
Björn W. Schuller¹,², Anton Batliner²,³, Christian Bergler³, Cecilia Mascolo⁴, Jing Han⁴, Julia Lefter⁵, Heysem Kaya⁶, Shahin Amiriparian², Alice Baird³, Lukas Stappen², Sandra Ott³, Maurice Gerczuk³, Panagiotis Tzirakis¹, Chloë Brown⁴, Jagmohan Chauhan¹, Andreas Grammenos⁵, Apinan Hashtanasombat⁴, Dimitris Spathis³, Tong Xia¹, Pietro Cicuta¹, Leon J. M. Rothkrantz⁵, Joeri A. Zwerts⁶, Jelle Treep⁶, Casper Kaandorp⁶

¹GLAM – Group on Language, Audio & Music, Imperial College London, UK
²EIHW – Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany
³Pattern Recognition Lab, FAU Erlangen-Nuremberg, Germany
⁴University of Cambridge, UK
⁵Delft University of Technology, The Netherlands
⁶Faculty of Science, Utrecht University, The Netherlands

schuller@iis.ee

Abstract

The INTERSPEECH 2021 Computational Paralinguistics Challenge addresses four different problems for the first time in a research competition under well-defined conditions: In the COVID-19 Cough and COVID-19 Speech Sub-Challenges, a binary classification on COVID-19 infection has to be made based on coughing sounds and speech; in the Escalation Sub-Challenge, a three-way assessment of the level of escalation in a dialogue is featured; and in the Primates Sub-Challenge, four species vs background need to be classified. We describe the Sub-Challenges, baseline feature extraction, and classifiers based on the ‘usual’ COMPARE and BoAW features as well as deep unsupervised representation learning using the AuDEEP toolkit, and deep feature extraction from pre-trained CNNs using the DEEP SPECTRUM toolkit; in addition, we add deep end-to-end sequential modelling, and partially linguistic analysis.

Index Terms: Computational Paralinguistics, Challenge, COVID-19, Escalation, Primates

1. Introduction

In this INTERSPEECH 2021 COMPUTATIONAL PARALINGUISTICS CHALLENGE (COMPARE) – the thirteenth since 2009 [1], we address four new problems within the field of Computational Paralinguistics [2] in a challenge setting:

In the COVID-19 Cough Sub-Challenge (CSS) and COVID-19 Speech Sub-Challenge (CSS), coughing sounds or speech are used to binary classify COVID-19 (or not) infection. In the present pandemic situation, great potential lies in low-cost, anywhere and anytime accessible real-time pre-diagnosis of COVID-19 infection. To date, the possibility has been shown [3], yet a controlled challenge test-bed is lacking. In the Escalation Sub-Challenge (ESS), participants are faced with three-way classification of the level of escalation in human dialogues. A range of applications exists including human-to-computer interaction, computer mediated human-to-human conversation, or public security. Finally, in the Primates Sub-Challenge (PRS), we classify four species of primates versus background noise. Real-life applications include wild-life monitoring in habitats, e.g., to save species from extinction.

For all tasks, a target class has to be predicted for each case. Contributors can employ their own features and machine learning algorithms; standard feature sets and procedures are provided. Participants have to use the pre-defined partitions for each Sub-Challenge. They may report results obtained from the Train(ing)/Dev(elopment) set – preferably with the supplied evaluation setups, but have only five trials to upload their results on the Test set per Sub-Challenge, whose labels are unknown to them. Each participation must be accompanied by a paper presenting the results, which undergoes peer-review and has to be accepted for the conference in order to participate. The organisers preserve the right to re-evaluate the findings, but will not participate in the Challenge. As evaluation measure, we employ in all Sub-Challenges Unweighted Average Recall (UAR) as used since the first Challenge from 2009 [1], especially because it is more adequate for (unbalanced) multi-class classifications than Weighted Average Recall (i.e., accuracy) [2, 4]. Ethical approval for the studies has been obtained from the pertinent committees. In section 2, we describe the challenge corpora. Section 3 details baseline experiments, metrics, and baseline results; concluding remarks are given in section 4.

