Pre-trained Speech Representations as Feature Extractors for Speech Quality Assessment in Online Conferencing Applications

Bastiaan Tamm[1], *Helena Balabin*[1], *Rik Vandenberghe*[1], *Hugo Van hamme*[2]

[1]Laboratory for Cognitive Neurology, KU Leuven, Belgium
[2]Processing Speech and Images, KU Leuven, Belgium

bastiaan.tamm@kuleuven.be

Abstract

Speech quality in online conferencing applications is typically assessed through human judgements in the form of the mean opinion score (MOS) metric. Since such a labor-intensive approach is not feasible for large-scale speech quality assessments in most settings, the focus has shifted towards automated MOS prediction through end-to-end training of deep neural networks (DNN). Instead of training a network from scratch, we propose to leverage the speech representations from the pre-trained wav2vec-based XLS-R model. However, the number of parameters of such a model exceeds task-specific DNNs by several orders of magnitude, which poses a challenge for resulting fine-tuning procedures on smaller datasets. Therefore, we opt to use pre-trained speech representations from XLS-R in a feature extraction rather than a fine-tuning setting, thereby significantly reducing the number of trainable model parameters. We compare our proposed XLS-R-based feature extractor to a Mel-frequency cepstral coefficient (MFCC)-based one, and experiment with various combinations of bidirectional long short term memory (Bi-LSTM) and attention pooling feedforward (AttPoolFF) networks trained on the output of the feature extractors. We demonstrate the increased performance of pre-trained XLS-R embeddings in terms a reduced root mean squared error (RMSE) on the ConferencingSpeech 2022 MOS prediction task.

Index Terms: speech recognition, MOS prediction, speech quality assessment

1. Introduction

Given the ubiquitous dependence on online conferencing applications in recent years, a reliable and automated assessment of speech quality has become increasingly important. Various factors such as jitter, latency, echo, packet loss, and distortion may affect listening, talking and conversational quality [1]. Ultimately, a reliable assessment of speech quality can help to guide and evaluate speech enhancement approaches (e.g., those presented in the ConferencingSpeech 2021 challenge [2]). Typically, speech quality assessment approaches are grouped into intrusive and non-intrusive methods, depending on whether they operate on the original unprocessed or the processed audio signal. Since objective metrics such as the speech-to-reverberation modulation energy ratio (SRMR) do not necessarily correlate with the perceived speech quality [3], human judgements in the form of Absolute Category Ratings (ACRs) are commonly used in non-intrusive settings.

The resulting ratings are then used to derive mean opinion score (MOS) values as a final subjective assessment of the speech quality of a given audio sample. However, the collection of human judgements is time- and labor-intensive and not feasible for large-scale evaluations of speech quality in online conferencing applications. Furthermore, efforts have been dedicated towards Machine Learning (ML) approaches for the automated prediction of MOS values for given speech fragments [4, 5], for example based on long short-term memory (LSTM) networks [6]. More recently, unsupervised pre-trained speech models such as wav2vec 2.0 [7] have made it possible to leverage pre-trained representations for specific fine-tuning tasks such as MOS prediction. This is typically realized by updating all model parameters using task-specific data and objective functions. While such models are typically pre-trained on English speech, there are also multilingual variants like XLS-R [8], consisting of models with up to two billion parameters trained on 436,000 hours of unlabeled speech.

Nonetheless, fine-tuning such a large number of model parameters is computationally intensive and thus not always an expedient solution. Moreover, while benchmarks (e.g., the combination of multiple automatic speech translation, automatic speech recognition and speech classification tasks that was used to evaluate wav2vec 2.0 [7]) aim to assess the generalizability of the performance of fine-tuning pre-trained language or speech representations on downstream tasks, they do not always translate into robustly improved performance on tasks in a specific domain or language [9]. As a result, the use of pre-trained models as fixed feature extractors for much smaller task-specific ML architectures might pose a more effective alternative to computationally intensive fine-tuning procedures.

That is why in this paper, we evaluate the potential of using the XLS-R model in a fixed feature extraction rather than a fine-tuning setting for MOS prediction for the ConferencingSpeech 2022 challenge. Our contributions can be summarized as follows. First, in addition to the two baselines provided in the ConferencingSpeech challenge, we propose a Mel-frequency cepstral coefficient (MFCC)-based approach serving as an additional baseline. Secondly, we train a MOS prediction model consisting of a bidirectional LSTM (Bi-LSTM) and an attention pooling feedforward (AttPoolFF) network based on the XLS-R feature extractor. Additionally, we test various combinations of the BiLSTM and AttPoolFF components placed on top of both the XLS-R and MFCC feature extractors. We show that the XLS-R feature extractor does not only outperform the classic MFCC feature extractor, but it also results in competitive model performances with regards to the root mean squared error (RMSE) on data from the ConferencingSpeech 2022 challenge [10].

