Aggregation Strategies of Wav2vec 2.0 Embeddings for Computational Paralinguistic Tasks

Mercedes Vetráb[1](✉) and Gábor Gosztolya[1,2]

[1] Institute of Informatics, University of Szeged, Szeged, Hungary
{vetrabm,ggabor}@inf.u-szeged.hu
[2] ELKH-SZTE Research Group on Artificial Intelligence, Szeged, Hungary

Abstract. Throughout the history of computational paralinguistics, numerous feature extraction, preprocessing and classification techniques have been used. One of the important challenges in this subfield of speech technology is handling utterances with different duration. Since standard speech processing features (such as filter banks or DNN embeddings) are typically frame-level ones and we would like to classify whole utterances, a set of frame-level features have to be converted into fixed-sized utterance-level features. The choice of this aggregation method is often overlooked, and simple functions like mean and/or standard deviation are used without solid experimental support. In this study we take `wav2vec 2.0` deep embeddings, and aggregate them with 11 different functions. We sought to obtain a subset of potentially optimal aggregation functions, because there are no general rules yet that can be applied universally between subtopics. Besides testing both standard and non-traditional aggregation strategies individually, we also combined them to improve the classification performance. By using multiple aggregation functions, we were able to achieve significant improvements on three public paralinguistic corpora.

Keywords: Paralinguistics · Wav2vec 2.0 · Embeddings · Aggregation

1 Introduction

In the past, the primary focus of automatic speech processing research was generating a transcription for an audio recording (i.e. Automatic Speech Recognition) [14]. From the 1990s to the present, several other topics have received more attention related to phenomena present in human speech, such as speaker recognition and diarisation ("who's speaking when") [13], detecting Parkinson's [15,16,38] or Alzheimer's [3,23,24] disease, assessing the level of depression [6], age and gender recognition [25], emotion recognition [21,41], and estimating the degree of sleepiness [5] or conflict intensity [11]. These subtopics are part of computational paralinguistics, which has recently started to receive more interest.

A. Karpov et al. (Eds.): SPECOM 2023, LNAI 14338, pp. 79–93, 2023.
https://doi.org/10.1007/978-3-031-48309-7_7

In this field, instead of generating transcriptions, we seek to identify non-verbal aspects of human communication such as tone of voice and other vocal cues. Here, we need to associate different lengths of audio recording inputs (i.e. utterances) with a single label output. The final aim is always a classification or a regression at the utterance level. For example, if we have two-minute-long recording, first we have to extract features from it, then classify whether the speaker is angry or not. This means that we have to calculate a fixed-dimensional, classifiable feature vector out of a varying-length recording. A typical strategy for this is to split the input into smaller chunks (i.e. frames) and calculate low-level descriptors (e.g. MFCCs) to get frame-level features. Then we feed them into a neural network to extract frame-level embeddings. Finally we aggregate them into an utterance-level feature and use it to classify the utterance.

Another key technical property of computational paralinguistics is that we typically have small-sized corpora. This usually does not make it suitable for using DNNs as classifiers, and deep learning methods are still in their early stages of development [21,33]. Traditional classification methodologies tend to perform better than end-2-end DNNs [10,29,34]. Nowadays scientists employ DNNs more and more frequently, but mostly for frame-level feature (i.e. embedding) extraction [39]. Deep neural network embeddings can reduce the feature space dimension while preserving important information. It has been effective in capturing complex relationships in the data and outperforming traditional feature extraction methods. The small size of paralinguistical datasets makes it difficult to train a feature extractor DNN from scratch, so usually a standard ASR corpus is used for pretraining. Standard examples are HMM/DNN acoustic models [9], x-vectors [31], ECAPA-TDNN [30] and wav2vec 2.0 [19].

In this study, we focus on the utterance-level aggregation step. Although researchers tend to use task specific aggregations including only the most popular metrics such as mean and standard deviation, our aim is to show that there are other efficient techniques available too. Some of them can handle different paralinguistic subtopics at the same time. With state-of-the-art self-supervised `wav2vec 2.0` DNN embeddings, we investigated 11 aggregation strategies including both traditional and less frequently employed ones. We conducted experiments on three different databases to find general trends across various paralinguistic subtopics. We found that certain non-traditional metrics can be highly effective for almost any subtopic, and traditional metrics vary in performance depending on the dataset. We were interested in the classification performance that could be obtained by combining different aggregation functions. By using sequential forward feature selection, we achieved relative error rate improvements of $4 - 10\%$ on the test scores in two datasets. We achieved a slight improvement on the third corpus. Our results, probably indicate that the effective summarisation of frame-level embeddings is a nontrivial task, and classification performance can be improved significantly using multiple aggregation functions, regardless of the actual paralinguistic subtopic. In addition, we present a novel approach rule set for aggregation selection where we identify general patterns using our results and provide guidelines for selecting appropriate aggregation methods for `wav2vec 2.0` embeddings.

