ViVoLAB System Description for the S2TC IberSPEECH-RTVE 2022 challenge

Antonio Miguel, Alfonso Ortega, Eduardo Lleida

ViVoLab, Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
amiguel@unizar.es, ortega@unizar.es, lleida@unizar.es

Abstract

In this paper we describe the ViVoLAB system for the IberSPEECH-RTVE 2022 Speech to Text Transcription Challenge. The system is a combination of several subsystems designed to perform a full subtitle edition process from the raw audio to the creation of aligned subtitle transcribed partitions. The subsystems include a phonetic recognizer, a phonetic subword recognizer, a speaker-aware subtitle partitioner, a sequence-to-sequence translation model working with orthographic tokens to produce the desired transcription, and an optional diarization step with the previously estimated segments. Additionally, we use recurrent network based language models to improve results for steps that involve search algorithms like the subword decoder and the sequence-to-sequence model. The technologies involved include unsupervised models like Wavlm to deal with the raw waveform, convolutional, recurrent, and transformer layers. As a general design pattern, we allow all the systems to access previous outputs or inner information, but the choice of successful communication mechanisms has been a difficult process due to the size of the datasets and long training times. The best solution found will be described and evaluated for some reference tests of 2018 and 2020 IberSPEECH-RTVE S2TC evaluations.

Index terms should be included as shown below.

Index Terms: Automatic speech recognition, Recurrent neural networks, Sequence-to-sequence models

1. Introduction

The advances in automatic speech recognition (ASR) in recent years and the current quality of state-of-the-art systems have made possible an increasing number of applications like automatic subtitling of multimedia contents, audio indexation and metadata generation among others. In this context, the 2018, 2020, and 2022 IberSPEECH-RTVE S2TC evaluations have contributed to establishing baselines for ASR in Spanish [1]. The challenge proposed the extraction of text content from several audios from the broadcast domain to assist metadata extraction and subtitling specialists. The difficulty of the tasks ranges from good quality of sound programs, turn-based interviews, and debates to very challenging noisy scenarios with speech overlap and music and noise distortions. In addition to the noise in broadcast audio, there are more challenges like spontaneous speech and different Spanish accents.

The ViVoLAB system combines several subsystems for a global design objective of providing a full subtitle edition process from the raw audio to the creation of aligned subtitle transcribed partitions. The subsystems include the following parts. A phonetic recognizer that takes the waveform and outputs a sequence of recognized phonemes. A phonetic subword recognizer, based on the tokenization of phoneme sequences to form groups of one or more phonemes, and allows the capture of context and co-articulation more effectively than the phoneme recognizer. A speaker-aware subtitle partitioner, which has been trained with subtitles and segments from audio datasets to predict the limits of a subtitle from the raw audio and the previous subsystem outputs. The final ASR module in this work is a sequence-to-sequence translation model working with the previous subsystems as input and the orthographic tokens as output and will produce the desired transcription estimation in a recursive feeding process. Optionally, a diarization step with the previously estimated segments can be applied using any of the recent systems developed in our group [2]. Additionally, we use recurrent language models to improve results for steps that involve search algorithms like the subword decoder and the sequence-to-sequence model. In general, we allow all the systems to access previous system outputs or access their inner layers. Nevertheless, the choice of successful communication mechanisms has been a difficult process due to the size of the datasets and long training times. The best solution found will be described and evaluated for some reference tests the IberSPEECH-RTVE S2TC evaluation.

This paper is organized as follows. In Section 2 we review state-of-the-art concepts referenced in our proposal. Section 3 describes in more detail all the subsystems. Experimental results are shown in section 4 and finally, conclusions are presented.

2. State of the art

The first systems that incorporated neuronal networks in speech recognition were hybrid systems [3] which still needed the sequence modeling provided by Hidden Markov Models (HMMs). The systems increased in size and depth of the networks involved [4], but the basic units were phonemes and the lexicon stored in a dictionary of prior knowledge, which required expert annotation. The performance was later improved by modeling the sequence using recurrent neural networks instead of HMMs [5]. Although the units were still phonemes, the Connectionist temporal classification (CTC) as loss function [6] to optimize the system had the advantage of simplifying the processing pipeline both at training and testing time.