2. Methods

2.1. Datasets

The dataset provided in the ConferencingSpeech 2022 challenge comprises four corpora, each with respective MOS la-

bels in a rating range of 1–5, which are derived from ACRs based on the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) recommendation P.808 [11]. The only exception is the IU Bloomington corpus, which follows ITU-R BS.1534 [12] and has a range of 0–100.

2.1.1. Tencent

The Tencent corpus consists of around 14,000 Chinese speech clips from a total of 1,089 speakers, including clips with and without reverberation. The length of the speech clips ranges from 5s to 13.5s.

2.1.2. Non-intrusive speech quality assessment (NISQA)

The NISQA corpus [13] is a publicly available collection of more than 14,000 English and German speech clips with real as well as simulated noise.

2.1.3. Indiana University (IU) Bloomington

The IU Bloomington corpus encompasses a total of 36,000 English speech samples from the Voices Obscured in Complex Environmental Settings (VOiCES) [14] and COnversational Speech In Noisy Environments (COSINE) [15] datasets. All speech clips are between three and six seconds long.

2.1.4. Public switched telephone network (PSTN)

The PSTN corpus consists of roughly 80,000 English speech samples based on LibriVox with a length of 10 seconds each, containing both clean samples as well as samples with artificially added background noise.

2.2. Dataset Division

The original dataset split for training and development provided by the challenge organizers consists of 80% of the Tencent corpus and 95% of the PSTN corpus. The remaining 20% and 5% of these corpora along with some samples from an additional corpus called TUB compose the test split.

We take two approaches to dataset division. In the first division, we aim to maximize the performance on the challenge's test split. Since the proportion of TUB samples is rather small (only 10% of the test set), we expect the distribution of the test split to be similar to that of the original training and development split. For this reason, we only include samples from the Tencent and PSTN corpora in this approach.

In the second division, we attempt to compare the generalizability of the various models by training and evaluating the models on a larger pool of datasets. For simplicity, this has been done by shuffling the samples in all corpora (IU Bloomington, NISQA, PSTN, Tencent) and using 85% for the training set and 15% for the validation set.

In order to compare model performance in a correct way, we construct the first division as subsets of the second division's training and validation sets, where only Tencent and PSTN samples are included. Since training is performed on the subsets, there is no overlap between the training samples from one approach and the validation samples from another approach.

2.3. Preprocessing

The raw audio is processed with a two-step procedure in order to generate the features for the MOS prediction-specific model architectures (see Section 2.4).

2.3.1. Feature Extraction

First, feature extraction is performed on the raw audio. For MFCC calculation, we use the implementation by torchaudio [16] with the default parameters and a sample rate of 16 kHz. The XLS-R feature extraction is based on the *facebook/wav2vec2-xls-r-300m* model available at the HuggingFace [17] model hub. Note that no fine-tuning is performed on the feature extraction modules.

2.3.2. Input Normalization

After the features are extracted for the entire dataset, the mean and variance of each channel are calculated for the training split. These parameters will be used to normalize the input data.

2.4. Models

We compare a total of six different model architectures, namely two baselines provided in the ConferencingSpeech 2022 challenge and four novel architectures based on combining different feature extractors and MOS prediction models.[1]

2.4.1. Baseline

The challenge organizers have provided two baseline models based on the NISQA speech quality prediction model [13]. The overall architecture consists of three parts. First, framewise features are calculated using a feed-forward or convolutional network. Next, the features are updated using a time-dependent network, e.g., a (Bi-)LSTM or a transformer model. Finally, the updated features are pooled and mapped to a one-dimensional output.

The first baseline model (Baseline1) uses a deep feed-forward network followed by an LSTM network and average pooling head. The model was trained for a total of 50 epochs on the entire development set. The second baseline model (Baseline2) uses a convolutional neural network (CNN) followed by a self-attention network and attention-pooling head. This model was trained on the entire development set for 80 epochs.