2 Proposed Methods

2.1 Wav2vec 2.0 Embeddings

To extract frame-level embeddings, we employed a self-supervised and fine-tuned `wav2vec 2.0` model [12]. The model has two main parts: (1) a Convolutional Neural Network (CNN) block, (2) a BERT-based transformer block. The first part encodes features by transforming the raw input waveform into a sequence of high-level feature representations (i.e. latent speech representation). The CNN has "dilation" between the filter weights, which allows the filter to capture information from a wider range of time steps in the input sequence, without increasing the number of parameters. The second part transforms the CNN output into a sequence of high-level feature vectors, which capture the relationships between the input waveform and the extracted features. It has a contextualised transformer architecture based on the widely used BERT model. The transformer consists of a multi-head self-attention mechanism and a position-wise feed-forward network [4]. The structure of a fine-tuned wav2vec 2.0 model can be seen in Fig. 1.

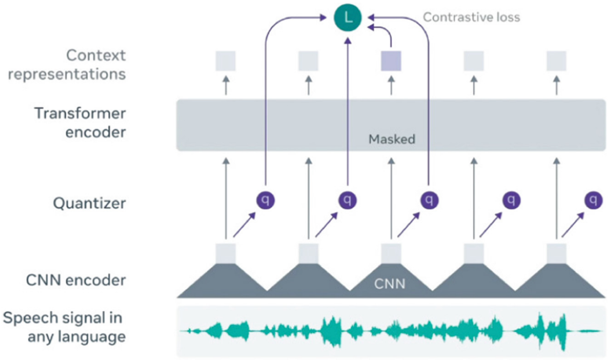

Fig. 1. The fine-tuned wav2vec 2.0 framework structure [1].

The model can be trained with the cross-lingual representation (XLSR) learning approach, which involves two steps: (1) pretraining the model by self-supervised learning on large unlabeled datasets of speech in different languages, (2) fine-tuning this model on a smaller labeled corpus with the target speech language (e.g. German). In this way, the model learns to share discrete tokens across languages. The pretraining step divides the input into small segments while applying random masking. Then it utilises a self-supervised learning (Contrastive Predictive Coding (CPC) approach), where the model encodes the segments into a set of discrete latent variables using two pre-defined codebooks and predicts a future latent variable from the discrete ones. In the fine-tuning step the original output layer is replaced with task-specific layers (typically a recurrent neural network (RNN) and a Softmax layer). Then, the modified network is optimised via Connectionist Temporal Classification (CTC) loss [4].

After the training and the fine-tuning, we can use the network as an embedding extractor by freezing the weights and removing the last few layers. We experiment with two setups, where we extract embeddings from: (1) the last layer of the CNN block, (2) the last layer of the Transformer block. When we feed the paralinguistic utterances into the model, the output of the last remaining layer serves as the embeddings. These feature vectors may contain relevant information about the speaker and other aspects of the speech signal.

2.2 Embedding Aggregation

Since databases contain recordings with different lengths, we have a different number of embedded features for each recording. This means one embedding for each frame window. The aim is to predict one label to one utterance, but we can not simply concatenate these embeddings because traditional classifiers handle only fixed-sized input vector. In order to address this issue, an aggregation step must be employed to transform frame-level embeddings into utterance-level features. The general process of an aggregation is shown in Fig. 2. We had an input recording consisting of y number of frame windows, each containing f number of low-level descriptors. To get an utterance-level feature vector from them, we combine all the y number of vectors by computing a statistical measure such as the mean, variance, or other value for each f features in the time axis. With aggregation we can obtain an utterance-level feature vector that has the same size as an original embedding vector. Since the size of this aggregated vector is independent of the number of the windows (and therefore, of the duration of the utterance), it can be used with any traditional classification method.

3 Databases

We performed our experiments on three public paralinguistic corpora, that covered a variety of topic. However, all three corpora had native German speakers. This allowed us to justifiably employ the same `wav2vec 2.0` model for frame-level embeddings extraction, as it was fine-tuned for German speech. All of the databases was used on one of the INTERSPEECH Computational Paralinguistic Challenges [26–28].

