Expert Systems With Applications 205 (2022) 117613

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Estimating the degree of conflict in speech by employing Bag-of-Audio-Words and Fisher Vectors

Gábor Gosztolya *

*MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary*
*University of Szeged, Institute of Informatics, Szeged, Hungary*

ARTICLE INFO

*Keywords:*
Conflict intensity estimation
Machine learning
Bag-of-Audio-Words
Fisher Vectors

ABSTRACT

The automatic detection of conflict situations from human speech has several straightforward applications such as the surveillance of public spaces, providing feedback about employees in call centers, and other roles in human–computer interactions. In this study we examine the potential of different state-of-the-art feature extraction techniques, all developed to be able to efficiently represent a variable-length speech utterance by a fixed-length feature vector. Besides the 'ComParE functionals' attribute set, which became the de facto standard feature set in the area of computational paralinguistics (which focuses on the automatic assessment of non-verbal phenomena being present in human speech), we experiment with two methods introduced quite recently: Bag-of-Audio-Words (BoAW) and Fisher Vectors (FV). Using three standard basic, low-level feature sets, we found that, while BoAW proved to be quite sensitive to its meta-parameter settings, with Fisher Vectors we were able to achieve state-of-the-art conflict intensity estimation performance on a public and widely-used corpus. Furthermore, by applying Principal Component Analysis on the frame-level attributes, we managed to achieve a 30% speed-up in the feature extraction step. Interestingly, in contrast with our previous paralinguistic studies, combining the different predictions with these feature extraction approaches, we were unable to achieve any further significant improvement. The highest correlation coefficient values we got on the test set lay in the range 0.850–0.860, while the authors of several previous studies were able to achieve similar values (i.e. 0.849, 0.856 and 0.853). Considering that in this task the target score to be estimated (i.e. the intensity of the conflict being present in the actual clip) is definitely prone to subjectivity and therefore to label noise, current efforts have probably achieved the highest correlation coefficients attainable, and match human performance.

## 1. Introduction

Conflicts are an inherent part of everyday human communication, either in their personal or in their public life. In a conflict situation, the people involved are pursuing incompatible goals (Hocker & Wilmot, 1995); each party involved perceives that their interests are being opposed or adversely affected by the other party (Rubin et al., 1994). This usually can lead to conversations being more intense than usual, manifesting itself in raised voices and in a greater number of interruptions (Kim et al., 2014).

Unfortunately, conflicts are one of the main causes of stress (Spector & Jex, 1998), and long-term stress leads to both mental and physical health problems such as depression, high blood pressure and eating disorders; therefore, early detection of conflict situations (and, of course, preventing their escalation) would be useful. With the rise of socially intelligent technologies, the automatic detection of conflicts could be the first step of handling them properly. Besides this, there are also straightforward applications for automatic conflict detection, such as helping security in public or publicly accessible areas, and monitoring incoming calls in call centers, where important feedback of the employees includes how they are able to handle conflicted situations.

Although indicators of conflict can be found in the use of specific words and phrases, in conversations conflicts are mostly expressed by non-verbal messages such as gestures, facial expressions, interruptions, prosody and speech intensity. For example, in the heat of a debate speakers tend to interrupt each other more frequently than usual, which leads to a more frequent occurrence of overlapping speech (Cooper, 1986; Ferguson, 1977), increasing speech signal intensity (i.e. volume) and articulation tempo (Kim et al., 2012).

It has also been shown that, to estimate the intensity of conflicts, non-verbal cues present in the speech of the parties involved contain most of the information, while the visual cues (i.e. gestures and facial expressions) offer no additional significant information (Kim et al.,

* Correspondence to: University of Szeged, Institute of Informatics, Szeged, Hungary.
*E-mail address:* ggabor@inf.u-szeged.hu.

https://doi.org/10.1016/j.eswa.2022.117613
Received 31 May 2020; Received in revised form 15 July 2021; Accepted 15 May 2022
Available online 28 May 2022

2012). Therefore, most scientific studies focus just on the audio signal, and ignore visual cues (see e.g. Caraty & Montacié, 2015; Kim et al., 2014, 2012; Rajan et al., 2019); of course, in some cases, the latter are also exploited (Georgakis et al., 2017; Panagakis et al., 2014). For the case of automatic conflict detection methods, this in practice means that it is sufficient to process just the speech of the subjects, while the video track (if any) can be discarded.

In this study, we will focus on conflict detection from audio. More precisely, we perform conflict intensity estimation: instead of just seeking to determine the presence of conflict situations, we also attempt to estimate how severe the actual disagreement was. From a machine learning perspective, this means that we should interpret our task as a regression one, where the target value corresponds to the actual conflict strength, instead of solving a binary classification task with the two classes "conflict" and "no conflict". From a wider perspective, this task belongs to the general domain of computational paralinguistics, which focuses on the non-verbal content of human speech (Schuller & Batliner, 2013).

Computational paralinguistics, despite being a relatively young field inside speech technologies, has already developed its standard set of tools and approaches. Although there exist some studies that apply AdaBoost.MH (Gosztolya et al., 2014; Schmitt & Minker, 2012), the extreme learning machine (Kaya & Salah, 2014) and Deep Neural Networks (Grósz et al., 2015; Huckvale & Beke, 2017), the standard method of machine learning applied is still that of Support Vector Machines (SVMs, Schölkopf et al., 2001). Regarding evaluation metrics, depending on the actual dataset, Pearson's or Spearman's correlation coefficient (CC) have become the standard (Grzybowska & Kacprzak, 2016; Kaya & Karpov, 2016; Schuller et al., 2016; Sztahó et al., 2015). As for classification problems, due to the imbalanced distribution of recordings corresponding to the classes (mirroring real-life distribution of specific events such as the different emotions), the so-called Unweighted Average Recall (UAR, Schuller et al., 2013) metric is normally used, which involves taking the mean of the class-wise recall scores.

In contrast, presently there is no consensus on which types of features are worth extracting from the speech recordings. Of course, the audio pre-processing scheme of calculating the spectral representation of the speech signal, and taking the Mel-scale filter band energies is borrowed from the automatic speech recognition (ASR) task. This leads to the concept of *frames*, i.e. small, equal-sized speech excerpts, usually calculated with a 10 ms step interval (i.e. we have 100 frames for each second of an utterance). In ASR, however, frame-level feature vectors can be used in the next step directly, as there the main classification step (namely phoneme classification) is performed at the frame level. (After this local phoneme classification step, the local class-conditional probability estimates are combined via a Hidden Markov Model to produce utterance-level hypotheses Bourlard & Morgan, 1994.)

In computational paralinguistics, however, where classification or regression is done at the segment level (e.g. treating each utterance as an independent example), simple frame-level feature extraction is not enough by itself. The reason for this is simple: most classification and regression methods require a fixed-size feature vector for each instance; however, in computational paralinguistics each utterance corresponds to one such instance, and the number of its frames is proportional to the duration, which is subject to vary. To apply these machine learning algorithms, we need feature extraction methods that can provide a feature vector with a size that is independent of the length of the recordings.

In this study we tested three such feature extraction approaches, which all rely on the frame-level feature vectors of the given utterance, but summarize them at the segment level in a completely different way. The first approach ('ComParE functionals') utilizes functions such as mean, standard deviation, and percentile statistics (Schuller et al., 2013). Bag-of-Audio-Words (or BoAW, Pancoast & Akbacak, 2012) defines clusters over the frame-level feature vectors; then, for a given utterance, the BoAW procedure categorizes each frame into one of these pre-defined clusters, and calculates the utterance-level feature vector as the frequency of the clusters. In contrast, the Fisher Vectors (or FV, Jaakkola & Haussler, 1998) representation models the distribution of the frame-level feature vectors via some generative method, usually by Gaussian Mixture Models (GMMs, Rabiner & Juang, 1993). The feature vector of an utterance is determined by measuring the change in the model parameters (e.g. mean and standard deviation of the Gaussian components in the case of GMMs) when it is adjusted to best fit the frames of the actual recording. To the best of our knowledge, this is the first study ever to utilize the Bag-of-Audio-Words and Fisher Vector techniques for conflict detection.

**Table 1**
Some key properties of the SSPNet Conflict Corpus.

| Set | No. of clips | Total duration | Conflict scores |
|---|---|---|---|
| Training | 793 | 6:36:13 | −0.68 ± 3.98 |
| Development | 240 | 2:00:00 | −0.21 ± 3.75 |
| Test | 397 | 3:18:16 | −0.58 ± 3.98 |
| Total | 1430 | 11:54:29 | −0.58 ± 3.94 |

The structure of this paper is as follows. In Section 2, we introduce the SSPNet Conflict corpus, which will be used in our experiments. Then, in Section 3, we briefly present the ComParE functionals feature extraction approach. Similarly, we describe the Bag-of-Audio-Words and the Fisher Vectors methods in Sections 4 and 5, respectively. We commence with a detailed explanation of our experimental setup in Section 6; then, in Section 7 we present and analyze our experimental results. Finally, in Section 9 we discuss classifier combination experiments.