The next step towards fully automated training was proposed in [7] where the transcription in characters was the objective of the network. The transcription to letters or characters had the advantage of a simple development of the systems but the size of the units started to grow progressively, to capture context-dependent features in specialized units [8] or even words [9][10]. Word models provide good performance but they are less flexible, for example, to define new words. The use of word piece selection also called tokens or subword units has been proven to be a better strategy both in terms of flexibility and performance [11]. This type of technique has been used successfully in large-scale language models using transformers. The type of backbone architecture of ASR systems has also evolved during the last decade with examples of recurrent neural networks [5], convolutional networks [12], and more recently transformers [13]. Finally, the use of unsupervised data [14] or partially labeled data [15][16] has been shown in recent works to improve the performance of the systems when unlabeled data are available. Unlabeled data [14, 17, 18] can be used to learn feature extractors, which take the waveform directly and do not need a frequential analysis designed by an expert, with the advantage that this process only requires large amounts of data which are easy to collect nowadays. To train these systems without labels some strategies have been proposed like predicting which segment continues a previous one [19], or predicting the labels of a previous clustering process [17].

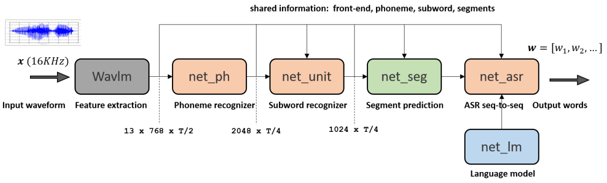

Figure 1: *Vivolab system overview.*

3. System description

The ViVoLAB system is a composition of some modules interacting as shown in Figure 1. We allow the systems to access previous system outputs or access their inner layers. This may help the development of a complex task like this since they can be trained and evaluated consecutively. In addition, having access to useful internal variables and intermediate results provides probe points in the system. The following sections describe each module in more detail.

3.1. Phoneme recognition

The phonetic recognition system takes the raw audio samples at 16kHz and outputs phoneme posterior probabilities every 10ms. The number of classes is 27 including a class for silence and Spanish language phonemes [20, 21]. During the training phase, Kaldi aligned phonetic transcriptions [22], were used as target labels for a cross-entropy loss function. The training process consisted in several iterations where only the best aligned segments were used to train the next model.

In table 1, we can see a summary of the type of layers and parameter size for each system. Other layers like ReLU activations or batch normalization layers are neglected for clarity. The phoneme network, net_ph, first uses the unsupervised model Wavlm [18] to extract feature vectors every 20ms. The model is WavLM Base+ which has 12 Transformer encoder layers with 8 attention heads and represents the signal with internal vectors of dimension 768. The output can be considered a down-sampled version with factor 2 with respect to the labels since there is a representation vector every 20ms. The hidden inputs and final layer of the Wavlm model are combined and down-sampled again with a factor of 2 to achieve higher computation speeds since the LSTM is operating at a factor of 4 times lower rate than in a standard speech application This is done in a Conv1d block with two layers and an output dimension of 1024. Then a two-layer bidirectional LSTM is applied to obtain 2048 channels, which are then linearly combined to the final embedding. To evaluate the cost we need to upsample a factor of 4 this signal. For example, if the original signal corresponds to T labels every 10ms, the output of the LSTM will have $T/4$ frames, then the need for an upsampling step.

Table 1: *Description of the layers used in the different subsystem networks. For each network, the main parts are listed with details about their configuration output size and the number of parameters. The time length T indicates the reference number of frames with a window advance of 10ms and N indicates the maximum sequence length. *Note that the Wavlm parameters are not added to the total number.*

