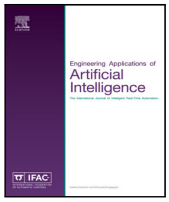
Engineering Applications of Artificial Intelligence 107 (2022) 104541

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Optimizing class priors to improve the detection of social signals in audio data

Gábor Gosztolya [a,b,*]

[a] University of Szeged, Institute of Informatics, Szeged, Hungary
[b] MTA-SZTE Research Group on Artificial Intelligence, ELRN, Szeged, Hungary

ARTICLE INFO

Keywords:
Audio processing
Social signals
Laughter detection
Filler events
Deep neural networks
a priori estimates
Optimization
CMA-ES

ABSTRACT

To detect social signals such as laughter and filler events in an audio recording, the most straightforward way is to utilize a Hidden Markov Model — or these days a Hidden Markov Model/Deep Neural Network (HMM/DNN) hybrid. HMM/DNNs, however, perform best if the DNN outputs are scaled by dividing them by the a priori class probabilities first, before applying a dynamic or Viterbi beam search. These class a priori probability values (or *priors* for short) are usually estimated by counting the frame occurrences of each class in the training set and then dividing these totals by the total number of frames. These estimates, however, may in fact be suboptimal for a number of reasons ranging from imprecise labeling to the overconfidence of DNNs. In this study we show empirically that more reliable scaling factors can be obtained by optimization. Using this approach, we managed to achieve a 6 – 9% relative error reduction both at the frame level and the segment level, using a public database containing spontaneous English mobile phone conversations.

1. Introduction

The classification and detection of different verbal cues such as laughter and filler events (i.e. sounds like “um”, “eh”, “er” etc.) has a relatively long history in speech technology. There were some attempts even in the early 1990s to identify speech excerpts as laughter or non-laughter (Wheatley et al., 1992), and many further such studies have been published since then (e.g. Kennedy and Ellis, 2004; Truong and van Leeuwen, 2007; Kantharaju et al., 2018). The automatic detection of filler events, however, has become popular only in the past decade (see e.g. Salamin et al., 2013; Gupta et al., 2013; Brueckner et al., 2017; Gosztolya et al., 2019, 2020; Baur et al., 2020).

A simpler approach of identifying social signals is to cut segments from spontaneous speech recordings and seek to distinguish these excerpts as either the vocalization(s) in question (e.g. laughter, filler events), or miscellaneous speech/silence (Truong and van Leeuwen, 2007; Neuberger and Beke, 2013). Many recent studies, however, performed evaluation only at the frame level (e.g. Gupta et al., 2013; Schuller et al., 2013; Gupta et al., 2016; Baur et al., 2020). A third approach, which in fact falls closer to real-life expectations, is to detect occurrences of the given phenomena within longer utterances, without relying on possible locations provided by human annotators. In this approach, we have to solve the *identification* and the *localization* tasks simultaneously. To do this, one may simply borrow techniques from the area of Automatic Speech Recognition (ASR); for example, to employ a Hidden Markov Model (HMM) in order to fuse the local (i.e. frame-level) likelihood estimates provided by a Gaussian Mixture Model (GMM) into segment-level occurrence hypotheses. More recently, as Deep Neural Networks (DNNs) were invented, HMM/GMMs have been replaced by the so-called HMM/DNN hybrid models (Mohamed et al., 2012) in ASR, and this technique can be straightforwardly applied in this task as well. Nowadays, recurrent neural architectures (applying units such as Long-short term memory (LSTM, Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRUs, Cho et al., 2014) as building blocks) have become state-of-the-art in ASR; nevertheless, there are several reasons for still employing the HMM/DNN model instead of applying a recurrent neural network. These include easier training, lower computational complexity and memory footprint; furthermore, non-recurrent neural networks were shown to have a competitive or even superior performance when the amount of training data is limited (Panzner and Cimiano, 2016; Schmitt et al., 2019), which is usually the case in social signal detection.

Before utilizing the DNN outputs in the Hidden Markov Model, first we should transform them using Bayes’ theorem. This in practice means that we have to divide them by the a priori probability values of the classes (or the *priors* for short), which are usually estimated via simple statistical methods. Unfortunately, in practice the estimates for both the a posteriori and a priori probabilities are likely to be imprecise. Artificial neural networks are known to underestimate the probability of rarer classes and overestimate the probability of the more common

* Correspondence to: MTA-SZTE Research Group on Artificial Intelligence, ELRN, Szeged, Hungary
E-mail address: ggabor@inf.u-szeged.hu.

https://doi.org/10.1016/j.engappai.2021.104541
Received 10 June 2021; Received in revised form 22 September 2021; Accepted 31 October 2021
Available online xxxx
0952-1976/

ones (Hand and Yu, 2001; Tóth et al., 2005; Buda et al., 2018). As regards the a priori probability estimates, one source of imprecision is the finite nature of the training set (Kar et al., 2016), and there could also be a distribution mismatch between the training set and the development set or the test set (Saerens et al., 2002). Hence, correcting the class a priori probability estimates might yield some improvement in the recognition performance, while it also provides us the opportunity to (linearly) correct a possible bias present in the posterior values.