## 2. The SSPNet Conflict Corpus

The SSPNet Conflict Corpus (Kim et al., 2014) contains recordings of Swiss French political debates taken from the TV channel "Canal9". It consists of 1430 recordings, 30 s each, making a total of 11 h and 55 min. The ground truth level of conflicts was determined by manual annotation performed by volunteers who did not understand French (French-speaking people were excluded from the list of annotators). Each 30-second long clip was tagged by 10 annotators; in the end each recording was assigned a score in the range [−10, 10], 10 denoting a very high level of conflict and −10 denoting the absence of conflicts. Although the database contains both audio and video recordings, following previous studies (see e.g. Brueckner & Schuller, 2015; Caraty & Montacié, 2015; Kaya, Özkaptan et al., 2015; Räsänen & Pohjalainen, 2013), we will rely on the audio data only, and discard the video track.

The audio clips of this dataset were later used in the Conflict sub-challenge of the Interspeech 2013 Computational Paralinguistic Challenge (or ComParE 2013 Schuller et al., 2013). Besides completely discarding video data, other steps were made to standardize the work on this dataset, and this setup has since been adopted by most researchers. Perhaps the most important one was that, instead of relying on cross-validation as Kim et al. did (Kim et al., 2014), separate training and test sets were defined. Some key properties of the whole SSPNet Conflict corpus and the training, development and test subsets can be seen in Table 1, while the distribution of the conflict scores is shown in Fig. 1. It can be seen that there are more clips with smaller conflict intensity values than those with high ones. Clearly, the distribution of the conflict scores is quite similar for all three subsets; overall the development set has a slightly higher mean conflict score, but the difference is probably not statistically significant.

The standard evaluation metrics used for this dataset were also defined in the ComParE challenge. Schuller et al. admitted that this was mainly a regression task and used Pearson's correlation coefficient (CC) to measure the performance. They, however, also converted the task into a binary classification one, defining the classes *low* and *high* based on the sign of the conflict score (Schuller et al., 2013). Classification

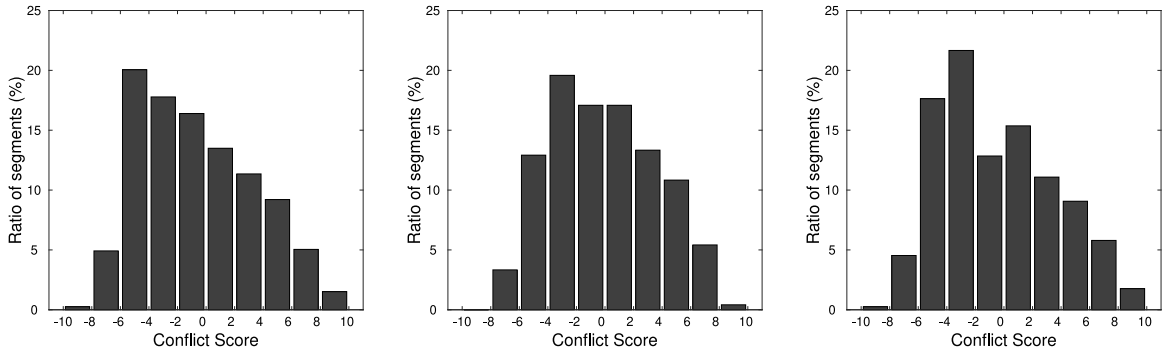


**Fig. 1.** Distribution of the conflict intensity of the clips in the training (left), development (middle) and test sets (right).

**Table 2**
Pearson's Correlation Coefficient (CC) and UAR scores given in the literature for the test set of the SSPNet Conflict Corpus, following the ComParE 2013 setup. "—" means that the given score was not provided.

| Method | CC | UAR |
|---|---|---|
| 2013 ComParE Challenge baseline (Schuller et al., 2013) | 0.816 | 80.8% |
| Speaker overlap (Grèzes et al., 2013) | – | 83.1% |
| Random Subset Feature Selection (Räsänen & Pohjalainen, 2013) | 0.826 | 83.9% |
| Speaker overlap + prosodic features (Brueckner & Schuller, 2015) | 0.838 | 84.3% |
| SLCCA Feature Selection (Kaya, Özkaptan et al., 2015) | – | 84.6% |
| Speaker Interruption (Caraty & Montacié, 2015) | – | 85.3% |
| Greedy Forward Feature Selection (Gosztolya, 2015) | 0.835 | 85.6% |
| Greedy Forward + Backward Feature Selection (Gosztolya, 2015) | 0.842 | 85.1% |
| Ensemble Nyström method (Huang et al., 2014) | 0.849 | – |
| Predicted Speaker Overlap (Gosztolya & Tóth, 2017) | 0.816 | – |
| Feature Selection + Predicted Speaker Overlap (Gosztolya & Tóth, 2017) | 0.856 | – |
| Spectrogram-based CNN (Segura et al., 2016) | 0.793 | 78.4% |
| End-to-end CNN + Average Pooling + Attention (Rajan et al., 2019) | 0.853 | 84.3% |

accuracy was measured by the Unweighted Average Recall (UAR) value; this metric was used both in the Challenge (e.g. Grèzes et al., 2013; Räsänen & Pohjalainen, 2013), and it has been used in research studies since then (e.g. Brueckner & Schuller, 2015; Caraty & Montacié, 2015; Kaya, Özkaptan et al., 2015). In our view, treating this task as a regression one is the proper approach, partly because representing conflict intensity as a numeric value contains more information than a binary class label, and also because we found that optimizing for the CC value leads to more robust models than maximizing the UAR score. (For details, see Gosztolya (2015).) Due to this, in this study we will primarily rely on the CC metric, and also include the UAR scores.

Table 2 lists the notable scores published in the literature for this dataset. It also shows specific trends: most of the early attempts either applied feature selection (Gosztolya, 2015; Kaya, Özkaptan et al., 2015; Räsänen & Pohjalainen, 2013) or utilized the amount of speaker overlap in some way (Brueckner & Schuller, 2015; Caraty & Montacié, 2015; Gosztolya & Tóth, 2017; Grèzes et al., 2013). We may also notice that more recent studies tend to utilize Deep Neural Networks in some way (Gosztolya & Tóth, 2017; Rajan et al., 2019; Segura et al., 2016).

## 3. The ComParE functionals feature set

As the first utterance-level feature extraction approach, we used the 'ComParE functionals' attributes developed by Schuller et al. (2013). The feature set includes energy, spectral, cepstral (MFCC) and voicing related frame-level features, from which specific functionals (e.g. mean, standard deviation, 1st and 99th percentiles, peak statistics etc.) are computed to provide utterance-level feature values. From the 65 frame-level attributes and their first-order derivatives, an utterance-level set with 6373 features is calculated overall. Over the years since its introduction, the ComParE functionals feature set, without a doubt, has evolved into the de facto standard solution for computational paralinguistics, and was utilized, among others, in tasks such as estimating speaker age (Grzybowska & Kacprzak, 2016), sleepiness (Schuller et al., 2017), sincerity (Schuller et al., 2016) and determining whether the speaker has a cold (Schuller et al., 2017). This feature set was extracted by using the openSMILE tool (Eyben et al., 2010).

## 4. Bag-of-audio-words representation

The second utterance representation technique we employed in this study is the Bag-of-Audio-Words approach. Although it is not as popular as ComParE functionals, in the past few years it has been used on a wide variety of tasks such as multimedia event classification (Pancoast & Akbacak, 2012), emotion recognition (Pokorny et al., 2015; Schmitt, Ringeval et al., 2016), acoustic event detection (Lim et al., 2015), snore sound classification (Schmitt, Janott et al., 2016) and determining whether a speaker has a cold (Schuller et al., 2017).

The BoAW approach, similarly to the ComParE functionals approach, also relies on the frame-level feature vectors; for the first step, we process the training set, where we perform a clustering on the set of all the input frame-level feature vectors. The number of clusters ($N$) is the key parameter of the method. The list of the resulting cluster centroids will form the *codebook*. Next, each original feature vector is replaced by a single index representing the nearest codeword (*vector quantization*). The feature vector for each utterance will be calculated by generating a histogram of these cluster indices; it is common to further apply some kind of normalization technique such as L1 normalization (i.e. divide each cluster count by the number of frames in the given utterance). Since the number of clusters is a meta-parameter of the BoAW method, each utterance will be represented by a vector of the same length (i.e. $N$), independently of the original length of the individual utterances. These fixed-length feature vectors can be used for utterance-level classification (for example, by using a Support Vector Machine) in the third step. For the mechanism of the Bag-of-Audio-Words process, see Fig. 2.