2. The Four Sub-Challenges

2.1. The COVID-19 Cough Sub-Challenge (CSS) and the COVID-19 Speech Sub-Challenge (CSS)

For the CSS and CSS, we employ two subsets from the Cambridge COVID-19 Sound database [5, 6]. The database was collected via the COVID-19 Sounds App since its launch in April 2020, aiming at collecting data to inform the diagnosis of COVID-19 based primarily on voice, breathing, and coughing. Participants were able to provide audio samples together with their COVID-19 test results via multiple platforms (a webpage, an Android app, and an iOS app). The participants also provided basic demographic, medical information, and reported symptoms. For the CSS and CSS, only cough sounds and voice recordings with COVID-19 positive/ negative test results were included separately, and only audio data and the corresponding COVID-19 test labels are provided. The quality of these data was manually checked. As they were crowd-sourced, the original audio data had varying sampling rates and formats; all of them were resampled (in a few cases, upsampled) and converted to 16 kHz and mono/16 bit, and further normalised recording-wise
to eliminate varying loudness. For the CCS, 725 recordings from 343 participants were provided, in total 1.63 hrs. In each cough recording, the participant provided one to three forced coughs. For the CSS, we use 893 recordings from 366 participants, in total 3.24 hrs. In each speech recording, the participant recorded speech content (“I hope my data can help to manage the virus pandemic.”) in one language (English, Italian, or German, etc.), one to three times. For each recording, a COVID-19 test result was available which was self-reported by the participant. To create the two-class classification task, the original COVID-19 test results were mapped onto either positive (denoted as ‘P’) or negative (‘N’). Note that Train, Dev, and Test sets contain mutually different speakers; within each set, however, speakers can occur more than once; thus, it is essential to stick to the partitioning provided.

2.2. The Escalation Sub-Challenge (ESS)

For the ESS, the INTERSPEECH COMPARE Escalation Corpus is provided, consisting of the Dataset of Aggression in Trains (TR) [7] and the Stress at Service Desk Dataset (SD) [8]. Both present unscripted interactions between actors, where friction appears as they spontaneously react to each other based on short scenario descriptions. While the datasets share the same procedure for eliciting interactions, the topics, the number of participants in the scene, and amount of overlapping speech, as well as the recording quality differ. The TR dataset consists of 21 scenarios of unwanted behaviours in trains and train stations (e.g., harassment, theft, travelling without a ticket) played by 13 subjects. It was annotated based on aggression levels on a 5 point scale by 7 raters (Krippendorff’s alpha = 0.77). Here, the annotation based on audio footage is used. The SD dataset contains scenarios of problematic interactions situated at a service desk (e.g., a slow and incompetent employee while the customer has an urgent request). It contains 8 subjects and the recordings were annotated for stress levels on a 5 point scale by 7 raters (Krippendorff’s alpha = 0.74), based on audio-visual footage. All original labels were mapped onto a 3 point scale: SD classes 1 and 2 and TR class 1 onto Low, SD class 3 and TR class 2 onto Medium, and the rest of the data onto High escalation. The language spoken in the Escalation Corpus is Dutch (two scenarios from SD where English was spoken were excluded). Manual transcriptions are provided. The corpus has been re-segmented based on linguistic information, resulting in 413 and 501 (test) segments, of an average length of 5 seconds. The challenge task is to use the SD dataset for training, and to recognise escalation levels in the TR dataset.