The need to fine-tune class a priori probabilities appears in many scientific areas, mostly in those which focus on aggregated data instead of dealing with the individual cases separately. Besides statistical studies (King and Lu, 2008; Hopkins and King, 2010), it has been applied in many machine learning areas such as natural language processing (Chan and Ng, 2006), image processing (Buda et al., 2018) and data mining (Forman, 2008; Esuli and Sebastiani, 2015). Because of the heterogeneous nature of these areas, there are many equivalent terms for the task of adjusting the class priors, such as *quantification* (Kar et al., 2016), *class prior estimation* (Chan and Ng, 2006), *class probability re-estimation* (Alaíz-Rodríguez et al., 2011), *multi-class thresholding* (Buda et al., 2018; Johnson and Khoshgoftaar, 2019) and *learning of class balance* (du Plessis and Sugiyama, 2012).

Clearly, searching for social signal occurrences via a HMM/DNN hybrid model has several properties in common with these tasks; most importantly that estimating class distribution is not an end in itself, but it is rather used to improve the accuracy of higher-level tasks such as classification. In our case, DNNs are used primarily not to classify individual frames, but to estimate local likelihoods, which are then combined over the time axis by a Hidden Markov Model. Furthermore, there may be a mismatch between training and test sets, and, similarly to the set-up of Balikas et al. (2015), we can expect to have a separate development set that could be used to fine-tune the class prior estimates.

To address the above-mentioned imprecision of both the a posteriori and the a priori probability estimates, we will treat the class prior vectors as trainable parameters, and optimize them in order to improve the detection of laughter and filler events in audio recordings. Since in a HMM/DNN hybrid model we combine the frame-level posterior estimates to form utterance-level hypotheses, we will also show that it is worth incorporating the HMM search step in the optimization process. To our knowledge, this is the first study that has fine-tuned the class prior vector in speech technology, and also for Hidden Markov Models.

The structure of this paper is as follows. First, we examine the a priori probability estimation procedure when looking for the occurrences of laughter and filler events in audio recordings using a HMM/DNN hybrid. Next, we list the methods we applied for class prior optimization. Then we describe our experimental setup, namely the database used, the parameters used for training the deep neural network and the evaluation metrics applied. Next, we present and analyze our results, and also examine the different class prior values tested. Lastly, we draw our conclusions.

2. Hidden Markov models for audio processing

In audio processing the standard way of handling the audio signal is to divide it into the so-called *frames*, which are equal-sized small chunks of usually 25 ms long with a 10 ms time step. (Therefore, there are 100 frames for each second of the audio.) A standard Hidden Markov Model expects the frame-level estimates of the class-conditional likelihood $p(x_t|c_k)$ values for the given (and in this case, frame-level) observation vector x_t and for each class c_k as its input. From these, it calculates the most probable state sequence; that is, for each frame it supplies the most probable class c_k, taking account *the whole utterance* (i.e. frame sequence). In traditional ASR, this sequence is used to obtain the (word-level) transcript of the speech utterance with its time alignment (i.e. for all words also its starting and ending time points within the utterance are provided). In other tasks, for example when detecting social signals in speech recordings, we can calculate the starting and ending points of the social signals uttered.

The frame-level $p(x_t|c_k)$ estimate values had been usually supplied by Gaussian Mixture Models. In the case of HMM/DNN hybrid models, however, we replace the GMMs with Deep Neural Networks. Unlike GMMs, which are generative methods, DNNs are discriminative classifiers and as such, they are known to estimate $P(c_k|x_t)$. The $p(x_t|c_k)$ values expected by the HMM can be obtained by employing Bayes' theorem as

$$p(x_t|c_k) = \frac{P(c_k|x_t) \cdot P(x_t)}{P(c_k)}. \quad (1)$$

Therefore, in a HMM/DNN hybrid model, the posterior estimates provided by the DNN component are to be divided by the $P(c_k)$ a priori probabilities of the phonetic classes. This will supply us with the required likelihood estimates within a scaling factor (the combined probability of the x_t observation vectors); luckily, as this scaling factor does not influence the subsequent search process, it can be ignored.

To build a HMM/DNN system on an audio database and reliably measure its performance, the given dataset has to be split into three different parts: to training, development and test sets. The training set is utilized to train the frame-level DNN acoustic model; this requires the presence of the frame-level class labels for the corresponding utterances. By evaluating this neural network on the utterances of the development and test sets, we obtain the above-mentioned (frame-level) $P(c_k|x_t)$ posterior estimates. Additionally, the $P(c_k)$ class prior estimates are usually also calculated based on statistics of the training set: they are typically calculated from the frequency of the frame-level class labels. Next, the development set is used to tune the hyperparameters of the Hidden Markov model such as a state insertion penalty, the weight of the language model, or, in our case, the optimized class prior estimate values. This step, of course, relies on the posterior estimates of the already trained DNN model. Lastly, the performance of the whole HMM/DNN hybrid model is measured on the test set; at this point, no further parameter adjustment is allowed.