Notice that the BoAW representation can be readily calculated for the test set as well: a feature vector unseen during the clustering

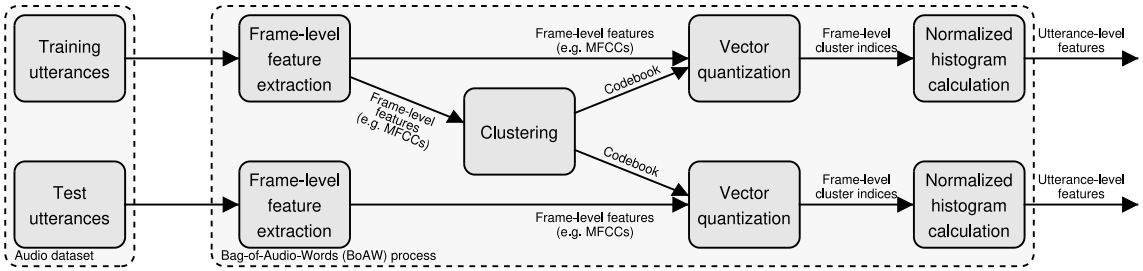


**Fig. 2.** Workflow of the Bag-of-Audio-Words representation used for audio processing.

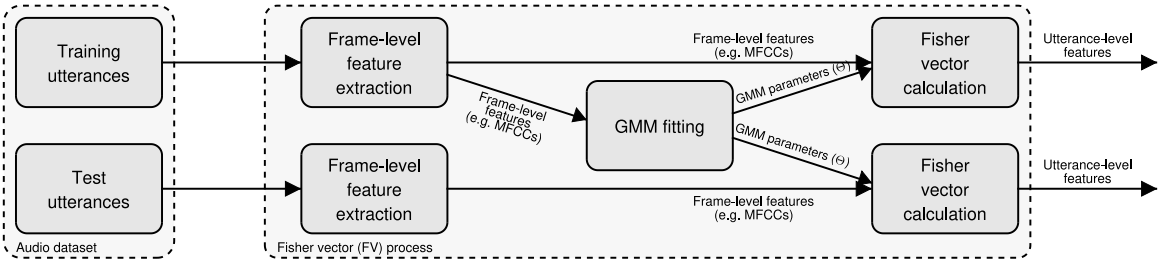


**Fig. 3.** Workflow of the Fisher vector representation used for audio processing.

step (such as those of the test set) can be assigned to one of the previously defined clusters based on its Euclidean distance from the cluster centers. (See the lower path of Fig. 2.)

Over the years, several refinements have been proposed for the original BoAW approach. For example, Pancoast and Akbacak argued that simply choosing the nearest cluster center is a crude simplification which in fact harms classification performance. To resolve it, they proposed a soft quantization representation where the distance to the nearest codeword is incorporated into the model, for example by choosing a fixed number of closest cluster centers for each frame (Pancoast & Akbacak, 2014). This trick is known to improve the classification performance (see e.g. Pancoast & Akbacak, 2014; Vetráb & Gosztolya, 2019), and it does not increase the execution time of the BoAW process significantly. Regarding the clustering step, Rawat et al. found that using simple random sampling of the input frames as codewords leads to similar accuracy scores to those using clustering for codebook creation (Rawat et al., 2013), and it is evidently significantly faster. For the above reasons, these modifications became standard practice, hence we will also apply these improvements in our experiments.

## 5. Fisher Vectors

Recalling the motivation behind the Bag-of-Audio-Words technique, a similar and perhaps even more informative representation approach is that of Fisher vectors (Jaakkola & Haussler, 1998). Fisher vectors were introduced in the image processing community, where a similar problem arose as in computational paralinguistics: handling a varying number low-level descriptors for each object. That is, in image processing first SIFT descriptors (describing occurrences of rotation- and scale-invariant primitives Lowe, 2004) are extracted from the images; of course, different images contain a different number of SIFT descriptors. FVs solve this issue by modeling the distribution of SIFTs using some generative method (e.g. Gaussian Mixture Models), and then they measure the change in the model parameter values when it is adjusted to best fit the SIFTs of the actual image.

The aim of the Fisher vector representation was to combine the generative and discriminative machine learning approaches by deriving a kernel from a generative model of the data (Jaakkola & Haussler, 1998). That is, let $X = \{x_1, \ldots, x_T\}$ be $d$-dimensional low-level feature vectors extracted from an input sample, and let their distribution be modeled by a probability density function $p(X|\Theta)$, $\Theta$ being the parameter vector of the model. The Fisher score describes $X$ by the gradient $G_\Theta^X$ of the log-likelihood function, i.e.

$$G_\Theta^X = \frac{1}{T}\nabla_\Theta \log p(X|\Theta). \tag{1}$$

This gradient function practically corresponds to the direction in which the model parameters (i.e. $\Theta$) should be changed to best fit the data. Notice that the size of $G_\Theta^X$ is already independent of the number of low-level feature vectors (i.e. of $T$), and it depends only on the number of model parameters (i.e. $\Theta$). The Fisher kernel between the sequences $X$ and $Y$ is then defined as

$$K(X,Y) = G_\Theta^X F_\Theta^{-1} G_\Theta^Y, \tag{2}$$

where $F_\Theta$ is the Fisher information matrix of $p(X|\Theta)$, defined as

$$F_\Theta = E_X[\nabla_\Theta \log p(X|\Theta) \nabla_\Theta \log p(X|\Theta)^T]. \tag{3}$$

Expressing $F_\Theta^{-1}$ as $F_\Theta^{-1} = L_\Theta^T L_\Theta$, we get the Fisher vectors as

$$\mathcal{G}_\Theta^X = L_\Theta G_\Theta^X = L_\Theta \nabla_\Theta \log p(X|\Theta). \tag{4}$$

When we utilize Gaussian Mixture Models to model the distribution of the low-level features (i.e. $p(X|\Theta)$) and assume a diagonal covariance matrix, the Fisher vector representation of an instance has a length of twice the number of Gaussian components for each feature dimension.

To apply Fisher vectors to audio processing, it is straightforward to use some standard frame-level features (e.g. MFCCs Rabiner & Juang, 1993) of the utterances as the low-level features (i.e. $X$). When using GMMs, a parameter of the method is the number of Gaussian components ($N$). The workflow of Fisher vectors used in audio processing is shown in Fig. 3.

Fisher Vectors was only recently discovered in audio processing. However, it has already been utilized for categorizing audio files as speech, music and other (Moreno & Rifkin, 2010), for speaker verification (Tian et al., 2014; Zajíc & Hrúz, 2016), for emotion recognition (Chen et al., 2016), for determining food type from eating sounds (Kaya, Karpov et al., 2015), for identifying Orca sounds (Wu et al., 2019), for detecting sleepiness (Gosztolya, 2019b), and for identifying Parkinson's disease (Egas López et al., 2019) and depression (Jain et al., 2014).

## 6. Experimental setup

### 6.1. Frame-level feature sets

Although the 'ComParE functionals' attribute set was designed to work on a specific frame-level feature set, both for the Bag-of-Audio-Words approach and for Fisher Vectors we have the possibility of employing various such feature sets. Of course, the choice of frame-level features can be expected to influence the final regression performance, but we cannot know in advance which type of attributes lead to the best classification or regression scores. Therefore, we tested three frame-level feature sets for these two utterance-level feature extraction techniques. The first one was the well-known Mel-frequency cepstral coefficients (MFCCs, Rabiner & Juang, 1993); we calculated 13 coefficients, along with the first and second order derivatives (i.e. MFCC + $\Delta$ + $\Delta\Delta$, 39 attributes overall). MFCCs were calculated using the HTK tool (Young et al., 2006).

The second frame-level attribute set we utilized was the raw Mel-frequency energy filter banks: we employed 40 bands and the energy of the signal, which, along with the $\Delta$ and $\Delta\Delta$ values, came to 123 feature values for each frame. Again, we utilized HTK to extract these values from the recordings (Young et al., 2006). As the last frame-level feature set, we chose the one that is utilized in the 'ComParE functionals' (utterance-level) feature set: it consists of four energy-related feature (including loudness, energy and Zero-Crossing-Rate), 55 spectral attributes (e.g. MFCCs, spectral energies and variances, skewness, kurtosis) and 6 voicing-related one (such as $F_0$, probability of voicing, logarithmic Harmonic-to-Noise Ratio, Jitter and Shimmer). These 65 frame-level attributes and their $\Delta$ values ('ComParE' frame-level feature set for short) were calculated by the OpenSMILE tool (Eyben et al., 2010).

For both the Bag-of-Audio-Words technique and for the Fisher Vectors approach it may make sense to use the $\Delta$ and $\Delta\Delta$ values; however, it might happen that these attributes just cause overfitting, and we can build a more robust (and more compact) model without them. Therefore, both approaches were tried in three variations: without using the derivatives, using the first-order derivatives only, and using both first and second order derivatives (with the exception of the ComParE frame-level feature set, where no $\Delta\Delta$ values were extracted).

### 6.2. Bag-of-Audio-Words parameters

We used the OpenXBOW package (Schmitt & Schuller, 2017), which is an open-source toolkit written in Java. We tested codebook sizes of 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 and 16384. We employed random sampling instead of k-means or k-means++ clustering for codebook generation and we allowed 5 parallel cluster assignments, i.e. for each frame we chose the 5 closest cluster centers. Before processing the frame-level features, we performed standardization in each case. As mentioned in Section 4, we experimented with building one codebook for the whole frame-level feature sets (i.e. including first and second order derivatives) as well as building separate codebooks for the basic attributes, the $\Delta$s and the $\Delta\Delta$ values. We also experimented with discarding the different derivatives, which led to four cases for the MFCC and FBANK cases and three for the ComParE feature set. Note that the largest utterance-level BoAW feature set ($N = 16384$ with three separate codebooks) consisted of 49152 attributes.