3.1 The iHEARu-EAT Database

The Munich University of Technology provided the iHEARu-EAT corpus [18], which includes approximately 2.9 h of close-to-native German speech from 30

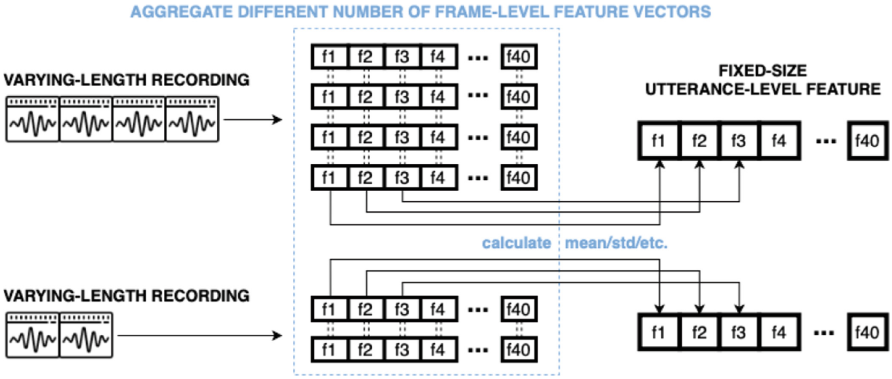

Fig. 2. Creating fixed-sized feature vectors from varying-length utterances.

subjects (15 females, 15 males). It was recorded in a quiet, slightly echoing office room, and the recordings has a sampling rate of 16 kHz. The classification task was to determine the type of food being eaten while speaking: apple, nectarine, banana, crisp, biscuit, gummy bear, and no food. The speakers completed various tasks, such as reading the German version of "The North Wind and the Sun" or providing a spontaneous narrative about their favourite activity. The database was divided into a training set (14 speakers), a development set (6 speakers) and a test set (10 speakers) in a speaker-independent manner.

3.2 The URTIC Database

The Institute of Safety Technology at the University of Wuppertal in Germany provided the Upper Respiratory Tract Infection Corpus (e.g.: URTIC) [18], which contains native German speech from 630 participants (248 females, 382 males). The corpus has a total duration of approximately 45 h. The classification task was to determine whether the speaker had a cold or not. The recordings were downsampled from a sampling rate of 44.1 kHz to 16 kHz. The task assigned to the participants included reading short stories, producing voice commands and speaking spontaneously about a personal experience. The corpus was divided into three sets (train, dev, test), each containing 210 speakers. The training and development sets contained 37 infected and 173 uninfected participants.

3.3 The AIBO Database

The FAU AIBO Emotion Corpus [32] contains recordings of 51 native German children speech, who were playing with a pet robot called AIBO. The recordings were taken from two schools: 9959 recordings from the Ohm school and 8257 from the Mont school, with a total duration of around 9 h. The Ohm subset was divided into a training set (7578 utterances, 20 children) and a development set (2381 utterances, 6 children), while the Mont subset served as the test set (8257 utterances). Classes were merged from the original 11 emotional classes into 5, as Anger (including angry, irritated and reprimanding), Emphatic, Neutral, Positive (motherly and joyful), and the Rest (helpless, surprised, bored, non-neutral other).

4 Experimental Setup

4.1 Wav2vec 2.0 Embeddings

The first step of our method is to extract frame-level embeddings from raw audio data. We used a pretrained and fine-tuned wav2vec 2.0 model and extracted features from two different layers: (1) the last layer of the CNN block and (2) the last layer of the modified Transformer block. These vectors may contain relevant information about the speakers and other aspects of the speech signal. The size of the embeddings was 512 for convolutional and 1 024 for hidden layers.

4.2 Embedding Aggregation

During aggregation, we used 11 different statistical methods to convert frame-level embeddings into an utterance-level feature vector. Besides the traditional approaches of *mean*, *median* and *standard deviation*, we experimented with the *skewness*, the *kurtosis*, the*minimum*, the *maximum* and the 1st, 25th, 75th, 99th *percentiles* (i.e. the value below which a given percentage k of scores falls). Note that median is identical to the 50th percentile. The 1st and 99th percentiles are frequently used as alternatives to minimum and maximum, because they are not that sensitive to outliers [22].

4.3 Classification and Evaluation

We used traditional Support Vector Machines (i.e. SVMs) for classification and utilized the Python port of the LibSVM implementation [2]. Following our previous experiments [7,35,37], we employed the ν-SVM method with a linear kernel. The complexity (C) was determined by testing powers of 10 between 10^{-5} and 10^{0}. To avoiding peeking and determine the optimal hyperparameter settings, we trained our models on the train set and evaluated them on the development set. In the end, we measured final performance of the best parameter, by training the model on the concatenation of train and dev sets and evaluating it on the test set. To measure the efficiency of an SVM model, we used the Unweighted Average Recall (i.e. UAR) metric [20], corresponding to taking the mean of the class-wise recall scores. This is a widely used metric for these corpora. [27,28,32].