System	Layer	Comment	Size	Params
net_ph	Input	Wavlm	13 x 768 x T/2	94.7M*
	2xConv1d	down. x2	1024 x T/4	9.4M
	2xLSTM	bidirectional	2048 x T/4	40.0M
	Linear	dim reduction	512 x T/4	1.0M
	Linear	up x4	512 x T	1.0M
	Linear	logits	27 x T	14.9k
(Total)				51.4M
net_unit	Input	Wavlm	13 x 768 x T/2	94.7M*
	2xConv1d	down x2	1024 x T/4	9.4M
	4xLSTM	bidirectional	2048 x T/4	109.1M
	Linear	dim reduction	512 x T/4	1.0M
	Linear	up x4	512 x T	1.0M
	Linear	logits	8000 x T	14.9k
(Total)				120.5M
net_asr	Input	Wavlm	13 x 768 x T/2	94.7M*
	2xConv1d	down x2	1024 x T/4	9.4M
	Linear	ph+unit	1024 x T/4	3.7M
	2xEncoder	FF+MHA	1024 x T/4	14.7M
	Embedding	prev. out	1024 x N	8.2M
	6xDecoder	FF+MHA+xMHA	1024 x N	62.9M
	Linear	output	8000 x N	8.2M
(Total)				107.1M
net_lm	Input	-	8000xN	-
	Embedding	input	300xN	2.4M
	4xLSTM	unidirectional	1500xN	64.8M
	Linear	output	8000xN	2.4M
(Total)				70.0M
net_seg	Input	-	13 x 768 x T/2	94.7M*
	2xConv1d	down x2	1024 x T/4	9.4M
	Linear	ph+unit	1024 x T/4	3.7M
	4xEncoder	FF+MHA	1024 x T/4	16.8M
	Linear	logits	2 x T/4	2.0k
(Total)				29.9M

The objective of the next Linear layer is to obtain a final embedding dimension of 512 at the label rate, then we use a linear layer to project the data to dimension $512 \cdot 4$ and we reshape the output of the layer to a tensor of shape $512xT/4x4$, where the final dimension is the upsampling factor. Finally, we reshape the last two dimensions together to interleave the new 4 samples per original sample to obtain $512xT$ [23]. The interleave will be correct if the factor is the last dimension since torch and numpy are row-major. For convenience, these reshaping operations are done in a single step after the linear layer.

The output of the Linear layer after the LSTM is used as output for other systems since it corresponds to the slower rate of a vector every 40ms, a factor 4 of reduction, which can help the rest of the systems to reduce computation time. Nevertheless, if we desire a prediction of the recognized phonemes we will continue with the final linear layer to obtain the predictions for the phoneme classes every 10ms. In the top part of Figure 2, we can observe the log-posterior probabilities of an example short segment and we can that thanks to good noise conditions we can see a clear path to align this sequence to the best phonemes.

3.2. Phonetic subwords

In this system, we define a higher level of abstraction in the ASR. Now the objective is to recognize subword units by using the raw input processed by the Wavlm network and the support of the previous phoneme recognizer. The subwords are automatically obtained using a BPE(Byte Pair Encoding) tokenizer [24] with the phonetic transcription of the training data instead of standard text. The objective is to control the number of units and the quality of the representation since, as we have seen in Figure 1, this module is not providing the final output but intermediate useful information. Then the recognition labels are now a set of subword units, in this work 4000 different combinations with begin and end of word special markings. Given the large number of units, many frequent words are described as a unique unit. As an example here we display the units corresponding to the phonetic transcription of some random words:

- ahora: `_aora_` (frequent words are complete units)
- circulaba: `_Tirkul aBa_` (most verbs share suffixes)
- parametrizacion: `_par am etr iTaTjon_`

The architecture of the subword network is very similar to the phonetic network and the training process is simplified since this type of subword unit can use the same alignment previously obtained for the phoneme model. It also operates the LSTM every 40ms which alleviates its high computational cost, since it has four layers. In this case, the LSTM input is the combination of a hidden vector from the net_ph and the output of the convolutional downsampling of the Wavlm output, which was also done in the previous module. The motivation for the extra effort of processing again the output of the Wavlm to have multiple combinations of the Wavlm hidden layers is related to the findings exposed in the original paper [18] where the authors show that different tasks can have different combinations of weights. Then, the objective here is to try to capture a new level of abstraction in the recognition process, the formation of words from subword units. The output of this system to the next module can be the hidden vectors after the LSTM with a lower rate or the final decoded unit sequence.