2.4.2. Proposed Models

The proposed model architectures are illustrated in Figure 1. Each model consists of two parts: a frozen feature extraction module (see Section 2.3) and a tunable regression head. We have combined two possible feature extractors (MFCC, XLS-R) with two possible regressors (AttPoolFF, Bi-LSTM + AttPoolFF) which makes four total models.

The goal of these four models is to demonstrate the power of state-of-the-art pre-trained speech embeddings (XLS-R) compared to more traditional features (MFCC) and training embeddings from scratch (baseline). Furthermore, we investigate if additional temporal support through a Bi-LSTM improves performance.

The regressor input is a sequence of 384 extracted features (roughly 8 seconds of audio). The Bi-LSTM network (if included in the model architecture) uses 2 layers and a size of 32 in each direction. The attention pooling network with feedforward connection (AttPoolFF) comes from the NISQA model [13] and uses a hidden size of 64. We apply batch normalization before the Bi-LSTM network and dropout of 0.1 after each layer. Subsequently, the outputs from the AttPoolFF module are mapped to the range (0,1) with a sigmoid function.

[1] Our Code: `github.com/lcn-kul/conferencing-speech-2022`

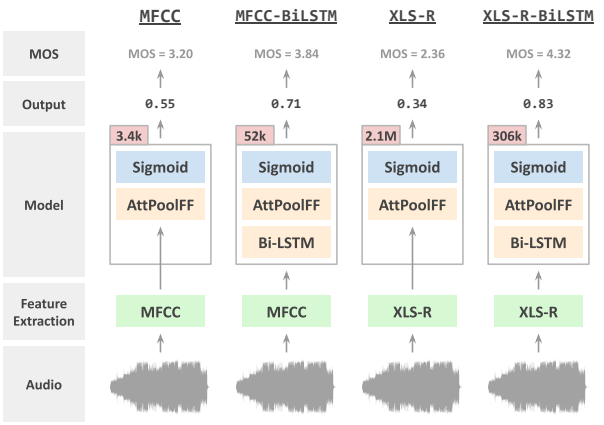

Figure 1: *Overview of the model components of all four model variants introduced in this paper. The MFCC and XLS-R networks (see green boxes) are used as fixed feature extractors. The Bi-LSTM and Attention-Pooling layers are trained on the challenge data. The resulting number of trainable parameters for each model are listed in the red boxes.*

We refer to these final outputs as normalized MOS values. This intermediate normalization step ensures that the output range of the AttPoolFF layer is unrestricted. During the evaluation of the model, the normalized MOS predictions can be mapped to the original 1–5 MOS range.

2.5. Evaluation

Model performance is assessed on a total of three datasets. First, we evaluate our proposed models on our own validation split (see Section 2.2). Secondly, to estimate the performance on the test set, we also evaluate our models on the subset of validation data that come from the Tencent and PSTN corpora. Lastly, we report the test performance of all models on the test data provided by the challenge organizers. We report the RMSE, Pearson correlation coefficient (PCC) and Spearman's rank correlation coefficient (SRCC) on the aforementioned validation and test datasets. More specifically, these metrics are calculated by first mapping the normalized MOS values (see Section 2.4) back to the original range of 1–5.

2.6. Training Details

The models are implemented using the PyTorch (v.1.11.0) and PyTorch Lightning (v.1.5.10) libraries in Python 3.9. Training is performed using the PyTorch Lightning trainer. The network is trained using the ADAM optimizer [18], a batch size of 64, and an MSE loss. We use a cyclical learning rate scheduler [19] with a base learning rate of 10^{-3} and a maximum learning rate of 10^{-2}. Each model is trained with a batch size of 64 for a total of 50 epochs, and the model with the lowest validation loss is selected. Each model was trained using an NVIDIA GeForce GTX 1660 Ti graphics card with 6 GB memory.

3. Results

3.1. Validation Results

Table 3 shows that the XLS-R Bi-LSTM model that was trained on the Tencent/PSTN samples outperforms the baseline with a RMSE reduction of 0.0206 (4.1% improvement) on the respective validation set. It is noteworthy that the baseline's training set included these samples, so it would have a slight advantage in this comparison. The MFCC-based models are substantially worse than the baseline model.

Surprisingly, the XLS-R model without a Bi-LSTM module performs much worse than its Bi-LSTM counterpart. We investigated this result by training the same model on the larger split including all four corpora. This resulted in rapid convergence and better performance (PCC: 0.8426, SRCC: 0.8372, RMSE: 0.5042 on validation split of reduced dataset). We therefore conclude that the low performance is likely because the reduced dataset is too small for the number of model parameters.