2.3. The Primates Sub-Challenge (PRS)

For the PRS, the Primate Vocalisations Corpus described in Zwerts et al. [9] is used. The global biodiversity crisis calls for effective monitoring methods to measure, manage and conserve wildlife. Using acoustic recordings is a non-invasive and potentially cost-effective way to identify and count species for environments like tropical forests, where opportunities for visual monitoring are limited. Several studies have applied automatic acoustic monitoring for a variety of taxa, ranging from birds [10] to forest elephants [11], and sporadically also for primates [12, 13, 14]. Zwerts et al. [9] recently collected acoustic data from a primate sanctuary in Cameroon. The recorded species were Chimpanzees (Pan troglodytes), Mandrills (Mandrillus sphinx), Red-capped mangabeys (Cercocebus torquatus) and a mixed group of Guenons (Cercopithecus spp.). The sanctuary houses primates under semi-natural conditions making background noise relatively comparable to natural forests, albeit less rich in biodiversity and also containing human related noise. Recordings were made between December 2019 and January 2020 with a timespan of 32 days, using Audiomoth (v1.1.0) recorders [15], mounted either on the fence or nearby the respective species’ enclosure, with 48 kHz sampling rate and 30.6 dB gain, yielding 358 GBs of acoustic data, with a total duration of 1112 hours [9]. A semi-automatic annotation process speeded up the manual annotation efforts, with 1) initial annotation based on spectrogram analysis and listening, 2) vocalisation detection based on energy/variation in certain frequency sub-bands (150 Hz - 2 KHz), and 3) final annotation based on spectrogram analysis and listening, yielding over 10k annotated vocalisations. For the background class, the recordings not annotated as vocalisation were sampled so as to exactly match the duration distribution of the annotated chunks of each species [9].

3. Experiments and Results

For all corpora, the segmented audio was converted to single-channel 16 kHz, 16 bits PCM format. Table 1 shows the number of cases for Train, Dev, and Test for the databases; partitions for CCS, CSS, and ESS were gender-balanced.

3.1. Approaches

**COMPARE Acoustic Feature Set:** The official baseline feature set is the same as has been used in the eight previous editions of the COMPARE challenges, starting from 2013 [16]. It contains 6373 static features resulting from the computation of functionals (statistics) over low-level descriptor (LLD) contours [17, 16]. A full description of the feature set can be found in [18].

**Bag-of-Audio-Words (BoAWs):** These have been applied successfully for, e.g., acoustic event detection [19] and speech-
Table 2: Results for the four Sub-Challenges. The official baselines for Test are highlighted (bold and greyscale); there are no official baselines for Dev. C: Complexity parameter of the SVM, for all from $10^{-3}$ to 1, only best result. N: Codebook size for Bag-of-Audio-Words (BoAW) splitting into two codebooks (COMPARE-LLDs/COMPARE-LLD-deltas) of the same given size, with 50 assignments per frame. DenseNet121: pre-trained CNN used for extraction of Deep Spectrum features. X: Threshold power levels for S2S AE under which was clipped. DfE: Linguistic feature extraction pipeline and SVM. END2Y: End-to-end learning with convolutional recurrent neural network hidden units $N_a$. UAR: Unweighted Average Recall. CSS: COVID-19 Coughing. ESS: Escalation Sub-Challenge. PRS: Primates Sub-Challenge. CI on Test: confidence intervals for Test, see explanation in text.

<table>
<thead>
<tr>
<th></th>
<th>CCS UAR [%]</th>
<th>CSS UAR [%]</th>
<th>ESS UAR [%]</th>
<th>PRS UAR [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dev</td>
<td>Test</td>
<td>CI on Test</td>
<td>Dev</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-15</td>
<td>60.7</td>
<td>66.7</td>
<td>59.7-75.3</td>
<td>64.5-65.5</td>
</tr>
<tr>
<td>-10</td>
<td>64.4</td>
<td>68.6</td>
<td>62.0-70.0</td>
<td>68.1-70.6</td>
</tr>
<tr>
<td>-5</td>
<td>67.6</td>
<td>70.8</td>
<td>63.9-74.6</td>
<td>70.7-72.8</td>
</tr>
<tr>
<td>-0</td>
<td>69.4</td>
<td>72.5</td>
<td>65.3-76.0</td>
<td>72.3-74.7</td>
</tr>
<tr>
<td>+5</td>
<td>72.1</td>
<td>75.1</td>
<td>70.3-78.6</td>
<td>75.3-77.9</td>
</tr>
<tr>
<td>+10</td>
<td>74.2</td>
<td>77.5</td>
<td>71.5-80.5</td>
<td>77.7-80.2</td>
</tr>
<tr>
<td>+15</td>
<td>76.4</td>
<td>79.2</td>
<td>73.6-81.5</td>
<td>80.2-82.7</td>
</tr>
</tbody>
</table>