Finally, the output of this subsystem can be used by the next module, the word recognizer, but there is a problem that is related to the type of model we have decided for the ASR, a sequence-to-sequence. The problem is due to the process of autoregressive generation of the output, which in extreme situations, high noise or unseen situations, the generation process can fall into loops that generate many insertions and degrade the performance. To alleviate this issue, some proposals have been added to the system during the development of the challenge. The one that affects this module is to define an additional state for the subword units similar to a HMM (Hidden Markov Model) and also related to the null state in CTC models, an ending state is added to all the subword units. The total number of classes in this module is 8000. The ending state is mandatory to finish a subword and it is visited only once (it does not have self-transition). This way by looking at the ending state visits while decoding, we can have better information about the limits of the units. Before this modification, the subword system was not able to tell the difference between pronouncing several consecutive times the same word. At the bottom of Figure 2, we can see the log-posterior probabilities of the same example short segment where we can see that the frequent complete words are easily aligned and that the final state learns to be activated only once at the end of each unit, helping with the segmentation and search process.

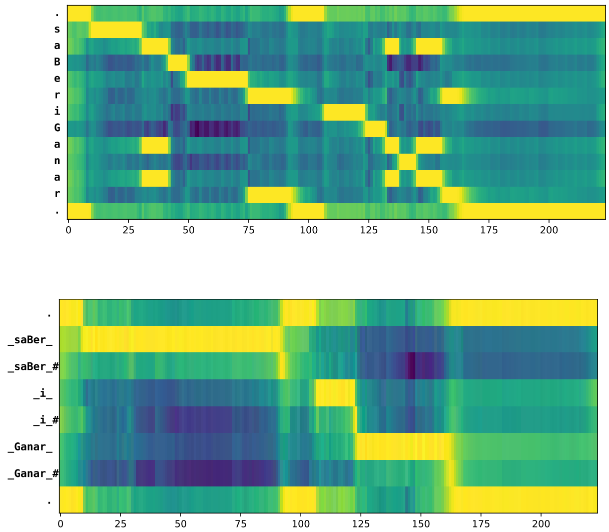

Figure 2: *Examples of phoneme (top) and subword (bottom) recognizer log-posterior probabilities obtained every 10ms for the first transcribed segment of RTVE 2018 audio SG-20180520 with Spanish transcription: 'saber y ganar'. The silence model is '.' and symbol '#' indicates end state of subword.*

3.3. Orthographic token based recognizer

The ASR core subsystem is the sequence-to-sequence network. The architecture is displayed in Table 1 and we can see that the first layers to process the output of Wavlm and the connection to previous subsystems are similar to what we have seen in the subword network. They integrate all available information so that the transformer encoder can process it. The low-rate embeddings of the phoneme and unit networks are concatenated to the downsampled signal coming from the Wavlm network. Then they are projected to a lower dimension of 1024 so that the transformer can operate with more moderate dimensions and at a lower rate which helps the computational cost. This system uses 8k tokens learned with the BPE tokenizer. We have tested two types of text preparation to learn the tokenizer and therefore the target labels: the first one had minimal processing and was able to include punctuation symbols and rich text format, the final system selected for this paper had a better performance since we applied several steps of text normalization and final text has accent marks and all other punctuation symbols are removed, numbers and roman numbers are converted to text among other operations.

The transformer encoder layers are composed of two blocks: Feed-Forward block and MHA block (Multi Head Self-Attention) with 12 heads [25], both of them use residual connections to modify the signal iteratively. The encoder has 2 layers and the decoder 6 layers that take as input the encoder output and the autoregressive output which is encoded in an embedding layer. The decoder layers have a Feed-Forwad block and two MHA blocks one for the encoder output and the other for the iterative processing of the prediction starting from the previous output. This last MHA has to be causally masked to avoid future information finding a trivial path to the loss function.

To fight the loop generation problem and provide a more stable reference to the decoder part of the transformer we propose a second alternative to the seq-to-seq network. The idea is to use a decoded subword sequence as additional input to the the decoder. The example in Figure 2 shows us that such best sequence decoding: '_saBer_ _i_ _Ganar_', could bring useful information to the system, and now the decoder can fix the attention in this sequence of tokens in addition to information captured by the encoder. To incorporate the information into the new version of the system we add a third MHA cross-attention module that takes the whole input to produce the outputs similarly to the MHA that takes the encoder output. We will refer to these models as 'net_asr (a)' the previous model and 'net_asr (b)' for the model that takes the decoded output and has a third cross-MHA.