For the PCC and SRCC metrics, shown in Tables 1 and 2 respectively, the baseline still outperforms our best model by 0.0153 (1.8% better) and 0.0169 (2.0% better) respectively. Further investigation is required to determine the reason why the XLS-R-based models perform similar to or better than the baseline on RMSE but worse for PCC and SRCC.

3.2. Test Results

The final row of Tables 1 and 3 shows the challenge results for the PCC and RMSE metrics respectively. Our XLS-R Bi-LSTM model outperforms the baselines for both metrics. For the PCC metric this is an improvement of 0.261 (47.4% better) over Baseline1 and 0.088 (12.2% better) over Baseline2. For the RMSE metric, our model shows a reduction of 0.401 (53.8% better) and 0.199 (36.6% better) over the baselines.

Our model also outperforms the other 17 competitors.[2] Compared to the next best model, our PCC and RMSE metrics improve by 0.015 (1.9% better) and 0.092 (21.1% better) respectively. Finally, for the smoothed RMSE metric, which is the primary evaluation metric, we improve by 3.9% compared to the next best model and by 37.3% and 20.3% compared to the baselines.

The large gap in RMSE metrics is due to outstanding performance on the unseen TUB dataset. For the PSTN and Tencent corpora, our model performs comparably to the next best model (1.7% better and 5.6% worse). But for the TUB corpus, our model performed 27.4% better than the next-closest competitor for that corpus and 42.9% better than the overall second-place model [10]. This suggests that the proposed model performs exceptionally well on unseen data. Further research is required to validate this claim.

4. Conclusions

In this paper, we have presented our submission for the ConferencingSpeech 2022. Our model outperforms the baselines by 53.8% and 36.6% and also the next best competitor by 21.1% on the RMSE metric. Through this work, we are able to demonstrate the effectiveness of using pre-trained wav2vec-based XLS-R in a feature extraction setting.

In future work, we aim to compare the generalizability of the wav2vec-based models versus the baseline model by evaluating them on larger unseen datasets. Finally, we would investigate the use of a Transformer instead of a Bi-LSTM in the model architectures.

5. Acknowledgements

This work was supported by KU Leuven Bijzonder Onderzoeksfonds (BOF) under grant C14/21/109.

[2]Results: `tea-lab.qq.com/challenge_rankings_2022.pdf`

	Baseline		**MFCC**		**XLS-R**	
	Baseline1	Baseline2	MFCC	MFCC-Bi-LSTM	XLS-R	XLS-R-Bi-LSTM
Validation PCC All Corpora	0.7054	0.8182	0.5763	0.7294	0.7835	**0.8398**
Validation PCC PSTN & Tencent	0.8017	**0.8770**	0.7114	0.8049	0.8058	0.8617
Test PCC PSTN & Tencent	0.551	0.724	/	/	/	**0.812**

Table 1: *Pearson correlation coefficient (PCC) of all models on different splits of the datasets provided in the ConferencingSpeech 2022 challenge. For each dataset, the best performance (i.e., the model with the highest PCC) is highlighted in bold.*

	Baseline		**MFCC**		**XLS-R**	
	Baseline1	Baseline2	MFCC	MFCC-Bi-LSTM	XLS-R	XLS-R-Bi-LSTM
Validation SRCC All Corpora	0.6995	0.8186	0.5769	0.7236	0.7755	**0.8323**
Validation SRCC PSTN & Tencent	0.7954	**0.8712**	0.7068	0.7986	0.7976	0.8543
Test SRCC PSTN & Tencent	/	/	/	/	/	/

Table 2: *Spearman's rank correlation coefficient (SRCC) of all models on different splits of the datasets provided in the ConferencingSpeech 2022 challenge. For each dataset, the best performance (i.e., the model with the highest SRCC) is highlighted in bold. At the time of writing, no SRCC test results have been made public.*

	Baseline		**MFCC**		**XLS-R**	
	Baseline1	Baseline2	MFCC	MFCC-Bi-LSTM	XLS-R	XLS-R-Bi-LSTM
Validation RMSE All Corpora	0.8224	0.5475	0.7579	0.6227	0.5674	**0.5000**
Validation RMSE PSTN & Tencent	0.6614	0.4965	0.6585	0.5551	0.5543	**0.4759**
Test RMSE PSTN & Tencent	0.745	0.543	/	/	/	**0.344**