3. A priori probability optimization

Unfortunately, in practice the above-referenced probability estimates are not precise, since many factors might affect the probability estimation process and reduce the quality of values. The $P(c_k|x_t)$ a posteriori values are primarily affected by a bias of DNNs towards the classes having more training examples, and by the limited sizes of training sets. As regards the $P(c_k)$ a priori scores, they are typically determined by counting the ratio of frames belonging to each class or state in the training set. These frame-level class labels used to come from a manual annotation of the training speech corpora, but this procedure was superseded by the application of an automated forced-aligned process. Still, regardless of the source of the frame-level class labels, they are prone to noise due to the imprecise positioning of phonetic boundaries (or, in our case, the occurrences of the social signals), and this noise is obviously propagated further to the $p(x_t|c_k)$ values.

Furthermore, the datasets available for speech recognition purposes typically take up dozens of hours (nowadays hundreds of hours is fairly common); yet, in the case of laughter detection, the available annotated materials are usually significantly smaller, taking up to only about ten or twenty hours or even less (see e.g. Neuberger et al., 2014). Since the a priori probability estimates are determined statistically on the training set, it is easy to see that the smaller the training set, the less reliable the estimated $P(c_k)$ scores will be.

Owing to these factors, both kinds of probability estimates are likely to be sub-optimal; adjusting the a priori estimates might lead to a significant improvement in the recognition accuracy values. More

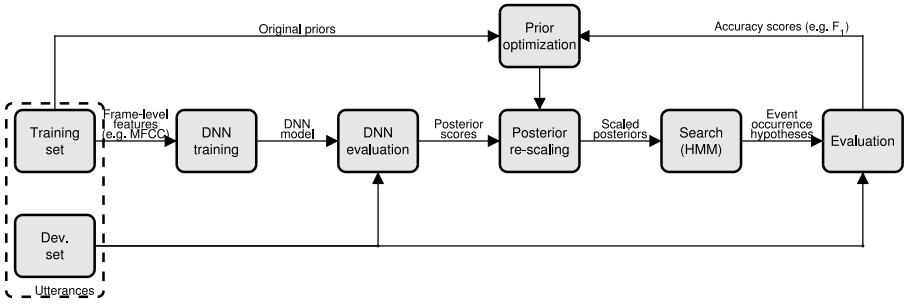


Fig. 1. The general workflow of the prior optimization process.

Table 1
The distribution of laughter and filler events in the SSPNet Vocalization Corpus.

Set	Total count			Total duration (min:sec)			Total duration (%)	
	Utterances	Laughter	Fillers	Utterances	Laughter	Fillers	Laughter	Fillers
Training	1583	649	1710	289:18	9:53	14:10	3.4%	4.9%
Development	500	225	556	91:18	4:18	4:54	4.7%	5.4%
Test	680	284	722	124:04	4:00	5:55	3.2%	4.8%
Total	2763	1158	2988	504:40	18:11	24:59	3.6%	5.0%

formally, in this study we will treat the a priori estimates as a hyper-parameter vector $P'(\mathbf{c})$ (with two straightforward formal requirements that $P'(c_k) \geq 0$ and $\sum_{k=1}^{K} P'(c_k) = 1$), and use the values

$$p'(x_t|c_k) = \frac{P(c_k|x_t)}{P'(c_k)} \tag{2}$$

as the input of the HMM instead of the ones derived following Bayes' theorem (i.e. Eq. (1)). Notice that by adjusting the $P'(\mathbf{c})$ prior estimates, we can also have the opportunity to correct a (linear) bias in the posterior estimates as well.

To determine the optimal $P'(\mathbf{c})$ vector, the studies listed above (e.g. Balikas et al., 2015; Kar et al., 2016) sought optimal classification on the development set. Translating this to our task means optimal *frame-level classification*. Now, however, we can go one step further. As the example-wise (i.e. frame-wise) likelihood estimates are combined to form event occurrence hypotheses by a Hidden Markov Model, it is straightforward to look for the prior vector that leads to the highest quality *occurrences*. This means that our optimized function should include this HMM search step as well. (For the general workflow of the proposed approach, see Fig. 1.)

3.1. Optimization

Notice that the choice of the actual optimization algorithm was not considered as an inherent part of the proposed workflow: in the prior optimization step described above we might utilize practically any method. When we have only two classes (the event we aim to locate, and everything else), even a simple grid search is sufficient, but with more classes, we might want to opt for some more sophisticated methods. In the experimental validation of our workflow proposed, we tested two such approaches, which are different by nature.

The first method we employed is the quite simple approach of **generating random values** and choosing the prior vector which leads to optimal social signal detection performance on the development set. Although at first glance this might seem to be a primitive technique, it was shown that this approach is actually more efficient for hyper-parameter optimization than grid search is. (The reader is kindly referred to the study of Bergstra and Bengio (2012).) In our experiments we generated random prior combinations, executed the search step with the Hidden Markov model on the development set, and picked the prior probability vector with the highest F_1 scores.

The other optimization method we employed was the **Covariance Matrix Adaptation Evolution Strategy** (CMA-ES, Hansen and Ostermeier, 2001). Evolution Strategies resemble Genetic Algorithms in that they mimic the evolution of biological populations by recombination and selection, therefore they can "evolve" efficient solutions for real-world tasks. CMA-ES is reported to be a competitive and reliable algorithm for badly conditioned, non-smooth (i.e. noisy) or non-continuous problems. It is viewed as a reliable and competitive method for both local and global optimization (Hansen and Kern, 2004). It has a further advantage that it requires little or no meta-parameter setting for optimal performance. This algorithm has been implemented in several programming languages such as Matlab, Java, C++, Octave and Python. Here we used the Java one with the default settings.