In the case of the AIBO and the URTIC corpora, we always standardized utterance-level features (i.e. converted them so as to have a zero mean and unit variance). Due to the unbalanced class distribution and the relatively large size of these corpora, we also employed downsampling on them (i.e. we discarded training examples from the more frequent classes), as these techniques proved

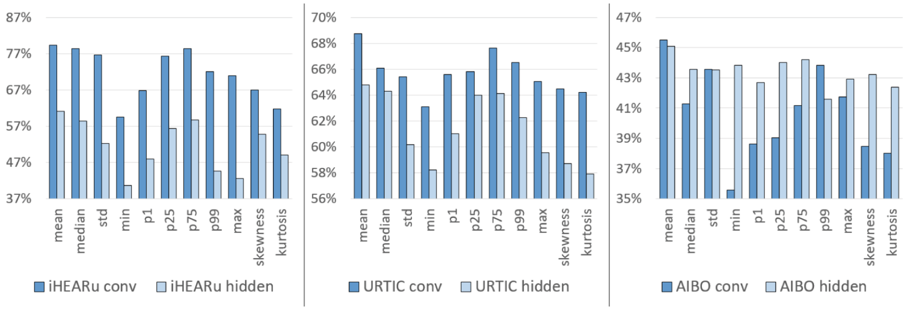

Fig. 3. Development results got from convolutional and transformer (i.e. hidden) layer embeddings while using different aggregation techniques. The x axis represents the aggregation method and the y axis represents the UAR value.

to be beneficial in our previous experiments [8,36]. In the case of iHEARu-EAT, we performed speaker-wise standardization, where the test set speaker IDs were determined by using the single Gaussian-based bottom-up Hierarchical Agglomerative Clustering algorithm [17,40].

5 Experimental Results

First, we compared the performance of the convolutional and the transformer (i.e. hidden) layer embeddings on the development set. Our best results for all 11 aggregation functions are shown in Fig. 3. From our results, convolutional embeddings significantly outperformed the hidden representations on the iHEARu-EAT corpus (79.4%–83.7% and 61.1%–62.9%, convolutional and hidden embeddings, respectively). On the other two corpora, it also had a slight advantage against hidden embeddings (URTIC: 63.3%–68.7% and 64.8%–66.1%, AIBO: 42.7%–45.5% and 43.6%–45.1%, convolutional and hidden embeddings, respectively). Although the hidden layer performed better in percentage terms on the AIBO database, but it varied greatly, proved unreliable and lost robustness. Upon closer inspection, the minimum and maximum aggregations differ from the previous pattern. The reason for the better performance scores with the hidden layer in the case of AIBO might be that emotion recognition requires a higher level analysis than the other two paralinguistic subtask, so the last hidden layer has a better comprehensive overview. A deeper layer, like a convolutional, can analyse smaller details, which is more advantageous in the case of sounds produced by cold or during eating. The other significant difference of the AIBO database compared with the others is that it contains recordings of children's speech. Changes in tones and speech skills can produce slight differences in the analysis results. Due to these observations, we decided to continue our research with convolutional embeddings. Our decision and recommendation is to use the convolutional layer because it behaves more robustly and has the same pattern in all three databases.

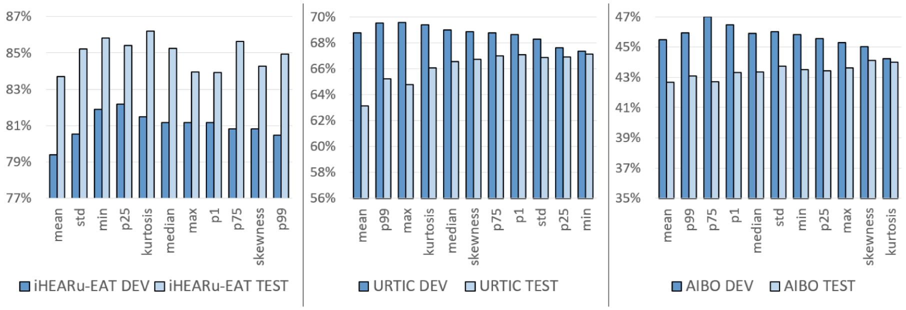

Fig. 4. Development and test results for embeddings got from the convolutional layer, when combining different aggregation strategies with sequential forward selection (SFS). The order of the selected aggregation methods is shown on the x axis.