Table 3: *Root mean squared error (RMSE) of all models on different splits of the datasets provided in the ConferencingSpeech 2022 challenge. For each dataset, the best performance (i.e., the model with the lowest RMSE) is highlighted in bold.*

	Entire Test Set			Only PSTN			Only TUB			Only Tencent		
Model	PCC	RMSE	RMSE-S	PCC	RMSE	RMSE-S	PCC	RMSE	RMSE-S	PCC	RMSE	RMSE-S
Baseline1	0.530	0.768	0.497	0.361	0.585	0.293	0.348	1.094	0.649	0.881	0.624	0.550
Ours	**0.800**	**0.365**	**0.320**	**0.636**	**0.351**	**0.247**	**0.796**	**0.443**	**0.418**	**0.967**	**0.300**	**0.295**

Table 4: *Preliminary test results for Baseline1 and our submitted XLS-R Bi-LSTM model. RMSE-S is the RMSE score smoothed by a third-order polynomial function. Our model outperforms Baseline1 on all metrics. The values differ slightly from those of Tables 1 and 3. This is because extra test samples were added to the TUB dataset after the preliminary results were announced.*

6. References

[1] V. Grancharov and W. Kleijn, "Speech Quality Assessment," in *Springer Handbook of Speech Processing*. Springer, 2008.

[2] W. Rao, Y. Fu, Y. Hu, X. Xu, Y. Jv, J. Han, Z. Jiang, L. Xie, Y. Wang, S. Watanabe, Z.-H. Tan, H. Bu, T. Yu, and S. Shang, "INTERSPEECH 2021 ConferencingSpeech Challenge: Towards Far-field Multi-Channel Speech Enhancement for Video Conferencing," *arxiv preprint arXiv:2104.00960*, 2021.

[3] A. R. Avila, H. Gamper, C. Reddy, R. Cutler, I. Tashev, and J. Gehrke, "Non-intrusive Speech Quality Assessment Using Neural Networks," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 631–635.

[4] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, "MBNET: MOS Prediction for Synthesized Speech with Mean-Bias Network," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 391–395.

[5] W.-C. Tseng, C.-y. Huang, W.-T. Kao, Y. Y. Lin, and H.-y. Lee, "Utilizing Self-Supervised Representations for MOS Prediction," in *Interspeech*, 2021, pp. 2781–2785.

[6] B. Cauchi, K. Siedenburg, J. F. Santos, T. H. Falk, S. Doclo, and S. Goetze, "Non-Intrusive Speech Quality Prediction Using Modulation Energies and LSTM-Network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 7, pp. 1151–1163, 2019.

[7] 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*, vol. 33, 2020, pp. 12 449–12 460.

[8] 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.

[9] K. Ethayarajh and D. Jurafsky, "Utility is in the Eye of the User: A Critique of NLP Leaderboards," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4846–4853.

[10] G. Yi, W. Xiao, Y. Xiao, B. Naderi, S. Möller, W. Wardah, G. Mittag, R. Cutler, Z. Zhang, D. S. Williamson, F. Chen, F. Yang, and S. Shang, "ConferencingSpeech 2022 Challenge: Non-intrusive Objective Speech Quality Assessment (NISQA) Challenge for Online Conferencing Applications," *arXiv preprint arXiv:2203.16032*, 2022. [Online]. Available: https://arxiv.org/abs/2203.16032

[11] B. Naderi and R. Cutler, "An Open source Implementation of ITU-T Recommendation P.808 with Validation," in *Interspeech*, 2020, pp. 2862–2866.

[12] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[13] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "NISQA: A Deep CNN-Self-Attention Model for Multidimensional Speech Quality Prediction with Crowdsourced Datasets," in *Interspeech*, 2021, pp. 2127–2131.

[14] C. Richey, M. A. Barrios, Z. Armstrong, C. Bartels, H. Franco, M. Graciarena, A. Lawson, M. K. Nandwana, A. Stauffer, J. van Hout, P. Gamble, J. Hetherly, C. Stephenson, and K. Ni, "Voices Obscured in Complex Environmental Settings (VOICES) corpus," in *Interspeech*, 2018, pp. 1566–1570.

[15] A. Stupakov, E. Hanusa, J. Bilmes, and D. Fox, "COSINE - A corpus of multi-party COnversational Speech In Noisy Environments," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2009, pp. 4153–4156.

[16] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "TorchAudio: Building Blocks for Audio and Speech Processing," *arxiv preprint arXiv:2110.15018*, 2022.

[17] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, 2020.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.