### 6.3. Fisher Vector parameters

We used the open-source VLFeat library (Vedaldi & Fulkerson, 2010) to fit GMMs and to extract the FV representation; from the various ports available, we employed the Matlab integration. When fitting Gaussian Mixture Models, we experimented with $N = 2, 4, 8, 16, 32, 64$ and 128 components. The number of extracted (utterance-level) features lay between 52 (MFCCs without any derivatives, $N = 2$) and 33 280 (ComParE + $\Delta$ features, $N = 128$).

### 6.4. Utterance-level regression

Our experiments followed standard paralinguistic protocols. After feature standardization, we used SVM with linear kernel for utterance-level regression (method $\nu$-SVR), using the LibSVM (Chang & Lin, 2011) library; the value of $C$ was tested in the range $10^{\{-5,\ldots,1\}}$, just like in our previous paralinguistic studies (e.g. Gosztolya, 2019c; Gosztolya et al., 2017, 2016). The optimal meta-parameter values ($C$ for SVM and $N$ both for BoAW and for Fisher vectors) were determined on the development set; finally, an SVR model was trained on the combined training and development sets, using the meta-parameters found.

## 7. Results

Fig. 4 shows the CC values obtained for the development set for the Bag-of-Audio-Words and the Fisher Vector techniques. Regarding the **BoAW approach**, it is clear that the smaller codebook sizes ($N \leq 64$) led to quite low scores. Overall, we got the best values with the MFCC features, followed by the ComParE attributes, while when we utilized the FBANK attributes, we always got CC values below 0.8. Regarding the use of the derivatives, when we did not use any $\Delta$ values, we almost always got the lowest scores. Using the whole frame-level feature vectors and building a common codebook for all the attributes turned out to be a better approach than this, but we got the highest scores by creating separate codebooks for the $\Delta$ (and sometimes the $\Delta\Delta$) values.

Table 3 shows the best CC values obtained for each tested approach along with the corresponding CC scores measured on the test set, and also the corresponding UAR classification percentages. (The best scores (within a small tolerance) for a given frame-level feature set are shown in **bold**.) Surprisingly, on the test set the MFCC features did not lead to really good results: the CC scores lay between 0.816 and 0.826. We should also mention that the configuration which proved to be the best on the development set with a CC value of 0.851 (using all the MFCC attributes, and building separate codebooks for the $\Delta$ and $\Delta\Delta$ attributes) led to a rather large feature set of nearly 50000 attributes. Regarding the UAR scores, the values between 78.6% and 81.8% are mediocre at best. (Although, of course, we listed the UAR scores for reference only, as we regard the correlation coefficient as a more appropriate and more reliable evaluation metric, and we chose all the meta-parameter values (i.e. $N$ and $C$) that gave the highest CC on the development set.) Overall, it seems that it was easy to overfit using the BoAW-MFCC approach.

When utilizing the FBANK attributes, we ended up with significantly lower CC scores on the development set (see also Fig. 4), but the scores were higher on the test set, which in our opinion suggests that it is less likely that we overfit using these frame-level features. Again, we got the lowest CC scores without any $\Delta$ attributes, but the other three cases all led to CC scores above 0.830 on the test set. Regarding the UAR values, in one case it was over 83% (on the test set), but comparing it to the scores reported earlier (see Table 2) it is clear that this score by no means outstanding. The corresponding $N$ values and feature set sizes are in the middle range, usually staying far below those of the MFCC cases; the fact that we needed smaller feature sets for the FBANK attributes means that we not only got better correlation scores by them, but they are also more feasible to employ in practice.

As for the ComParE frame-level features, we see test set performance values of 0.835–0.851. In fact, when we employed the first-order derivatives as well, and built a separate codebook for the basic and the $\Delta$ values, we got an average performance on the development set, but the CC score got on the test set appeared to be very high. Among the three tested frame-level attributes, the 'ComParE' one proved to be the most useful with the BoAW approach; notice, however, that we had to use the highest $N$ value tested (i.e. 16384) to achieve this performance.

Examining the performance of the **Fisher Vector** based models on the right hand side of Fig. 4, we can make similar observations as those

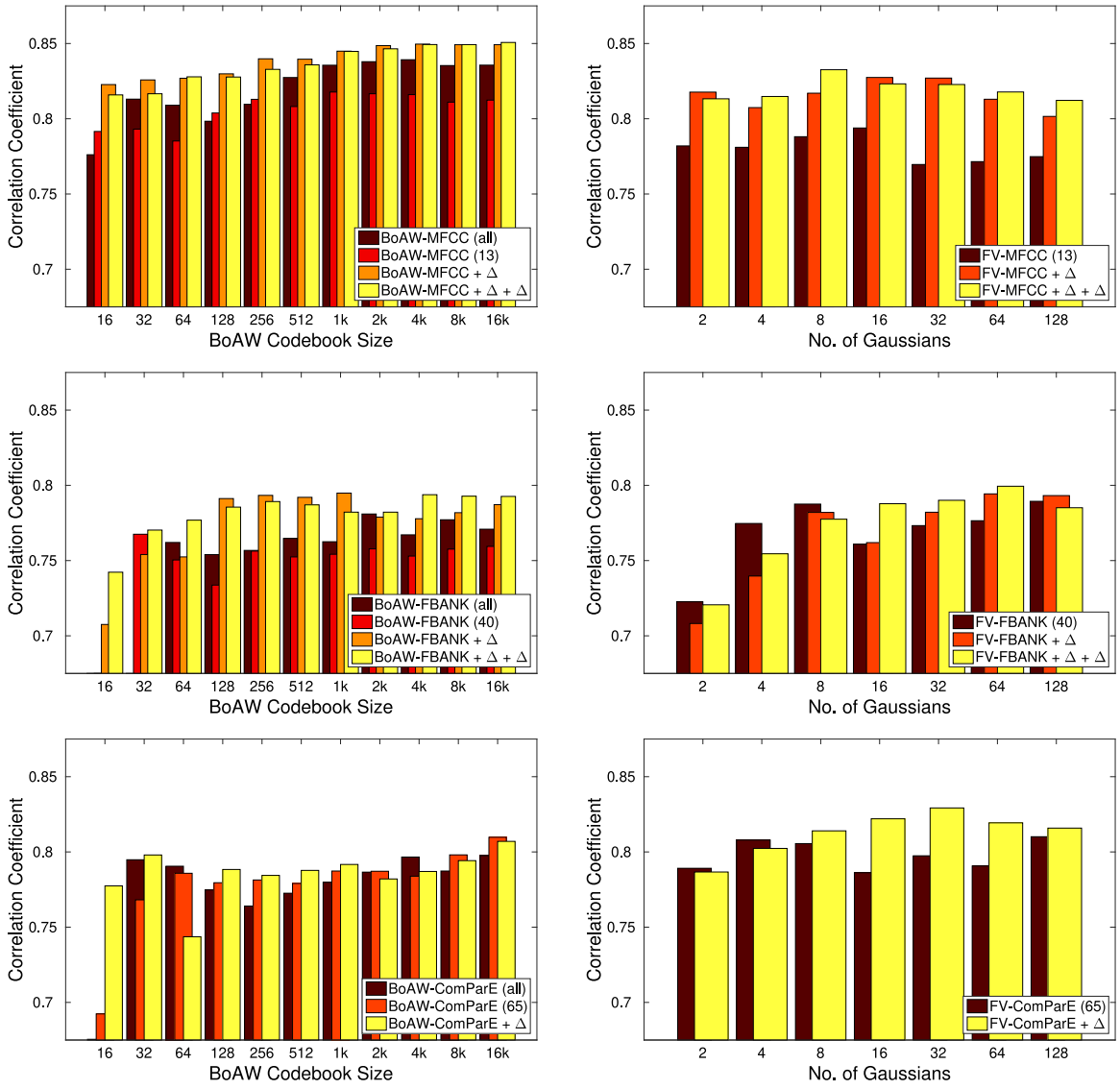


**Fig. 4.** Pearson's Correlation Coefficient (CC) values measured on the development set for the various frame-level feature sets tested for the Bag-of-Audio-Words (left) and the Fisher Vector (right) techniques.