**CSS: COVID-19 Coughing. ESS: Escalation Sub-Challenge. PRS: Primates Sub-Challenge. CI on Test: confidence intervals for Test, see explanation in text.**

Based emotion recognition [20]. Audio chunks are represented as histograms of acoustic LLDs, after quantisation based on a codebook. One codebook is learnt for the 65 LLDs from the COMPARE feature set, and another one for the 65 deltas of these LLDs. In Table 2, results are given for different codebook sizes. Codebook generation is done by random sampling from the LLD/deltas in the training data. Each LLD/delta is assigned to the 10 audio words from the codebooks with the lowest Euclidean distance. Both BoAW representations, one from the LLDs and one from the deltas, are concatenated. Finally, a logarithmic term frequency weighting is applied to compress the numeric range of the histograms. LLDs are extracted with the OPENSMILE toolkit, BoAWs are computed using OPENXBOW [21].

**Deep Spectrum:** The feature extraction Deep Spectrum toolkit is applied to obtain first deep representations from the input audio data utilising pre-trained convolutional neural networks (CNNs) [22]. Deep Spectrum features have been shown to be effective, e.g., for speech processing [23]. First, audio signals are transformed into mel-spectrogram plots using a Hanning window of width 32 ms and an overlap of 16 ms. From these, 128 Mel frequency bands are computed. The spectrograms are then forwarded through DenseNet121 [24], a pre-trained CNN, and the activations of the ‘avg_pool’ layer of the network are extracted, resulting in a 2048-dimensional feature vector.

**AUDEEP:** Another feature set is obtained through unsupervised representation learning with recurrent sequence to sequence autoencoders, using AUDEEP [25, 26]. These explicitly model the inherently sequential nature of audio with Recurrent Neural Networks (RNNs) within the encoder and decoder networks [25, 26]. Here, Mel-scale spectrograms are first extracted from the raw waveforms in a data set. In order to eliminate some background noise, power levels are clipped below four different given thresholds in these spectrograms, which results in four separate sets of spectrograms per data set. Subsequently, a distinct recurrent sequence to sequence autoencoder is trained on each of these sets of spectrograms in an unsupervised way, i.e., without any label information. The learnt representations of a spectrogram are then extracted as feature vectors for the corresponding instance. Finally, these feature vectors are concatenated to obtain the final feature vector. For the results shown in Table 2, the autoencoders’ hyperparameters were not optimised.

**DfE:** Escalation is marked by an increase in arousal coming from acoustic rather than linguistic features; yet, semantic notations might additionally play a role [27, 28]. To this aim, we developed a lightweight Dutch Linguistic Feature Extractor (LFE) tool kit at [https://github.com/auDeep/auDeep](https://github.com/auDeep/auDeep), only best result. N: Codebook size for ..., the 10 audio words from the codebooks with the lowest Euclidean distance. Both BoAW representations, one from the LLDs and one from the deltas, are concatenated. Finally, a logarithmic term frequency weighting is applied to compress the numeric range of the histograms. LLDs are extracted with the OPENSMILE toolkit, BoAWs are computed using OPENXBOW [21].
compression, the pipeline uses a bidirectional Long Short-Term Memory (LSTM) RNN with an attention module (BLAtt), followed by two feedforward layers. The output of this last layer is used as feature input for the SVM evaluation.

**End2You**: We utilise the multimodal profiling toolkit End2You [33] to perform end-to-end learning. For our purposes, we utilise the Emo-18 [34] deep neural network that uses a convolutional network to extract features from the raw time representation and then a subsequent recurrent network with Gated Recurrent Units (GRUs) which performs the final classification. For training the network, we split the raw waveform into chunks of 100 ms each (except for the PRS Sub-Challenge with chunks of 70 ms). These are fed into a three layer convolutional network comprised of a series of convolution and pooling operations which try to find a robust representation of the original signal. The extracted features are passed to a two layer GRU to capture the temporal dynamics in the raw waveform.