4. The SSPNet vocalization corpus

We used the SSPNet Vocalization Corpus (Salamin et al., 2013), consisting of 2763 short audio segments from British English spontaneous telephone conversations involving 120 speakers, containing 2988 laughter and 1158 filler events. The total duration of this dataset is 8 h and 25 min, which makes it one of the largest corpora used for social signal detection. In this corpus only 3.6% of the duration consists of laughter, and 5.0% corresponds to filler events, while the rest of the recordings (i.e. 91.4%) consists of miscellaneous speech (51.2%) and silence (40.2%). Unfortunately, in the publicly available annotation only the occurrences of the laughter and filler events are indicated, therefore we had to merge the "miscellaneous speech" and the "silence" categories, leaving us with three classes: "laughter", "filler" and "miscellaneous" (meaning both silence and non-filler non-laughter speech).

We used the standard division of the dataset into a training, a development and a test set, introduced at the Interspeech Computational Paralinguistics Challenge (ComParE) in 2013 (Schuller et al., 2013). For the key properties of this corpus and this division, the reader is referred to Table 1. Roughly 60% of the 2763 clips (1583) formed the training set, while 500 clips were assigned to the development set and 680 clips to the test set. We would like to add that, although such relatively large development and test sets are required to reliably measure the performance of any classification method applied, this means that the training set is less than five hours overall.

5. Experimental setup

5.1. DNN parameters

As the DNN component of our HMM/DNN hybrid recognizer, we applied a deep network, consisting of rectified linear units as hidden

neurons (Glorot et al., 2011; Tóth, 2013). Following the results of preliminary tests, we used DNNs having five hidden layers, each containing 256 neurons and applying the ReLU activation function. The output layer had three neurons (that is, the same as the number of our classes) with the softmax activation function. Since DNN training is known to be a stochastic procedure due to the random initialization of the weights, we trained five DNN models, and averaged out the computed metric scores. This was done both for the baseline models and in the experiments performed using the optimized class priors.

We used the frame-level feature set introduced in the ComParE 2013 Challenge (Schuller et al., 2013), which consists of the 39 MFCC + Δ + $\Delta\Delta$ components together with voicing probability, harmonic-to-noise ratio (HNR), fundamental frequency (F_0) and zero-crossing rate, and their first-order derivatives. These 47 attributes were extended with their means and standard derivatives in a 9-frame-long neighborhood, which led to a total of 141 frame-level features (Schuller et al., 2013). We used the openSMILE tool (Eyben et al., 2010) to extract these features. Again, following the results of preliminary tests, we trained our neural networks on a 33 frame-wide sliding window by utilizing the feature vectors of the 16 neighboring frames from both sides.

5.2. HMM state transition probabilities

Our Hidden Markov model consisted of only three states, each one corresponding to a different acoustic event. In this scenario, the HMM state transition probabilities practically correspond to a (simple) language model. Following the study of Salamin et al. (2013), we built a frame-level (state) bi-gram language model calculated based on the training set. The probability values provided by this language model were fused with the acoustic likelihoods (i.e. the DNN outputs divided by the actual class priors) via weighted sum, where the optimal language model weight was determined on the development set.

5.3. Standard approaches

Although the traditional approach of social signal detection by HMM/DNNs is to divide the DNN outputs (i.e. the posterior estimates) by the class priors, class imbalance can also be handled during the DNN training step. To provide a wider comparison, we also test two such sampling strategies: *downsampling* and *uniform sampling*. In the downsampling approach, we discard examples from the majority class(es) (Gupta et al., 2013; Buda et al., 2018; Gosztolya et al., 2020); in our experiments, we realized this strategy by randomly discarding samples from the far most frequent "miscellaneous" class to make its frequency match that of filler frames. Although downsampling treats the class imbalance issue efficiently, it also evidently reduces the variance of the training examples. To this end, we decided to test the approach of "uniform sampling" as well, where we select the same number of training samples for each class *within each DNN training epoch*. The main difference compared to downsampling is that here we do not discard training examples at all, but samples belonging to the more frequent classes are not used in each training epoch, while the instances of minority classes might be used multiple times. Since in both cases, the frequency of the training samples of all classes are similar, there was no need to divide the DNN outputs by any sort of class prior estimate.

5.4. Evaluation metrics

In the social signal detection task, where the distribution of the classes is far from uniform, traditional classification accuracy has only a limited reliability. A straightforward choice might be the Area-Under-the-Curve (AUC) score of the frame-level posterior estimates of the more important classes of interest (in our case laughter and filler events) instead, which is indeed employed by several researchers in the social signal detection task (see e.g. Gupta et al., 2013; Schuller et al., 2013; Brueckner and Schuller, 2014). However, because the reliability of frame-level AUC in this task has been questioned both theoretically and experimentally (for the details, see Gosztolya, 2015), in our opinion performance can be more reliably judged by measuring the quality of event occurrence hypotheses at the utterance level (i.e. after employing the Hidden Markov model).