**Table 3**
The performance scores obtained when using the Bag-of-Audio-Words and the ComParE functionals representations. Best values and those falling close to them are shown in **bold**.

| Codebook type | Frame-level feature set | $N$ | No. of features | Correlation | | UAR | |
|---|---|---|---|---|---|---|---|
| | | | | Dev. | Test | Dev. | Test |
| ComParE functionals | | — | 6373 | 0.820 | 0.835 | 79.7% | 84.8% |
| Common | MFCC + Δ + ΔΔ | 2048 | 2048 | 0.838 | **0.826** | 79.6% | **81.8%** |
| Separate | MFCC | 8192 | 8192 | 0.811 | **0.824** | 79.2% | 81.2% |
| | MFCC + Δ | 4096 | 8192 | **0.850** | **0.822** | **80.5%** | 81.2% |
| | MFCC + Δ + ΔΔ | 16384 | 49152 | **0.851** | **0.816** | 78.6% | 79.1% |
| Common | FBANK + Δ + ΔΔ | 2048 | 2048 | 0.781 | **0.845** | 77.4% | 82.3% |
| Separate | FBANK | 16384 | 16384 | 0.759 | 0.828 | 77.4% | 81.9% |
| | FBANK + Δ | 1024 | 2048 | **0.795** | **0.835** | **78.3%** | 81.6% |
| | FBANK + Δ + ΔΔ | 4096 | 12288 | **0.794** | 0.831 | 76.8% | **83.3%** |
| Common | ComParE + Δ | 16384 | 16384 | 0.798 | 0.835 | 77.4% | 81.5% |
| Separate | ComParE | 16384 | 16384 | **0.810** | 0.835 | 78.7% | 80.6% |
| | ComParE + Δ | 16384 | 32768 | **0.807** | **0.851** | **79.5%** | **83.7%** |

**Table 4**
The performance scores obtained when using the Fisher Vectors representation.

| Frame-level feature set | $N$ | No. of features | Correlation | | UAR | |
|---|---|---|---|---|---|---|
| | | | Dev. | Test | Dev. | Test |
| MFCC | 16 | 416 | 0.794 | 0.793 | **78.6%** | 79.2% |
| MFCC + $\Delta$ | 16 | 832 | 0.827 | 0.818 | 50.0% | 50.0% |
| MFCC + $\Delta$ + $\Delta\Delta$ | 8 | 624 | **0.833** | **0.844** | 71.4% | **82.5%** |
| FBANK | 128 | 10496 | **0.789** | 0.787 | 74.6% | 79.8% |
| FBANK + $\Delta$ | 64 | 2048 | **0.795** | **0.845** | 73.2% | 83.4% |
| FBANK + $\Delta$ + $\Delta\Delta$ | 64 | 12288 | **0.799** | **0.850** | **75.4%** | **86.1%** |
| ComParE | 128 | 16640 | 0.810 | **0.850** | 80.2% | **84.5%** |
| ComParE + $\Delta$ | 32 | 8320 | **0.829** | **0.856** | **82.7%** | **84.5%** |

**Table 5**
The number of frame-level attributes kept by PCA for the two frame-level feature sets.

| Preprocessing | Inform. kept | Feature set | |
|---|---|---|---|
| | | FBANK | ComParE |
| No standardization | 95% | 21 | 1 |
| | 99% | 45 | 2 |
| Standardization | 95% | 55 | 58 |
| | 99% | 91 | 92 |
| Total (i.e. no PCA) | | 123 | 130 |

for the BoAW technique. In most cases it was worth utilizing the first-order derivatives, but the second-order derivatives (if any) only led to a slight additional improvement. In general, MFCCs gave the best performance, followed by the ComParE frame-level features, while the FBANK attribute set clearly led to the lowest CC scores.

Examining the best CC values on the development set for each case tested (and, of course, the corresponding CC and UAR values on the test set) in Table 4, we see, however, that unlike that with the Bag-of-Audio-Words method, now we got quite high values on the test set as well. When using all feature-level attributes, we measured CC values of 0.844, 0.850 and 0.856, MFCC, FBANK and ComParE feature sets, respectively. Overall, it seems that the Fisher Vector representation is a more robust technique than Bag-of-Audio-Words: it was less likely to overfit, regardless of the actual frame-level attribute set we employed.

## 8. Applying principal component analysis

The Bag-of-Audio-Words and the Fisher Vectors representation approaches both summarize the frame-level feature vectors of each utterance. When doing this, they both treat each frame-level dimension as of equal importance, albeit in a different way: the BoAW process calculates the Euclidean distance of specific frames, while the FV process models each dimension with a separate Gaussian distribution. Clearly, when specific frame-level attributes are correlated (which is obviously the case, for example, for the neighboring filter banks in FBANK, since they overlap), it might cause the BoAW method and/or the FV method to treat them as more important ones than the other features, and this could lead to a suboptimal regression performance. Another drawback of this redundancy is manifested in the execution times. That is, the size of the feature set calculated by Fisher vectors is the product of $N$ and the size of the frame-level attribute set; and for BoAW, the number of operations required for assigning one frame-level attribute vector to the $N$ closest clusters is also proportional both to the number of frame-level features and to $N$. By reducing the number of frame-level attributes, we can achieve a proportional speed-up in execution times of the feature extraction step, while having fewer (utterance-level) attributes can also be expected to speed up the subsequent machine learning step.

To remove the redundancy present within the frame-level feature values, and to project the feature set into orthogonal space (ideal for GMMs when determining the Fisher Vectors), we could apply Principal Component Analysis (PCA, Jolliffe, 1986). Therefore, next we will present our experiments by first transforming the 'FBANK' and the 'ComParE' frame-level feature vectors by PCA, and apply the BoAW and FV methods in the second step. (We did not test this with MFCCs, since MFCCs already have quasi-orthogonal components.) We also tested whether it was worth standardizing the feature vectors before using PCA. As being standard for applying PCA, we decided to keep 95% and 99% of the total information; this led to $1-45$ dimensional vectors without standardization, and $55-92$ dimensional vectors when we first applied standardization (see Table 5).

Fig. 5 shows the correlation coefficient values we obtained on the development set. In general, we may conclude that it is worth standardizing the frame-level feature vectors before PCA, as in most cases, the corresponding CC values were higher than those obtained without this standardization step. Specifically, for the ComParE feature set we got quite low CC scores without standardization (i.e. in the range $0.333-0.659$) both for BoAW and for FV. The reason for this is probably that, without standardization, only a few directions held most of the information (see Table 5 again), but this did not make high-precision conflict intensity estimation possible.

When employing standardization before PCA, the values associated with the 95% and 99% cases were quite similar to each other and also to the original feature set (i.e. without PCA), although in 3 cases out of 4, retaining 95% of the information brought only a slight improvement. Examining the best CC values on the development set and the corresponding scores on the test set, we might conclude that applying PCA was actually not really efficient for improving the quality of predictions for the Bag-of-Audio-Words approach (see Table 6), as we experience significant drops for the FBANK feature set, and in only two cases and one case experienced practically the same performance, development and test sets, respectively. We could observe the same tendencies for Fisher Vectors (see Table 7): with filter banks we could only achieve a CC score of 0.830 on the test set, while when we used all original frame-level attributes, we measured a CC value of 0.850. For the ComParE frame-level attributes with Fisher vectors, the scores were slightly more competitive (CC values of 0.840 and 0.845 for the "std. + PCA (95%)" and "std. + PCA (99%)" cases, respectively), but we were again unable to surpass the case of omitting the PCA step.

However, as discussed above, PCA can also be used to reduce the *number of frame-level features*, which speeds up the calculation step of the utterance-level features for both the Bag-of-Audio-Words and the Fisher vectors approaches. (Of course, the amount of the actual speed-up is also affected by the optimal $N$ value for both methods.) Fewer (utterance-level) features also means that our machine learning model (in this case, SVR) can be evaluated more quickly. To reflect this speed-up, we expressed this relative execution time both in Tables 6 and 7; in each case, 100% means not using PCA at all.

Examining these values, it is clear that by utilizing Principal Component Analysis, it is possible to achieve notable speed-ups along with only slight drops in the conflict intensity estimation performance. For example, using the ComParE frame-level feature set with standardization along with PCA and retaining 99% of the information, we end up with 92 frame-level attributes instead of the original 130, leading to a speed-up of cca. 30% along with practically the same performance