3.4. Subtitle partitioner

The segment prediction system has the objective of providing useful segments for the ASR operation and, optionally, a former diarization step. Thanks to this task, the system is closer to a fully automatic subtitle operation. The segments generated help the ASR decoder to avoid a sliding window scheme since it can process the segments provided by the segmentation module with defined limits.

The system is similar in the inputs to the ASR module. It takes the Wavlm representation, the phoneme hidden variables, and the unit recognizer low-rate hidden variables. With this information, it applies only the decoder steps of a transformer and finally produces a binary output predicting if the frame corresponds to the same segment or a different segment. To train the system we generate three types of a situation artificially: the segment does not have a segment frontier and it is speech, the segment is fully noise and discarded, and the segment has an optimal subtitle cut that needs to be predicted. The training data to solve these situations can be generated artificially by collating different segments from the database or predicting the end of the subtitle in longer audio from a given position since we have that limit in the database. The limits obtained with this method are used by the ASR to generate the decoded words for each segment y a fully automatized process.

3.5. Language model

As language model, we use a simple network based on a causal four layer LSTM trained using the set of tokens previously obtained for the seq-to-seq model. The model is later applied during the search process and mixed with a weight of 0.08 meaning that we have not achieved significative gains by using the language model.

4. Experiments

The IberSPEECH-RTVE 2018, 2022, and 2022 challenge datasets have been released by the collaboration of Corporacion Radio Television Española (RTVE), the main public service broadcaster in Spain, and the RTVE Chair at the University of Zaragoza [1]. They are a collection of different shows from various styles and genres. The training material comprises around 500h and two dev partitions with 57h and 15h. In 2022 an additional training set has been released with 300h. The transcriptions come from the broadcast subtitles and they are sometimes misaligned or the text is not exact but a human interpretation.

The acoustic models have been trained using several datasets: Albayzin [26], a phonetically balanced corpus with 12h, Speech-Dat-Car [27], a corpus recorded in a car in different driving conditions 18h, the Domolab [28] corpus, recorded in a domotic environment 9h length, TCSTAR [29] transcriptions of Spanish parliament sessions 111h and Commonvoicees [30], a multilingual corpus that employs crowdsourcing for both data collection and data validation, 400h. RTVE 2018 train, dev1, dev2 RTVE 2022 train more than 800h. In addition, more training material was added from a diverse scrap of online videos and social networks for a total of 10kh. The training data is later filtered after performing forced alignment and selecting transcribed segments with sufficient quality for training. To augment data MUSAN [31] and other noises and music data downloaded from the Internet were used. Artificially generated impulse responses were used to simulate different acoustic environments [32]. The language model was trained using Spanish Wikipedia, RTVE challenge provided subtitles, and text news obtained from different Spanish newspapers. In Table 2 we show average results for all challenges for the two proposed systems. We observe the system using an extra cross-MHA per decoder layer using the best-decoded unit sequence is the best performing system. Results are encouraging since their analysis has shown that we have critical points to improve, especially in the segment generation which is now deleting a great amount of audio with transcription and that generates an excessive number of deletions.

Table 2: *WER % of the submitted systems evaluated on RTVE2018 RTVE2020 RTVE2022 sets.*

System	RTVE2018	RTVE2020	RTVE2022
net_asr (a)	17.57	22.13	20.87
net_asr (b)	**16.49**	**21.86**	**20.57**

The computational cost of the system is smaller than models of similar layers and sizes in the number of parameters since all the sequence models operate at a factor of 4 smaller rates. The module that consumes most of the time is the net_asr transformer. We have measured on a Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz with a Nvidia RTX3090 graphic card and the computation time to duration ratio is smaller than 7% for both net(a) and net_asr(b).