If we take a closer look at the aggregations, we observe same general trends. The mean aggregation produced the best results on each database, which, as it is perhaps the most frequently used method, is not that surprising. Standard deviation appear to be a promising alternative for a potential combination. Regarding percentiles, the central ones (i.e. 25%, 75%) have competitive performances, so we should pay more attention to these non-traditional aggregations. We would like to recommend their usage more, especially the 75th percentile. The traditional median metric (which is the same as the 50% percentile), had a varying performance depending on the database, while it follows the accuracy curve of the percentiles. This curve shows that if you take all the frame-level vectors of a recording and sort the values for each feature in ascending order, which part of the ordered sequence is the best descriptor of that recording. Last, but not least, for all corpora we obtained very low results with the minimum and maximum aggregations (where minimum is practically the 0th, while the maximum is the 100th percentile). It tells us that wav2vec 2.0 embeddings frequently contain outlier values, which has a significant drawback in classification. Instead of these, the 1st and 99th percentiles are promising alternatives. Although low percentiles may also be minor outliers, but the trend clearly shows that their use is more advisable than the minimum and maximum. Lastly, we tested skewness and kurtosis aggregations, but they gave a significantly lower performance overall.

5.1 Feature Combinations

In the second series of experiments, we wanted to further improve the classification performance, so we used sequential forward selection (SFS) to combine multiple aggregated feature vectors. The basic idea behind SFS is to initialize a subset with only the best method, and then iteratively add one more aggregation to the subset, based on which combination provides the greatest improvement in performance. To combine a subset of aggregations, we took the mean of the corresponding posterior estimates and we measured the efficiency of the averaged posterior by calculating the UAR score. SFS helps to reduce the risk of overfitting, as the selected subset is more likely to be the most relevant and informative for the given prediction task. Our results are shown in Fig. 4.

With the iHEARu-EAT database we were able to improve the development test scores up to the 4th iteration. When adding further aggregation approaches, the development UAR scores naturally decrease. However test set scores behave quite differently as they fluctuate and remain in the 84–86% range.

Table 1. The best development and test results for different aggregation strategies and their combinations for the iHEARu-EAT paralinguistic corpora.

iHEARu-EAT		
Aggregation	Dev	Test
Mean	79.4%	83.7%
Median	78.4%	82.6%
75th percentile	78.4%	81.2%
mean+std+min+p25	**82.2%**	**85.4%**
All	80.5%	85.0%

Table 1 contains an overall statistic about the iHEARu-EAT database. The first three rows show the three best aggregations from the previous experiment of the convolutional layer, which are the mean, the median and the 75th percentile. The penultimate row shows the best result obtained with the combination approaches. This subset of aggregations determined by the development set, contains the mean, standard deviation, minimum and the 25th percentile. Here, we report an 8% relative error rate improvement. The last row shows the UAR scores we obtained when we combined all of the aggregation methods. It has a score close to the best combination, but we noticed that if we include too much unnecessary information, we can lose its ability to generalise.

With the URTIC database, we found that we can improve the development results up to the 3rd iteration. After that, for the development set the UAR scores also naturally decrease, but the test set evaluation scores have further improvements of between 66% and 67%.

Table 2 contains an overall statistic about the database. It has the same pattern as the previous one. The three best simple aggregations from the convolutional layer, were the mean, 75th percentile and 99th percentile, where the percentiles improved the generalisation ability as the higher test results indicate. The best subset of aggregations contains the mean, 99th percentile and the maximum and we report a 1.16% dev, 2.69% test relative error rate improvement against the mean only and 2.81% dev, −2.41% test scores against the 75th percentile. When we combined all of the aggregation methods we got an increase in the test values. In our opinion these results indicate that there is a significant

Table 2. The best development and test scores for different aggregation strategies and their combinations for the URTIC paralinguistic corpora.

URTIC		
Aggregation	Dev	Test
Mean	68.7%	63.1%
75th percentile	67.6%	66.4%
99th percentile	66.5%	66.2%
mean+p99+max	**69.5%**	64.8%
All	67.3%	**67.1%**

difference between the feature distribution of the development and the test sets, because different aggregation types seemed to be important in case of these sets.

Table 3. The best development and test results for different aggregation strategies and combinations for the AIBO paralinguistic corpora.