### 3.2. Challenge Baselines and Interpretation

For the sake of transparency and reproducibility of the baseline computation, in line with previous years, we use an open-source implementation of SVMs with linear kernels. The provided scripts employ the Scikit-learn toolkit with its class LinearSVC for the classification based on functionals, BoAW, AuDeep, DIFe, and Deep Spectrum features. All feature representations were scaled to zero mean and unit standard deviation (StandardScaler of Scikit-Learn), using the parameters from the respective training (when Train and Dev were fused for the final classifier, the parameters were calculated on this fusion). The complexity parameter $C$ was always optimised during the development phase. Each Sub-Challenge package includes scripts that allow participants to reproduce the baselines and perform the testing in a reproducible and automatic way (including pre-processing, model training, model evaluation on Dev, and scoring by the competition and further measures). This year, we provide the six approaches outlined above. The same way as in the last three years, we chose the highest results on Test for defining the baselines, irrespective of the corresponding results on Dev, in order to prevent participants from surpassing the official baseline by simply repeating or slightly modifying other constellations that can be found in Table 2. A fusion of the best configurations (each different approach with its best parameters) with Majority Voting is given in the last row. As can be seen in Table 2, for CCS, the baseline is fusion of best with $UAR = 73.9\%$; for CSS, the baseline is based on Compare with $UAR = 72.1\%$; for ESS, BoAWs define the baseline with $UAR = 70.8\%$; and for PRS, the baseline is fusion of best with $UAR = 87.5\%$.

We provide two types of 95% confidence intervals, see the column ‘CI on Test’ in Table 2: First, we did 1000x bootstrapping for Test (random selection with replacement) and computed UARs, based on the same model that was trained with Train and Dev; the CI for these UARs is given before the slash. Then, we did 100x bootstrapping for the corresponding combination of Train and Dev, and employed the different models obtained from these combinations to get UARs for Test and subsequently, CIs, as displayed after the slash. Note that for this type of CI, the Test results are often above the CI, sometimes within and in a few cases below. Obviously, reducing the variability of the samples in the training phase with bootstrapping results on average in somehow lower performance.

Figure 1 displays the confusion matrices for the four sub-challenges for Dev corresponding to the best result on Test; e.g., for CCS, best Test result (without fusion) is 72.9% UAR for $N = 2000$; displayed is the confusion matrix corresponding to the UAR of 64.7%. Especially for CCS but for CSS as well, positive is frequently confused with negative, which may be tuned in a use case. For ESS, confusion between the extreme classes L and H are almost non-existent. The high UAR for PRS is mirrored by the high values in the diagonal – all five classes are predicted in a range of 10% absolute, from 79% to 89%.

### 4. Concluding Remarks

This year’s challenge is new by four new tasks (COVID-19 Cough and Speech, Escalation, and Primates), all of them highly relevant for applications. Besides the by now ‘classic’ approaches Compare and Bag-of-Audio-Words (BoAWs), we further featured sequence-to-sequence autoencoder-based audio features by the AuDeep toolkit, Deep Spectrum, a Dutch Linguistic Feature Extractor (DIFe) as well as End2End Deep Sequence Modelling. For all computation steps, scripts are provided that can, but need not be used by the participants. We expect participants to obtain better performance measures by employing novel (combinations of) procedures and features including such tailored to the particular tasks.

### 5. Acknowledgements

We acknowledge funding from the DFG’s Reinhart Koselleck project No. 442218748 (AUDIOMOUS), the EU’s HORIZON 2020 Grant No. 115902 (RADAR CNS), and the ERC project No. 833296 (EAR).

---

1. https://github.com/end2you/end2you

2. For PRS, only 10x was executed, because of the large number of data points.

3. This holds apart from End2You that would have required too time-consuming computation cycles.
6. References