5. Conclusions

In this paper, we have described the ViVoLAB system for the S2TC IberSPEECH-RTVE 2022 challenge. The system is a full ASR engine that takes a raw 16kHz signal and produces different outputs from phoneme level recognition, subword units, subtitle estimated limits, and estimate the sequence of words. The key aspects of the systems are modularity, which will allow a continuous module upgrading policy in the future to benefit the rest of the subsystems, the access to useful probe information during the development of the model, which also helps other related applications like subtitle partitioning and diarization, the specific design which tries to reduce the impact of generation loops in seq-to-seq architectures by using end state in subword units and, optionally, the decoded subword sequence as additional input to the transformer architecture, fast operation at search time which can achieve lower than 7% computation to duration time ratios. The system has been evaluated in 2018, 2020, and 2022 S2TC IberSPEECH-RTVE evaluations.

6. Acknowledgements

This work was supported in part by the European Union's Horizon 2020 research and innovation program under Marie Skłodowska-Curie Grant 101007666; in part by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU" / PRTR underGrants PDC2021-120846-C41 PID2021-126061OB-C44, and in part by the Government of Aragon(GrantGroup T36 20R).

7. References

[1] E. Lleida, A. Ortega, A. Miguel, V. Bazán-Gil, C. Pérez, M. Gómez, and A. de Prada, "Albayzin 2018 evaluation: The iberspeech-rtve challenge on speech technologies for spanish broadcast media," *Applied Sciences*, vol. 9, no. 24, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/24/5412

[2] I. Viñals, P. Gimeno, A. Ortega, A. Miguel, and E. Lleida, "ViVoLAB Speaker Diarization System for the DIHARD 2019 Challenge," in *Proc. Interspeech 2019*, 2019, pp. 988–992.

[3] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. USA: Kluwer Academic Publishers, 1993.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] A. Graves, "Sequence transduction with recurrent neural networks," in *In Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012.

[6] F. S. G. F. Graves, A. and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In ICML, 2006*, 2006.

[7] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[8] H. Liu, Z. Zhu, X. Li, and S. Satheesh, "Gram-ctc: Automatic unit selection and target decomposition for sequence labelling," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2188–2197.

[9] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[10] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," *arXiv preprint arXiv:1703.07754*, 2017.

[11] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," *arXiv preprint arXiv:1902.01955*, 2019.

[12] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6460–6464.

[13] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[15] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *arXiv preprint arXiv:2005.09629*, 2020.

[16] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[17] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[18] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[19] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[20] J. Lombart, A. Miguel, and E. Lleida, "Articulatory feature extraction from voice and their impact on hybrid acoustic models," in *Advances in Speech and Language Technologies for Iberian Languages*. Springer, 2014, pp. 138–147.

[21] I. Viñals, D. Ribas, V. Mingote, J. Llombart, P. Gimeno, A. Miguel, A. O. Giménez, and E. Lleida, "Phonetically-aware embeddings, wide residual networks with time-delay neural networks and self attention models for the 2018 nist speaker recognition evaluation." in *Interspeech*, 2019, pp. 4310–4314.

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

[23] Y. Sugawara, S. Shiota, and H. Kiya, "Convolutional neural networks without any checkerboard artifacts," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1317–1321.

[24] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] F. Casacuberta, R. Garcia, J. Llisterri, C. Nadeu, J. Pardo, and A. Rubio, "Development of spanish corpora for speech research (albayzin)," in *Workshop on International Cooperation and Standardization of Speech Databases and Speech I/O Assesment Methods, Chiavari, Italy*, 1991, pp. 26–28.

[27] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, "Speechdat-car. a large speech database for automotive environments." in *LREC*, 2000.

[28] R. Justo, O. Saz, V. Guijarrubia, A. Miguel, M. I. Torres, and E. Lleida, "Improving dialogue systems in a home automation environment," in *1st International ICST Conference on Ambient Media and Systems*, 2010.

[29] H. Van den Heuvel, K. Choukri, C. Gollan, A. Moreno, and D. Mostefa, "Tc-star: New language resources for asr and slt purposes." in *LREC*, 2006, pp. 2570–2573.

[30] 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.

[31] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[32] M. A. . B. J. Diaz-Guerra, D., "gpurir: A python library for room impulse response simulation with gpu acceleration," *Multimed Tools Appl*, vol. 80, no. 24, 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-09905-3