AIBO		
Aggregation	Dev	Test
Mean	45.5%	42.7%
99th percentile	43.8%	42.9%
Standard dev.	43.5%	43.3%
mean+p99+p75	**47.0%**	42.7%
All	44.2%	**44.0%**

With the AIBO database we found that we could improve development scores up to the 3rd iteration. After that, the development set UAR scores also naturally decrease, but the test set evaluation scores have further improvements between 43% and 44%. Table 3 contains an overall statistic about the database with the same pattern as the previous one. The three best aggregations from the convolutional layer, were the mean, 99th percentile and standard deviation. Here, the second and the third best aggregation also gave slight improvements on the test values. The best subset of aggregations contains the mean, 99th percentile and the 75th percentile. We have relative error rate improvements between 1.41% and 8.05%. When we combined all of the aggregation methods, we observed the same behaviour as that for the URTIC database.

In the view of the three databases, lower than middle percentile values works better for iHEARu-EAT while higher values performs better for URITC and AIBO corpora. Clearly, there is another global tendency about needing 3 or 4 iterations of the SFS to improve the efficiency of our model. As we can see, combinations bring improvements on the test set as well, which means it increases the generalisation ability of the model. These significant improvements represented in relative error reduction values of 8.0–10.4% (iHEARu-EAT), 4.6–10.8% (URTIC) and 0–2.3% (AIBO), and they were obtained using simple, easy-to-implement and quick-to-calculate aggregation techniques. New aggregations can be easily calculated alongside traditional metrics because it can be done in parallel in the stage where all the frame-level features are available for one recording. Each new metric introduces as many new features as we originally had. This leads to an utterance-level feature vector of length 1536–2048, which always contains one or two non-traditional percentile values. This does not drastically increase the dimensionality for a casual set of features extracted from the bottleneck layer (which commonly used in paralinguistics).

6 Conclusion and Future Work

In this study we focused on applying aggregation strategies for deep neural network embeddings in the field of computational paralinguistics.

We performed our experiments on three public paralinguistic corpora, that have a variety of topics, but were uniform in their spoken language. We used a `wav2vec 2.0` DNN to extract embeddings and then we investigated 11 more- and less-traditional aggregation strategies for combining frame-level embeddings into utterance-level features. In the second set of experiments, we used sequential forward selection to improve our results and find the overall best aggregation methods and global tendencies across databases.

We found a well-defined general pattern between aggregations. The traditional standard deviation and median aggregations are heavily topic dependent. The mean aggregation is always a good choice, but it is not the only one. Our first results indicate that middle percentile aggregations are competitive techniques. This is true for both the convolutional and hidden layers. Overall, it seems that wav2vec 2.0 embeddings can be expected to contain extreme values, which are not really useful for classification. Owing to this, aggregation methods that are sensitive to outliers might be expected to perform less robust than those that can handle the outlying values better. For the former, the obvious examples are the minimum and maximum, which were clearly outperformed by the first and 99th percentiles. We also found that choosing only one aggregation technique leads to a suboptimal classification performance. In the second phase where we performed SFS initialized with the mean, there was a trend across databases, as the peak of improvement fell on the combination of the first 3–4 techniques. The best combinations typically include the mean, a non-traditional percentile value below and/or above the median. Using multiple aggregations simultaneously, we were able to make improvements on both the development and test sets. Based on all of these, we see a trend in our model. It will have a better generalisation ability if we apply at least the above-mentioned 3 types of aggregations together. This way, we can improve the generalization ability of the model, while keeping the feature space below 2048. The computational demand does not increase drastically due to possible parallelization. Our results suggests that aggregating embedding vectors by just using one function leads to a significant information loss. Quite surprisingly, combining *all* aggregated feature sets led to significant improvements on the test set, which could indicate that there is a big stochastic difference between the development and test data.

This, in our opinion, indicates that aggregating the frame-level embeddings is a task which is far from trivial, and that significant improvements can be obtained in the classification performance using other techniques instead of the traditional mean and/or standard deviations. In the case of SFS, working with posteriors is a more time and memory consuming choice, but because of the possible differences, in the near future we plan to retrain our models with the concatenated feature sets. Another possible future direction is to gain more insights using other aggregation methods and datasets, and systematically explore them. Additional research opportunity is to give more emphasis on the important vectors with weighted aggregation, where weights can be learned.

Acknowledgements. This research was supported by the NRDI Office of the Hungarian Ministry of Innovation and Technology (grant no. TKP2021-NVA-09), and within the framework of the Artificial Intelligence National Laboratory Program (RRF-2.3.1-21-2022-00004).

