]. Available: https://arxiv.org/abs/2006.11477

[10] 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.

[11] 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," 2021. [Online]. Available: https://arxiv.org/abs/2111.09296

[12] L. F. D'Haro and R. E. Banchs, "Automatic correction of asr outputs by using machine translation," in *Interspeech*, vol. 2016, 2016, pp. 3469–3473.

[13] A. Mani, S. Palaskar, N. V. Meripo, S. Konam, and F. Metze, "Asr error correction and domain adaptation using machine translation," 2020. [Online]. Available: https://arxiv.org/abs/2003.07692

[14] H. Futami, H. Inaguma, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the knowledge of bert for ctc-based asr," *arXiv preprint arXiv:2209.02030*, 2022.

[15] E. Lleida, A. Ortega, A. Miguel, V. Bazán, C. Pérez, M. Zotano, and A. De Prada, "RTVE2018 Database Description," 2018. [Online]. Available: http://catedrartve.unizar.es/reto2018/RTVE2018DB.pdf

[16] E. Lleida, A. Ortega, A. Miguel, V. Bazán-Gil, C. Pérez, M. Gómez, and A. De Prada, "RTVE2020 Database Description," 2020. [Online]. Available: http://catedrartve.unizar.es/reto2020/RTVE2020DB.pdf

[17] 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

[18] 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.

[19] F. Karamitroglou, "A proposed set of subtitling standards in europe," *Translation journal*, vol. 2, no. 2, pp. 1–15, 1998.

[20] "Long audio alignment overview," https://cmusphinx.github.io/wiki/longaudioalignment/, accessed: 2022-03-10.

[21] J. Umesh, M. Kocour, M. Karafiát, J. Švec, F. López, K. Beneš, M. Diez, I. Szöke, J. Luque, K. Veselỳ, L. Burget, and J. Ĉernockỳ, "BCN2BRNO: ASR System Fusion for Albayzin 2022 Speech to Text Challenge," in *Proc. IberSPEECH 2022*, 2022.

[22] 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.

[23] 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.

[24] 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.

[25] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 116–121. [Online]. Available: https://aclanthology.org/P18-4020

[26] J. Tiedemann, "Parallel data, tools and interfaces in opus," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

[27] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.