Due to this, we first used a HMM to perform event occurrence detection, and calculated our accuracy metrics based on these detected occurrences. To decide whether the event occurrence hypothesis returned by the HMM actually corresponds to a specific one marked by a human annotator, we combined two approaches, requiring that they both be fulfilled at the same time. In the first one (see e.g. Gosztolya, 2015; Pokorny et al., 2016), we expect that they refer to the same event type (in our case both has to be laughter, or both has to be filler) and also that their time intervals intersect. The second one was inspired by the NIST standard for Spoken Term Detection evaluation (NIST, 2006); following this, the center of the two occurrences has to be close (within 500 ms) to each other.

We applied the information retrieval metrics of *precision*, *recall* and their harmonic mean, *F-measure* (or F_1), which we regard as appropriate and straightforward metrics for the current task. Since we have two social signals, and these metrics are calculated for both of them, we have to summarize them in some way; for this, we opted for *macro-averaging*, meaning that we averaged the precision and recall scores of the two phenomena, and calculated the F_1 value from these averages. As it is also common to calculate these metrics at the frame level, we will measure the effectiveness of the prior optimization techniques by using both values. We optimized the class prior vectors independently for the two (evaluation) approaches used. Of course, due to the difference in the evaluation process, the F_1 values cannot be compared to the AUC scores reported in the previous works (e.g. in Gupta et al., 2013; Brueckner and Schuller, 2013; Gupta et al., 2016).

5.5. Class prior optimization

Because we optimized the prior values of the three classes, we had a three-dimensional optimization task. As explained in Section 3, we optimized the prior vector on the development set, while the test set was used for final model evaluation. Before performing a search via the HMM, we normalized the prior probability estimates supplied by both optimization methods to sum up to one. In the random optimization process, we generated 1000 prior vectors overall, following a uniform distribution. Regarding the CMA-ES method, it permits setting the initial vector of the search process; we used the original class priors (i.e. those calculated on the training set) for this purpose. The other parameters of CMA-ES were kept on their default setting (i.e. initial standard deviation was 0.2, initial population size was 25, while function tolerance (used in the termination criteria) was 10^{-9}). We tuned the language model weight for each class prior probability estimate vector on the development set.

For reference, we also tested the method of choosing the a priori estimate vector which leads to the best frame-level *classification* performance of the development set. That is, after re-scaling the frame-level DNN outputs by Bayes' theorem with the actual prior estimate, for each frame we select the class with the highest *transformed* likelihood; after this step, we simply take the traditional classification accuracy on these frame-level class hypotheses, where ground truth frame labels come from the manual annotation. This is an approach that is similar to those described in the literature (e.g. Kar et al., 2016; Balikas et al., 2015), where the prior estimates are tuned in order to achieve optimal *classification*.

Table 2
Optimal averaged F-measure values for the standard approaches tested.

Evaluation	DNN training sampling	Prior probability estimation approach	Development set			Test set		
			Laughter	Filler	Avg.	Laughter	Filler	Avg.
Segment-level	Full sampling	Statistical (training set)	69.9%	77.1%	73.5%	60.1%	66.7%	63.4%
	Downsampling	No/uniform	67.3%	76.7%	72.2%	59.8%	66.5%	63.4%
	Uniform sampling	No/uniform	62.2%	74.2%	69.0%	59.5%	64.8%	62.6%
Frame-level	Full sampling	Statistical (training set)	73.1%	70.6%	71.9%	61.4%	58.1%	59.8%
	Downsampling	No/uniform	72.2%	70.2%	71.4%	59.5%	58.1%	58.9%
	Uniform sampling	No/uniform	63.8%	66.0%	64.9%	58.0%	55.4%	56.7%

Table 3
Optimal **segment-level** averaged F-measure values when using different strategies for prior probability estimation.

Prior probability estimation approach		Development set			Test set		
		Laughter	Filler	Avg.	Laughter	Filler	Avg.
No/uniform class priors		64.4%	75.0%	69.7%	63.9%	65.3%	64.8%
Statistical	Training set (baseline)	69.9%	77.1%	73.5%	60.1%	66.7%	63.4%
	Training + dev. sets	69.9%	77.2%	73.6%	60.8%	**66.9%**	63.9%
	Training + dev. + test sets	69.8%	77.1%	73.5%	60.3%	**66.9%**	63.6%
	Test set (for reference only)	69.6%	76.9%	73.3%	59.4%	66.7%	63.1%
Opt. (Random)	Average	63.9%	74.5%	69.4%	63.6%	64.6%	64.4%
	Best	70.3%	77.0%	73.6%	65.1%	66.4%	65.8%
Opt. (CMA-ES)	Classification	65.9%	75.1%	70.6%	64.9%	64.9%	64.9%
	HMM (proposed)	**72.5%**	**77.3%**	**75.3%**	**65.8%**	66.4%	**66.6%**

6. Results

6.1. Standard approaches

Firstly, we investigated the standard approaches: besides full database sampling DNN training, we tested the performance of downsampling and uniform sampling. As described in Section 5.3, we utilized the standard, statistical method of counting the ratio of frames of each class in the training set to transform the DNN outputs in the full sampling case, while for downsampling and uniform sampling, there was no need for such a transformation.