References

1. Baevski, A., Auli, M., Conneau, A.: Wav2vec 2.0: learning the structure of speech from raw audio (2020). https://ai.meta.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 1–27 (2011). https://doi.org/10.1145/1961189.1961199
3. Chen, J., Ye, J., Tang, F., Zhou, J.: Automatic detection of Alzheimer's Disease using spontaneous speech only. In: Proceedings of the Interspeech 2021, pp. 3830–3834 (2021). https://doi.org/10.21437/Interspeech.2021-2002
4. Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M.: Unsupervised Cross-lingual Representation Learning for Speech Recognition (2020). https://doi.org/10.48550/ARXIV.2006.13979
5. Egas-López, J.V., Gosztolya, G.: Deep Neural Network Embeddings for the estimation of the degree of sleepiness. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 7288–7292 (2021). https://doi.org/10.1109/ICASSP39728.2021.9413589
6. Egas-López, J.V., Kiss, G., Sztahó, D., Gosztolya, G.: Automatic assessment of the degree of clinical depression from speech using X-Vectors. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8502–8506 (2022). https://doi.org/10.1109/ICASSP43922.2022.9746068
7. Egas-López, J.V., Vetráb, M., Tóth, L., Gosztolya, G.: identifying conflict escalation and primates by using ensemble x-vectors and fisher vector features. In: Proceedings of the Interspeech 2021, pp. 476–480 (2021). https://doi.org/10.21437/Interspeech.2021-1173

8. Gosztolya, G.: Using the Fisher vector representation for audio-based emotion recognition. Acta Polytechnica Hungarica **17**, 7–23 (2020)
9. Gosztolya, G., Tóth, L., Svindt, V., Bóna, J., Hoffmann, I.: Using acoustic deep neural network embeddings to detect multiple sclerosis from speech. In: Proceedings of ICASSP, pp. 6927–6931 (2022)
10. Gosztolya, G., Beke, A., Neuberger, T.: Differentiating laughter types via HMM/DNN and probabilistic sampling. In: Speech and Computer, SPECOM 2019. vol. 11658, pp. 122–132 (2019)
11. Grezes, F., Richards, J., Rosenberg, A.: Let me finish: automatic conflict detection using speaker overlap. In: Proceedings of the Interspeech 2013, pp. 200–204 (2013). https://doi.org/10.21437/Interspeech.2013-67
12. Grosman, J.: Fine-tuned XLSR-53 large model for speech recognition in German (2021). https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-german
13. Han, K.J., Kim, S., Narayanan, S.S.: Strategies to improve the robustness of Agglomerative Hierarchical Clustering under data source variation for speaker diarization. IEEE Trans. Audio Speech Lang. Process. **16**, 1590–1601 (2008). https://doi.org/10.1109/TASL.2008.2002085
14. Hinton, G., et al.: Deep Neural Networks for Acoustic Modeling in Speech Recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**, 82–97 (2012). https://doi.org/10.1109/MSP.2012.2205597
15. Jeancolas, L., et al.: X-Vectors: new quantitative biomarkers for early Parkinson's Disease detection from speech. Front. Neuroinform. **15**, 1–18 (2021). https://doi.org/10.3389/fninf.2021.578369
16. Kadiri, S., Kethireddy, R., Alku, P.: Parkinson's Disease detection from speech using Single Frequency Filtering Cepstral Coefficients. In: Proceedings of the Interspeech 2020, pp. 4971–4975 (2020). https://doi.org/10.21437/Interspeech.2020-3197
17. Kaya, H., Karpov, A., Salah, A.: Fisher vectors with cascaded normalization for paralinguistic analysis. In: Proceedings of the Interspeech 2015, pp. 909–913 (2015). https://doi.org/10.21437/Interspeech.2015-193
18. Krajewski, J., Schieder, S., Batliner, A.: Description of the upper respiratory tract infection corpus (urtic). In: Proceedings of the Interspeech 2017 (2017)
19. Lin, W.W., Mak, M.W.: Wav2spk: a simple DNN architecture for learning speaker embeddings from waveforms. In: Proceedings of Interspeech, pp. 3211–3215 (2020)
20. Metze, F., Batliner, A., Eyben, F., Polzehl, T., Schuller, B., Steidl, S.: Emotion recognition using imperfect speech recognition. In: Proceedings of the Interspeech 2010, pp. 478–481 (2010). https://doi.org/10.21437/Interspeech.2010-202
21. Mustaqeem, Kwon, S.: CLSTM: deep feature-based speech emotion recognition using the hierarchical ConvLSTM network. Mathematics **8**, 1–19 (2020). https://doi.org/10.3390/math8122133
22. Oflazoglu, C., Yildirim, S.: Recognizing emotion from Turkish speech using acoustic features. In: EURASIP Journal on Audio Speech and Music Processing 2013 (2013). https://doi.org/10.1186/1687-4722-2013-26
23. Pappagari, R., et al.: Automatic detection and assessment of Alzheimer Disease using speech and language technologies in low-resource scenarios. In: Proceedings of the Interspeech 2021, pp. 3825–3829 (2021). https://doi.org/10.21437/Interspeech.2021-1850
24. Pérez-Toro, P., et al.: Alzheimer's detection from English to Spanish using acoustic and linguistic embeddings. In: Proceedings of Interspeech 2022, pp. 2483–2487 (2022). https://doi.org/10.21437/Interspeech.2022-10883