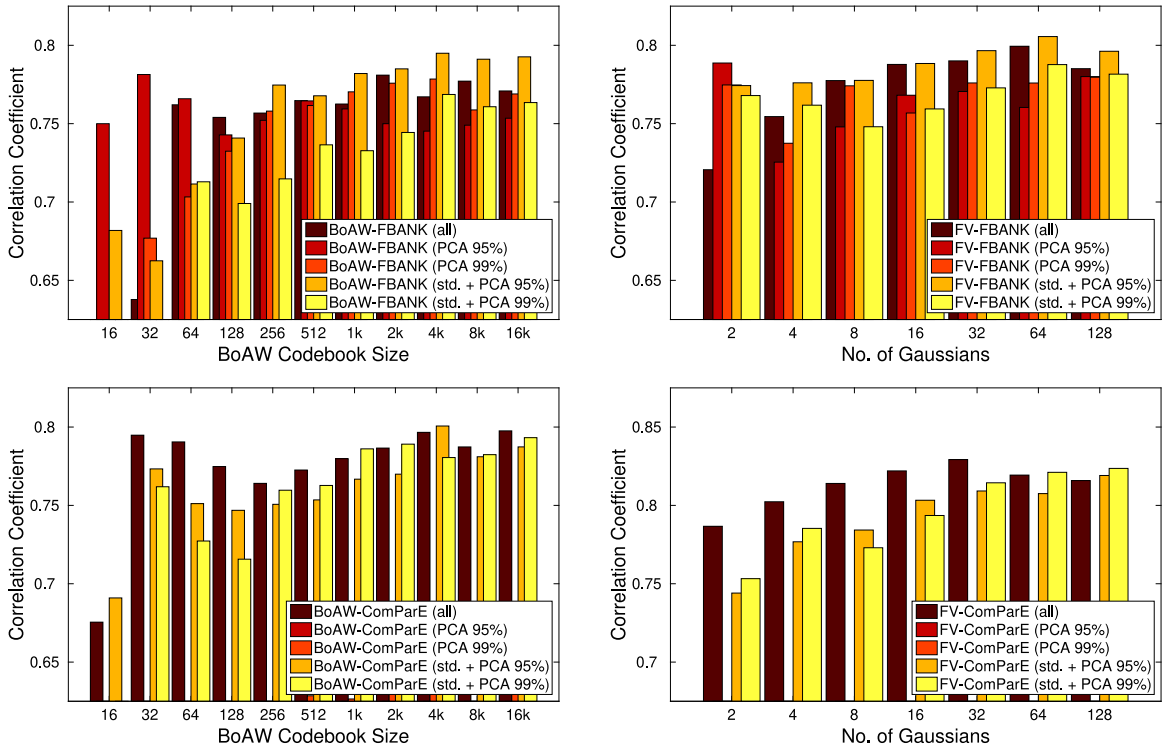


**Fig. 5.** Pearson's Correlation Coefficient (CC) values measured on the development set for the various frame-level feature sets tested for the Bag-of-Audio-Words (left) and the Fisher Vector (right) techniques.

**Table 6**
The performance scores obtained when using the Bag-of-Audio-Words representation on the frame-level attributes transformed by PCA.

| Frame-level features | PCA parameters | $N$ | Relative Exec. times | Correlation | | UAR | |
|---|---|---|---|---|---|---|---|
| | | | | Dev. | Test | Dev. | Test |
| FBANK + $\Delta$ + $\Delta\Delta$ | No PCA | 2048 | 100.0% | 0.781 | **0.845** | 77.4% | **82.3%** |
| | PCA (95%) | 32 | 0.3% | 0.781 | 0.741 | **78.5%** | 77.3% |
| | PCA (99%) | 4096 | 73.2% | 0.778 | 0.795 | 75.2% | 80.7% |
| | std. + PCA (95%) | 4096 | 89.4% | **0.795** | 0.791 | 54.4% | 66.7% |
| | std. + PCA (99%) | 4096 | 148.0% | 0.769 | 0.766 | 74.3% | 78.6% |
| ComParE + $\Delta$ | No PCA | 16384 | 100.0% | **0.798** | **0.835** | 77.4% | 81.5% |
| | PCA (95%) | 16384 | 0.8% | 0.617 | 0.515 | 55.3% | 68.8% |
| | PCA (99%) | 16384 | 1.5% | 0.631 | 0.625 | 75.1% | 72.0% |
| | std. + PCA (95%) | 4096 | 11.2% | **0.801** | 0.815 | **79.5%** | 81.0% |
| | std. + PCA (99%) | 16384 | 70.8% | **0.793** | **0.834** | **79.8%** | **82.5%** |

**Table 7**
The performance scores obtained when using the Fisher Vector representation on the frame-level attributes transformed by PCA.

| Frame-level features | PCA parameters | $N$ | Relative Exec. times | Correlation | | UAR | |
|---|---|---|---|---|---|---|---|
| | | | | Dev. | Test | Dev. | Test |
| FBANK + $\Delta$ + $\Delta\Delta$ | No PCA | 64 | 100.0% | **0.799** | **0.850** | 75.4% | **86.1%** |
| | PCA (95%) | 128 | 21.9% | 0.780 | 0.787 | **80.3%** | 79.8% |
| | PCA (99%) | 128 | 46.9% | 0.780 | 0.830 | 77.9% | 80.5% |
| | std. + PCA (95%) | 64 | 28.6% | **0.806** | 0.829 | 77.4% | 82.0% |
| | std. + PCA (99%) | 64 | 47.4% | 0.788 | 0.819 | 76.1% | 81.8% |
| ComParE + $\Delta$ | No PCA | 32 | 100.0% | **0.829** | **0.856** | **82.7%** | **84.5%** |
| | PCA (95%) | 16 | 0.2% | 0.590 | 0.516 | 50.0% | 50.0% |
| | PCA (99%) | 4 | 0.1% | 0.659 | 0.588 | 78.4% | 73.2% |
| | std. + PCA (95%) | 128 | 89.2% | **0.819** | **0.845** | 79.7% | **85.1%** |
| | std. + PCA (99%) | 128 | 141.5% | **0.824** | **0.840** | 78.9% | 83.8% |

(actually, the UAR scores improved by 1% absolute on the test set). For the Fisher vectors, this case actually led to higher execution times due to a larger optimal $N$ value (128 vs. 32), but retaining only 95% of the information was eligible for the same performance with a 10% reduction in the execution times (and again with a slight improvement in the UAR value on the test set). Therefore, although we were unable to improve prediction quality by applying PCA, we could efficiently calculate a more compact attribute set, therefore reduce the execution times of the feature extraction step while the retaining the original level of regression performance.

**Table 8**
The performance of several combined methods.

| Feature sets | Correlation | | UAR | |
|---|---|---|---|---|
| | Dev. | Test | Dev. | Test |
| ComParE functionals | 0.820 | 0.835 | 79.7% | 84.8% |
| ComParE functionals + BoAW (MFCC+Δ) | 0.855 | 0.839 | 80.1% | 81.5% |
| ComParE functionals + BoAW (FBANK+Δ) | 0.824 | 0.846 | 80.1% | 85.5% |
| ComParE functionals + BoAW (ComParE+Δ) | 0.824 | 0.850 | 80.5% | 84.7% |
| BoAW-MFCC + BoAW-FBANK + BoAW-ComParE | 0.852 | 0.833 | 80.6% | 81.2% |
| ComParE func. + BoAW-MFCC + BoAW-FBANK + BoAW-ComParE | 0.855 | 0.839 | 80.1% | 81.5% |
| ComParE functionals + FV (MFCC+Δ) | 0.849 | 0.855 | 75.4% | 83.2% |
| ComParE functionals + FV (FBANK+Δ) | 0.828 | 0.857 | 79.6% | 85.5% |
| ComParE functionals + FV (ComParE+Δ) | 0.836 | 0.855 | 81.8% | 85.3% |
| FV-MFCC + FV-FBANK + FV-ComParE | 0.850 | 0.860 | 75.5% | 84.4% |
| ComParE functionals + FV-MFCC + FV-FBANK + FV-ComParE | 0.852 | 0.860 | 76.7% | 84.1% |
| ComParE functionals + BoAW-MFCC + FV-MFCC | 0.858 | 0.849 | 79.9% | 81.8% |
| ComParE functionals + BoAW-FBANK + FV-FBANK | 0.828 | 0.856 | 79.6% | 85.6% |
| ComParE functionals + BoAW-ComParE + FV-ComParE | 0.836 | 0.855 | 81.8% | 85.3% |

## 9. Combining the predictions

Based on our prior experience, it might be beneficial to combine the predictions obtained by using different classification methods and/or feature sets (Gosztolya, 2015, 2019a, 2019b). To do this, a simple-yet-efficient method is *late fusion*, when we train separate machine learning methods for the individual feature sets, and perform this fusion by taking the weighted mean of the estimates. For classification, by 'estimates' we usually mean the posterior scores of the classes, while for regression we can simply take the output scores. The combination weights were determined on the development set in steps of 0.05.

Table 8 shows the scores obtained by combining the different approaches. After examining the results of our previous tests (see Tables 3 and 4), we decided to use the predictions obtained via using the original frame-level attributes and their first-order derivatives; for the BoAW approach this also meant building separate codebooks for the Δ values. Checking the results for Bag-of-Audio-Words, we can see that the combination turned out to be successful in two cases out of three: the CC scores for the test set rose from 0.822 to 0.839 and from 0.835 to 0.846, MFCC and FBANK feature sets, respectively. For the ComParE frame-level attributes, however, the predictions remained practically unchanged (0.851 to 0.850). When fusing the three BoAW-based predictions (either with or without the ComParE functionals case), we can see at most a slight improvement on the development set, but none on the test set. The reason for it is obviously the high performance of the MFCCs on the development set and the corresponding 0.822 score on test: since fusion weights were set based on the development set scores, this BoAW-MFCC model was considered to be quite important, which was not justified by test set performance. Actually, when we fused all four models in question (i.e. all three BoAW and the ComParE functionals), the results were identical to the 'ComParE functionals + BoAW-MFCC' case.