Table 2 shows the F_1 scores averaged out for the five DNN models for the laughter and the filler events, and the corresponding macro-averaged F-measure values both on the development and on the test set. Surprisingly, among the three baseline approaches, full sampling led to the best scores, outperforming both downsampling and uniform sampling. Since we found that neither downsampling nor uniform sampling was able to balance the posterior estimates better than using full database sampling and dividing the DNN outputs by the standard class prior estimates, in the following we will use the latter approach as our baseline.

6.2. Prior probability estimation strategies

Table 3 shows the **segment-level** F_1 values we got for the laughter and filler events, averaged out for the five DNN models, and the corresponding macro-averaged F-measure scores for all approaches tested. We can see that, surprisingly, using the standard class prior vector (see the baseline score) assisted the detection on the development set, but on the test set the F_1 values for the laughter and the combined case actually fell slightly.

The next interesting observation is that calculating the prior probability estimates on the training set alone yields worse (segment-level) F_1 scores than when we utilize the development and test sets as well (although without the test set the F_1 values obtained are slightly better). This, in our view, indicates that the training set by itself is just too small (290 min, taking up only 57% of the dataset), and extending it with the 91-minute development set and with the 124-minute test set significantly improves the quality of the a priori probability estimates. The slight drop measured when adding the test set to the a priori estimate calculation is probably due to a mismatch between the development set and the test set of this particular database (see Table 1). Note that we included the test set in the statistical prior estimation process just for the sake of performance comparison; since our study focuses on fine-tuning the prior estimates, we do not regard this as peeking.

When we randomly generated the prior vectors, the average performance was no better than when we used no priors at all, which is not surprising. When choosing the vector which performed best, however, we see a great improvement for the laughter events; and although the F_1 value obtained for the filler events decreased slightly on both sets (from 77.1% to 77.0% and from 66.7% to 66.4%, development and test sets, respectively), it was countered by the improvements in the laughter events (i.e. from 69.9% to 70.3% and from 60.1% to 65.1%), leading to improvements in the average F_1 scores. This, in our opinion, means that it is indeed worth adjusting the class a priori prior estimates on the development set, and that generating random values is a viable way for such an optimization.

Using CMA-ES for *classification* optimization brought mixed results. Although the F_1 values were usually better than without using a priori probabilities, they were only slightly higher than the baseline scores. This accords with the fact that detecting laughter and filler events is not a frame-level classification task, while this approach focused just on frame-level classification. However, by incorporating the Hidden Markov Model into the optimization process we could outperform both the baseline scores and those obtained by the other approaches: the combined F_1 scores rose from 73.5% to 75.3%, and from 63.4% to 66.6%, development and test sets, respectively. This approach outperforms even the case of using the development and/or the test set as well (F_1 values of 63.9% and 63.6% on the test set).

Our findings regarding the **frame-level** F_1 values (see Table 4) basically mirror the segment-level tendencies. The main difference is perhaps that now using even the prior vector calculated in the traditional way (i.e. our baseline) led to better results than not using class priors at all. Using random prior values, on average, did not improve the performance, but choosing the vector which performed best on the development set improved the F-measure value from 59.8% to 62.0% on the test set. Seeking optimal (frame-level) classification actually made all the F_1 scores worse, but incorporating the HMM search step in the CMA-ES optimization process led to the highest combined F_1 value on the test set at the frame level as well: the 62.2% achieved means a relative error reduction score of 6% compared to our baseline.

Table 4
Optimal **frame-level** averaged F-measure values when using different strategies for prior probability estimation.

Prior probability estimation approach		Development set			Test set		
		Laughter	Filler	Avg.	Laughter	Filler	Avg.
No/uniform class priors		54.4%	67.4%	61.9%	53.8%	57.3%	56.5%
Statistical	Training set (baseline)	73.1%	70.6%	71.9%	61.4%	58.1%	59.8%
	Training + dev. sets	**73.5%**	70.6%	72.1%	62.4%	58.2%	60.3%
	Training + dev. + test sets	73.0%	70.9%	72.0%	61.6%	58.4%	60.0%
	Test set (for reference only)	72.7%	70.5%	71.7%	61.1%	58.0%	59.7%
Opt. (Random)	Average	54.2%	66.6%	61.4%	53.2%	56.7%	55.9%
	Best	71.6%	**72.3%**	72.2%	63.6%	**60.1%**	62.0%
Opt. (CMA-ES)	Classification	62.7%	67.8%	65.5%	59.5%	57.7%	58.8%
	HMM (proposed)	73.2%	71.8%	**72.9%**	**64.7%**	59.0%	**62.2%**

Table 5
Optimal segment-level and frame-level averaged precision, recall and F-measure values for some chosen prior estimation strategies on the test set for the DNN models trained with full database sampling.