25. Přibil, J., Přibilová, A., Matoušek, J.: GMM-based speaker age and gender classification in Czech and Slovak. J. Electr. Eng. **68**, 3–12 (2017). https://doi.org/10.1515/jee-2017-0001
26. Schuller, B., Steidl, S., Batliner, A.: The INTERSPEECH 2009 emotion challenge. In: Proceedings of the Interspeech 2009, pp. 312–315 (2009). https://doi.org/10.21437/Interspeech. 2009–103
27. Schuller, B., et al.: The INTERSPEECH 2017 computational paralinguistics challenge: addressee, cold & snoring. In: Proceedings of the Interspeech 2017, pp. 3442–3446 (2017). https://doi.org/10.21437/Interspeech.2017-43
28. Schuller, B., et al.: The INTERSPEECH 2015 computational paralinguistics challenge: Nativeness, Parkinson's & eating condition. In: Proceedings of the Interspeech 2015, pp. 478–482 (2015). https://doi.org/10.21437/Interspeech.2015-179
29. Schuller, B.W., et al.: The INTERSPEECH 2019 computational paralinguistics challenge: Styrian dialects, continuous sleepiness, baby sounds & orca activity. In: Proceedings of the Interspeech 2019, pp. 2378–2382 (2019). https://doi.org/10.21437/Interspeech.2019-1122
30. Sheikh, S.A., Sahidullah, M., Hirsch, F., Ouni, S.: Introducing ECAPA-TDNN and Wav2Vec2.0 Embeddings to Stuttering Detection (2022). https://doi.org/10.48550/ARXIV.2204.01564
31. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-Vectors: robust DNN embeddings for speaker verification. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 5329–5333 (2018). https://doi.org/10.1109/ICASSP.2018.8461375
32. Steidl, S.: Automatic classification of emotion related user states in spontaneous children's speech. Logos-Verlag Berlin, Germany (2009). https://d-nb.info/992551641
33. Tzirakis, P., Zhang, J., Schuller, B.W.: End-to-end speech emotion recognition using deep neural networks. In: 2018 IEEE international Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5089–5093 (2018)
34. Van Segbroeck, M., et al.: Classification of cognitive load from speech using an i-vector framework. In: Proceedings of the Interspeech 2014, pp. 751–755 (2014). https://doi.org/10.21437/Interspeech.2014-114
35. Vetráb, M., Gosztolya, G.: Speech emotion detection form a Hungarian database with the Bag-of-Audi-Words technique (in Hungarian). In: Proceedings of MSZNY, pp. 265–274. Szeged (2019)
36. Vetráb, M., Gosztolya, G.: Using hybrid HMM/DNN embedding extractor models in computational paralinguistic tasks. Sensors **23**, 5208 (2023)
37. Vetráb, M., et al.: Using spectral sequence-to-sequence autoencoders to assess mild cognitive impairment. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 6467–6471 (2022). https://doi.org/10.1109/ICASSP43922.2022.9746148
38. Vásquez-Correa, J., Orozco-Arroyave, J.R., Nöth, E.: Convolutional Neural Network to model articulation impairments in patients with Parkinson's Disease. In: Proceedings of the Interspeech 2017, pp. 314–318 (2017). https://doi.org/10.21437/Interspeech.2017-1078
39. Wagner, J., Schiller, D., Seiderer, A., Andre, E.: Deep learning in paralinguistic recognition tasks: are hand-crafted features still relevant? In: Interspeech, pp. 147–151 (2018). https://doi.org/10.21437/Interspeech.2018-1238

40. Wang, W., Lu, P., Yan, Y.: An improved hierarchical speaker clustering. Acta Acustica **33**, 9–14 (2008)
41. Zhao, Z., Bao, Z., Zhang, Z., Cummins, N., Wang, H., Schuller, B.: Attention-enhanced connectionist temporal classification for discrete speech emotion recognition. In: Proceedings of the Interspeech 2019, pp. 206–210 (2019). https://doi.org/10.21437/Interspeech.2019-1649