Turning to the Fisher vectors representation, it is clear that we got somewhat higher scores than in the BoAW cases: fusing the ComParE functionals approach with one of the FV models, we measured 0.855–0.857 CC scores on the test set. Combining the three FV models led to a CC value of 0.860, while adding the ComParE functionals predictions to the mix did not change this performance. Overall, these CC values support our previous conclusion that Fisher vectors seem to be a more descriptive utterance representation approach than the Bag-of-Audio-Words approach, and the extracted attribute set turns out to be more compact as well.

Lastly, we combined the ComParE functionals predictions with the estimates obtained by both the BoAW and the FV representations for each frame-level feature set tested (see the last block of Table 8). We found that, although the CC values on the development set appeared to be somewhat different, on the test set they were quite similar (i.e. in the range 0.849–0.856). Interestingly, the UAR scores also appeared to be around 85%, with the exception of the MFCC case, where we got a significantly lower value (i.e. 81.8%). Of course, meta-parameter setting and model selection was all done by focusing on the CC metric, so we cannot really expect a huge improvement in the UAR values anyway.

To sum up, combining the estimates produced by the different models led to slight improvements, as the CC scores rose from 0.816–0.856 to 0.839–0.860. Still, no matter which models we fused, in most cases we got CC values around 0.855–0.860. Examining the scores reported in previous studies for the SSPNet Conflict corpus (see Table 2), we can see several similar values (e.g. 0.849 Huang et al., 2014, 0.856 Gosztolya & Tóth, 2017 or 0.853 Rajan et al., 2019). Still, these scores were achieved by quite different state-of-the-art techniques, including ensemble learning, feature selection, classifier combination, Convolutional Neural Networks, and of course Bag-of-Audio-Words and Fisher vectors. In our opinion this probably indicates that scores presented in the current studies are already close to the glass ceiling for this task, i.e. we are near the highest possible score achievable.

To justify this opinion, we ask the reader to recall that for this corpus (and also for conflict intensity estimation in general) the task is to estimate the *annotated* conflict intensity score of each clip. Of course, manual annotation in this task is quite prone to human subjectivity, therefore label noise is very likely to be present in the target scores, and this will inevitably lead to a performance limit for any statistical approach applied. In our opinion, current studies have already attained the highest correlation coefficients achievable, and they match human performance.

## 10. Conclusions and discussion

In this study we focused on the task of automatically estimating conflict intensity from short audio clips. For this, for the first time in the scientific community, we utilized two recent, state-of-the-art feature extraction techniques: Bag-of-Audio-Words (BoAW) and Fisher vectors (FV). Since both these techniques construct segment-level feature vectors based on the frame-level attribute vectors of the recordings, we tested three typical standard frame-level attribute sets: MFCCs, filter banks and the ComParE feature set developed by Schuller et al. (2013). We also experimented with applying PCA first on the frame-level feature vectors to remove redundancy and project them into a quasi-orthogonal space; lastly, we also combined the different approaches tested to improve the conflict intensity estimation performance.

Our results indicate that all the approaches tested yielded competitive performance scores. Overall, Fisher vectors led to more accurate and more robust predictions than BoAW did. Among the frame-level

feature sets tested, we found MFCCs to be the most unreliable one, while the ComParE set led to the best performance on the test set; this was true for both BoAW and FV. The relatively low performance of MFCCs was especially surprising in the Fisher vector case, since it employs GMMs to model the distribution of the feature vectors, for which MFCCs are supposed to be an ideal choice. This, and the low performance of the models employing PCA, in our view indicates that FVs are quite robust regarding the input features. Despite the common expectations of GMMs, we find that for the input frame-level attributes being quasi-orthogonal is clearly secondary compared to providing an informative representation of the recordings. Although both MFCCs (which are calculated from the filter banks by applying a Discrete Cosine Transform (DCT, Nasir et al., 1974)) and the application of PCA transforms the original feature vectors into a quasi-orthogonal space, it seems that actual SVR performance is harmed more by losing a significant amount of relevant information during this process.

In the last part of our study, we experimented with combining the predictions obtained by the different regression models. Although we anticipated that this fusion would improve regression performance, we found that the correlation scores rose only by a very slight amount. Comparing the CC values of 0.850–0.860 with the state-of-the-art scores found in the recent literature (in the range 0.849–0.856), we hypothesize that current computational methods have already attained the best performance achievable by statistical approaches. Indeed, the conflict scores are inherently subjective to a certain degree, as they reflect the annotator's *opinion* about conflict intensity. Putting different cultural standards aside (since for this particular corpus all the annotators were from North America), the annotated scores are still influenced by the annotator's judgment, personality and even his current mood, which, from a machine learning viewpoint, means that the target scores are noisy to some extent. While this label noise can be reduced by taking the mean of several annotator's scores (as it was done by the creators of the SSPNet Conflict Corpus), it is inevitably still present in the intensity values of each clip; and since it is just noise, it per def cannot be precisely estimated by statistical methods. Of course, our score of 0.860 presented is actually the highest correlation value ever reported in a scientific study for this particular corpus, but it represents only a marginal improvement over the previous metric values reported. And for the reasons explained above, we do not really expect that it will be significantly outmatched in the future.

## CRediT authorship contribution statement

**Gábor Gosztolya:** Conceptualization, Methodology, Investigation, Writing, Software, Visualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This study was supported by the National Research, Development and Innovation Office of Hungary via contract NKFIH FK-124413. This research was also supported by the grant TKP2021-NVA-09 of the Hungarian Ministry for Innovation and Technology, and within the framework of the Artificial Intelligence National Laboratory Program (MILAB). Gábor Gosztolya was also funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences and by the Hungarian Ministry of Innovation and Technology New National Excellence Program ÚNKP-21-5-SZTE.

## References

Bourlard, H., & Morgan, N. (1994). *Connectionist speech recognition – A hybrid approach*. Kluwer Academic.

Brueckner, R., & Schuller, B. (2015). Be at odds? Deep and hierarchical neural networks for classification and regression of conflict in speech. In *Conflict and multimodal communication* (pp. 403–429). Springer International Publishing.

Caraty, M.-J., & Montacié, C. (2015). Detecting speech interruptions for automatic conflict detection. In *Conflict and multimodal communication* (pp. 377–401). Springer International Publishing, (Chapter 18).

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*, 1–27.

Chen, S., Li, X., Jin, Q., Zhang, S., & Qin, Y. (2016). Video emotion recognition in the wild based on fusion of multimodal features. In *Proceedings of ACM international conference on multimodal interaction* (pp. 494–500).

Cooper, V. W. (1986). Participant and observer attribution of affect in interpersonal conflict: an examination of noncontent verbal behavior. *Journal of Nonverbal Behavior*, *10*, 134–144.

Egas López, J. V., Orozco-Arroyave, J. R., & Gosztolya, G. (2019). Assessing parkinson's disease from speech by using Fisher vectors. In *Proceedings of interspeech* (pp. 3063–3067). Graz, Austria.

Eyben, F., Wöllmer, M., & Schuller, B. (2010). Opensmile: The Munich versatile and fast open-source audio feature extractor. In *Proceedings of ACM multimedia* (pp. 1459–1462).

Ferguson, N. (1977). Simultaneous speech, interruptions and dominance. *British Journal of Social and Clinical Psychology*, *16*, 295–302.

Georgakis, C., Panagakis, Y., Zafeiriou, S., & Pantic, M. (2017). The conflict escalation resolution (CONFER) database. *Image and Vision Computing*, *65*(1), 37–48.

Gosztolya, G. (2015). Conflict intensity estimation from speech using greedy forward-backward feature selection. In *Proceedings of interspeech* (pp. 1339–1343). Dresden, Germany.

Gosztolya, G. (2019). Posterior-thresholding feature extraction for paralinguistic speech classification. *Knowledge-Based Systems*, *186*, Article 104943.

Gosztolya, G. (2019). Using Fisher vector and bag-of-audio-words representations to identify styrian dialects, sleepiness, baby & orca sounds. In *Proceedings of interspeech* (pp. 2413–2417). Graz, Austria.

Gosztolya, G. (2019). Using the bag-of-audio-word feature representation of ASR DNN posteriors for paralinguistic classification. In *Proceedings of interspeech* (pp. 2413–2417). Graz, Austria.

Gosztolya, G., Busa-Fekete, R., Grósz, T., & Tóth, L. (2017). DNN-based feature extraction and classifier combination for child-directed speech, cold and snoring identification. In *Proceedings of interspeech* (pp. 3522–3526). Stockholm, Sweden.

Gosztolya, G., Grósz, T., Busa-Fekete, R., & Tóth, L. (2014). Detecting the intensity of cognitive and physical load using AdaBoost and deep rectifier neural networks. In *Proceedings of interspeech* (pp. 452–456). Singapore.

Gosztolya, G., Grósz, T., Szaszák, G., & Tóth, L. (2016). Estimating the sincerity of apologies in speech by DNN rank learning and prosodic analysis. In *Proceedings of interspeech* (pp. 2026–2030). San Francisco, CA, USA.