Evaluation	Prior probability estimation approach	Laughter events			Filler events			Avg.
		Prec.	Rec.	F_1	Prec.	Rec.	F_1	F_1
Segment-level	Statistical (training set)	58.0%	62.4%	60.1%	66.3%	67.1%	66.7%	63.4%
	Opt. by random (HMM, best)	66.8%	63.6%	65.1%	65.2%	67.6%	66.4%	65.8%
	Opt. by CMA-ES (HMM) (proposed)	71.2%	61.4%	65.8%	60.7%	73.4%	66.4%	66.6%
Frame-level	Statistical (training set)	54.6%	70.4%	61.4%	54.3%	62.6%	58.1%	59.8%
	Opt. by random (HMM, best)	68.8%	59.3%	63.6%	58.1%	62.3%	60.1%	62.0%
	Opt. by CMA-ES (HMM) (proposed)	68.0%	61.8%	64.7%	53.3%	66.2%	59.0%	62.2%

Table 6
Significance levels ("p") of the F_1 improvements obtained by the different prior optimization strategies tested.

Level	Prior probability estimation approach	Significance (p)		
		Lau.	Fil.	Avg.
Segment	Best random	<0.01	–	<0.01
	CMA-ES (classif.)	<0.01	–	<0.01
	CMA-ES (HMM)	<0.01	–	<0.01
Frame	Best random	<0.01	<0.01	<0.01
	CMA-ES (classif.)	–	–	–
	CMA-ES (HMM)	<0.01	= 0.02	<0.01

Table 5 shows the precision and recall scores for the baseline and the best a priori probability estimation approaches for the test set. We can see that, for laughter events, the improvements for both optimization approaches came from the higher precision score: the segment-level baseline value of 58% rose to 66.8–71.2%, while recall remained around the baseline value of 62.4%. At the frame level the precision values rose even more, but there the recall scores fell significantly (i.e. from 70.4% to 59.3% and 61.8%).

In the case of filler events, the F_1 values remained roughly at the same level after both optimization approaches; we can see, however, that the precision and recall scores behave quite differently. When we optimized the prior vectors by selecting random values, the precision scores remained at the baseline level, or, at the frame level, the precision values rose slightly (i.e. from 54.3% to 58.1%). Regarding the recall scores, using the CMA-ES method to optimize the prior estimates led to higher values, but again the difference is not that high. These opposing trends, however, are probably only there because we maximized the F_1 values, hence a lower precision and a higher recall score is fine as long as it leads to the same (or higher) F-measure value.

6.3. Significance of the improvements

Recall that, to handle DNN random weight initialization, we trained five DNN models. Therefore, for each prior vector we get five F_1 scores, which allows us to test the significance of the improvements. Since we cannot expect the F-measure values to follow a normal distribution, we employed the Mann–Whitney U test (Mann and Whitney, 1947) (or Wilcoxon rank-sum test) to calculate the significance level for the improvements achieved.

The significance (i.e. p) values for the improvements achieved in the F_1 scores on the test set can be seen in Table 6; we denoted the cases without any improvement by "—". We can see that all improvements were significant at the level of $p < 0.01$, with the exception of filler events at the frame level when using the CMA-ES optimization method. Still, in this case, p appeared to be 0.0159, so our approach led to significantly better scores at the level of $p < 0.05$.

7. Discussion

First we focus on the results obtained for the standard approaches. Among the three tested approaches, full database sampling performed best; the (relatively) low performance of downsampling is probably due to the low variance of the samples used for training. In our opinion, uniform sampling suffered from the very same phenomenon, as only a fraction of the "miscellaneous" training samples were used within each training epoch, probably leading to the DNN model to overfit on the actual examples of the two minority classes.

The distribution of the different events affected our results at further points as well. By examining Table 1, we can confirm that there is a mismatch between the development set and the test set of this particular database: there are significantly more laughter events in the development set (4.7% of the duration) than either in the training set (3.4%) or in the test set (3.2%), and the filler events are somewhat more frequent there as well. (It was also noted in previous studies, e.g. in Gosztolya, 2015.) This phenomenon might explain why did the application of any statistical-based prior values lowered the segment-level scores, compared to not using any kind of class priors at all on the test set (see Table 3), while the corresponding values improved on the development set.

Another interesting observation was that the tendency of the scores corresponding to the laughter events were opposing for the level of segments and the level of frames. When using any of the statistical prior estimation strategies, we experienced improvements in the frame-level F_1 scores on the test set (Table 4), but the exactly same prior values led to lower F_1 scores on the level of segments (Table 3). This difference between the two evaluation approaches, in our opinion, indicates that the laughter occurrences themselves can be detected quite

reliably without dividing the DNN outputs by the a priori estimates. However, the starting and ending positions provided by this approach are quite imprecise, and could be improved by using even the statistical prior estimation strategy. This hypothesis is also supported by the corresponding precision and recall values (see Table 5): since laughter events are quite long, it is easy to find a part of them, which is regarded as a detection by the segment-level criteria, but at the level of frames missing parts of laughter events lead to a lower recall score. However, it is probably necessary for avoiding false alarms, therefore improving the precision values.

Overall, by employing the proposed method improved the combined F_1 values both on the level of segments and on the level of frames. This seems to indicate that the shortcomings of the baseline values do not simply come from the mismatch between the training and test sets or from the limited size of the training set alone. Another reason might be the tendency of the a posteriori scores (like the DNNs overestimating the probability of certain classes). Even using the original class priors leads to suboptimal social signal detection performance, and more reliable values can be obtained via optimization. Furthermore, it is beneficial to incorporate the Hidden Markov Model into the optimization process instead of focusing on simple frame-level classification.