Gosztolya, G., & Tóth, L. (2017). DNN-based feature extraction for conflict intensity estimation from speech. *IEEE Signal Processing Letters*, *24*(12), 1837–1841.

Grèzes, F., Richards, J., & Rosenberg, A. (2013). Let me finish: Automatic conflict detection using speaker overlap. In *Proceedings of interspeech* (pp. 200–204).

Grósz, T., Busa-Fekete, R., Gosztolya, G., & Tóth, L. (2015). Assessing the degree of nativeness and Parkinson's condition using Gaussian processes and deep rectifier neural networks. In *Proceedings of Interspeech* (pp. 1339–1343). Dresden, Germany.

Grzybowska, J., & Kacprzak, S. (2016). Speaker age classification and regression using i-vectors. In *Proceedings of interspeech* (pp. 1402–1406). San Francisco, CA, USA.

Hocker, J. L., & Wilmot, W. W. (1995). *Interpersonal conflict*. Brown & Benchmark.

Huang, D.-Y., Li, H., & Dong, M. (2014). Ensemble Nyström method for predicting conflict level from speech. In *Proceedings of APSIPA* (pp. 2418–2422). Siem Reap, City of Angkor Wat, Cambodia.

Huckvale, M., & Beke, A. (2017). It sounds like you have a cold! testing voice features for the interspeech 2017 computational paralinguistics cold challenge. In *Proceedings of interspeech* (pp. 3447–3451). Stockholm, Sweden.

Jaakkola, T. S., & Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *Proceedings of NIPS* (pp. 487–493). Denver, CO, USA.

Jain, V., Crowley, J. L., Dey, A., & Lux, A. (2014). Depression estimation using audiovisual features and Fisher vector encoding. In *Proceedings of ACM multimedia* (pp. 87–91).

Jolliffe, I. (1986). *Principal component analysis*. Springer-Verlag.

Kaya, H., & Karpov, A. A. (2016). Fusing acoustic feature representations for computational paralinguistics tasks. In *Proceedings of interspeech* (pp. 2046–2050). San Francisco, CA, USA.

Kaya, H., Karpov, A. A., & Salah, A. A. (2015). Fisher vectors with cascaded normalization for paralinguistic analysis. In *Proceedings of interspeech* (pp. 909–913).

Kaya, H., Özkaptan, T., Salah, A. A., & Gürgen, F. (2015). Random discriminative projection based feature selection with application to conflict recognition. *IEEE Signal Processing Letters*, *22*(6), 671–675.

Kaya, H., & Salah, A. A. (2014). Combining modality-specific extreme learning machines for emotion recognition in the wild. In *Proceedings of ICMI* (pp. 487–493). Istanbul, Turkey.

Kim, S., Valente, F., Filippone, M., & Vinciarelli, A. (2014). Predicting continuous conflict perception with Bayesian Gaussian processes. *IEEE Transactions on Affective Computing*, *5*(2), 187–200.

Kim, S., Valente, F., & Vinciarelli, A. (2012). Automatic detection of conflicts in spoken conversations: Ratings and analysis of broadcast political debates. In *Proceedings of ICASSP* (pp. 5089–5092). Kyoto, Japan.

Lim, H., Kim, M. J., & Kim, H. (2015). Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation. In *Proceedings of Interspeech* (pp. 3325–3329). Dresden, Germany.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Moreno, P. J., & Rifkin, R. (2010). Using the Fisher kernel method for web audio classification. In *Proceedings of ICASSP* (pp. 2417–2420). Dallas, TX, USA.

Nasir, A., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, *100*(1), 90–93.

Panagakis, Y., Zafeiriou, S., & Pantic, M. (2014). Audiovisual conflict detection in political debates. In *Proceedings of ECCV* (pp. 304–314). Zurich, Switzerland.

Pancoast, S., & Akbacak, M. (2012). Bag-of-audio-words approach for multimedia event classification. In *Proceedings of interspeech* (pp. 2105–2108). Portland, OR, USA.

Pancoast, S., & Akbacak, M. (2014). Softening quantization in bag-of-audio-words. In *Proceedings of ICASSP* (pp. 1370–1374). Florence, Italy.

Pokorny, F. B., Graf, F., Pernkopf, F., & Schuller, B. W. (2015). Detection of negative emotions in speech signals using bags-of-audio-words. In *Proceedings of ACII* (pp. 1–5).

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Prentice Hall.

Rajan, V., Brutti, A., & Cavallaro, A. (2019). ConflictNET: ENd-to-end learning for speech-based conflict intensity estimation. *IEEE Signal Processing Letters*, *26*, 1668–1672.

Räsänen, O., & Pohjalainen, J. (2013). Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. In *Proceedings of interspeech* (pp. 210–214). Lyon, France.

Rawat, S., Schulam, P. F., Burger, S., Ding, D., Wang, Y., & Metze, F. (2013). Robust audio-codebooks for large-scale event detection in consumer videos. In *Proceedings of interspeech* (pp. 2929–2933). Lyon, France.

Rubin, J. Z., Pruitt, D. G., & Kim, S. H. (1994). *Social conflict: Escalation, stalemate, and settlement*. McGraw-Hill Book Company.

Schmitt, M., Janott, C., Pandit, V., Qian, K., Heiser, C., Hemmert, W., & Schuller, B. (2016). A bag-of-audio-words approach for snore sounds' excitation localisation. In *Proceedings of speech communication* (pp. 89–96).

Schmitt, A., & Minker, W. (2012). Novel strategies for emotion recognition. In *Towards adaptive spoken dialog systems* (pp. 99–112). Springer International Publishing.

Schmitt, M., Ringeval, F., & Schuller, B. (2016). At the border of acoustics and linguistics: Bag-of-audio-words for the recognition of emotions in speech. In *Proceedings of Interspeech* (pp. 495–499). San Francisco, CA, USA.

Schmitt, M., & Schuller, B. (2017). openXBOW – Introducing the Passau open-source crossmodal Bag-of-Words toolkit. *Journal of Machine Learning Research*, *18*, 1–5.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471.

Schuller, B., & Batliner, A. (2013). *Computational paralinguistics: emotion, affect and personality in speech and language processing*. Wiley Publishing.

Schuller, B., Steidl, S., Batliner, A., Hantke, S., Bergelson, E., Krajewski, J., Janott, C., Amatuni, A., Casillas, M., Seidl, A., Soderstrom, M., Warlaumont, A. S., Hidalgo, G., Schnieder, S., Heiser, C., Hohenhorst, W., Herzog, M., Schmitt, M., Qian, K., .... Zafeiriou, S. (2017). The interspeech 2017 computational paralinguistics challenge: Addressee, cold & snoring. In *Proceedings of interspeech* (pp. 3442–3446).

Schuller, B., Steidl, S., Batliner, A., Hirschberg, J., Burgoon, J. K., Baird, A., Elkins, A., Zhang, Y., Coutinho, E., & Evanini, K. (2016). The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language. In *Proceedings of interspeech* (pp. 2001–2005). San Francisco, CA, USA.

Schuller, B. W., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Salamin, H., Polychroniou, A., Valente, F., & Kim, S. (2013). The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In *Proceedings of interspeech* (pp. 148–152). Lyon, France.

Segura, C., Balcells, D., Umbert, M., Arias, J., & Luque, J. (2016). Automatic speech feature learning for continuous prediction of customer satisfaction in contact center phone calls. In *Proceedings of IberSPEECH* (pp. 255–265). Lisbon, Portugal.

Spector, P., & Jex, S. (1998). Development of four self-report measures of job stressors and strain: interpersonal conflict at work scale, organizational constraints scale, quantitative workload inventory, and physical symptoms inventory. *Journal of Occupational Health Psychology*, *3*(4), 356–367.

Sztahó, D., Kiss, G., & Vicsi, K. (2015). Estimating the severity of Parkinson's disease from speech using linear regression and database partitioning. In *Proceedings of interspeech* (pp. 498–502). Dresden, Germany.

Tian, Y., He, L., yi Li, Z., lan Wu, W., Zhang, W.-Q., & Liu, J. (2014). Speaker verification using Fisher vector. In *Proceedings of ISCSLP* (pp. 419–422). Singapore, Singapore.

Vedaldi, A., & Fulkerson, B. (2010). Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of ACM multimedia* (pp. 1469–1472).

Vetráb, M., & Gosztolya, G. (2019). Érzelmek felismerése magyar nyelvű hangfelvételekből akusztikus szózsák jellemzőreprezentáció alkalmazásával. In *Proceedings of MSZNY* (pp. 265–274). Szeged, Hungary.

Wu, H., Wang, W., & Li, M. (2019). The DKU-LENOVO system for the INTERSPEECH 2019 computational paralinguistic challenge. In *Proceedings of interspeech* (pp. 2433–2437). Graz, Austria.

Young, S., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., & Woodland, P. (2006). *The HTK book*. Cambridge, UK: Cambridge University Engineering Department.

Zajíc, Z., & Hrúz, M. (2016). Fisher vectors in PLDA speaker verification system. In *Proceedings of ICSP* (pp. 1338–1341). Chengdu, China.