7.1. Class prior values

Fig. 2 shows the prior estimates got by applying the methods described above. When we calculated these values via the statistical approach (see the first five cases), they appeared to be quite similar. Still, we can see that when we used the training and development sets, both prior estimates increased overall, but they fell when we made use of the test set as well. This probably explains why the F_1 values we got after applying the HMM also fell in this case.

Examining the prior estimates found by optimization, it is hard to see any general trend. (Recall that when we generated random vectors, the same values proved to be optimal at both the frame level and the segment level, so these two cases are shown as one in Fig. 2.) In our opinion, this is due to the limitations of the random optimization method, which cannot explore the local context of a good hypothesis. The two cases that involved applying the CMA-ES method, however, agree in that the a priori probability estimate of the laughter class should be much higher (around 7%–8%) than their occurrence in the dataset (about 4%). This was also reflected in the precision and recall scores (see Table 5): a higher prior estimate tends to reduce the transformed posterior scores, leading to fewer false alarms (i.e. higher precision) but also lower recall values.

For the filler events, we found different optimal values at the segment level and at the frame level, which indicates that it is relatively easy to locate filler events, but their precise starting and ending points are harder to determine. This is quite reasonable, though, as filler events can easily be confused with certain phonemes (see e.g. Gosztolya et al., 2019), which is not the case with laughter occurrences.

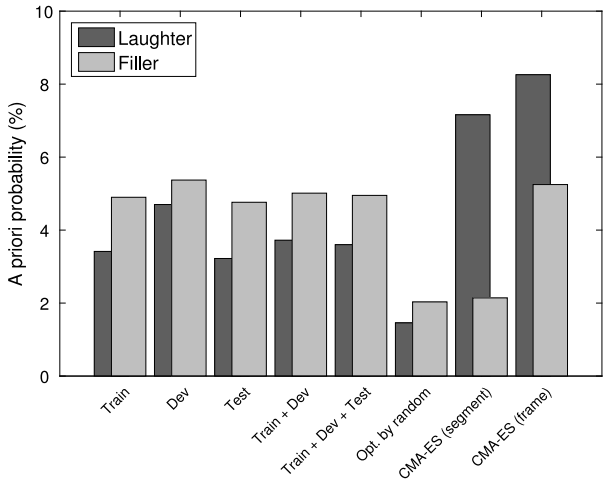


Fig. 2. The various a priori probability estimates determined by the different approaches tested.

8. Conclusions

In this study, we examined the class a priori probability estimates when using the DNN output scores as input for a Hidden Markov Model. First we raised theoretical objections about why we consider the prior values computed in the usual way to be only rough estimates, then we experimentally adjusted the class priors in order to improve segment and frame level accuracy scores. In the end, we found that prior optimization is a viable way of improving accuracy: by generating random class prior vectors and choosing the one which leads to the best accuracy score on the development set, we were able to reduce the error scores on the test set by 5%–6%. By applying the state-of-the-art optimization method of CMA-ES, we were able to achieve further improvements, leading to relative error reduction values of 6%–9%, found to be significant with $p < 0.01$.

By analyzing the F_1 values got by relying on the statistical approach, we found that this improvement was not simply due to the small size of training and/or development sets, but also due to the fact that we could also counter the bias present in the posterior estimates provided by the DNNs.

We validated our approach on a single, although relatively large database. Still, we do not think that it limits the potential applicability of the proposed prior estimate optimization method. Our approach, besides allowing to improve the efficiency of a social signal detection system, also offers a way to adjust the behavior of a previously trained HMM/DNN hybrid model (for example, in real application scenarios). Being an alternative to re-training the DNN acoustic model with a different class distribution or with new training data, our procedure of fine-tuning the class prior estimates could be a lightweight-yet-efficient solution for addressing this need as well.

Regarding the limitations of the method, we should mention that it is based on the assumption that the development and test sets are similar. Practically, by optimizing the prior vector we adjust the operation of the whole HMM/DNN system to best fit the development set. If the distribution of the specific events we seek to detect (in our case, laughter and filler events) is not similar in these two subsets (or the development set is not representative to real-life conditions), our performance scores could even fall below that of the baseline on the test set (or the HMM/DNN system would work suboptimally in the real-life scenario). Of course, this is a limitation of practically all methods that have a statistical basis (for example, machine learning techniques).

Another potential limitation of the approach is related to the number of classes. The size of the class prior vector is the same as the number of classes, which in our case was three (corresponding to laughter events, filler events and a class representing any other speech event); therefore, we performed the optimization step in a 3-dimensional space. However, the dimensionality of this search space increases proportionally to the number of classes; therefore, when the number of events is large, searching for the best prior vector becomes unfeasible. Such a setup, for example, is that of automatic speech recognition with context-dependent tied states (Odell, 1995), where the number of states (i.e. classes) is typically in the thousands.

Despite this, the application areas of the method proposed is not limited to that of social signal detection, or even to speech processing. In fact, in our view it can probably be applied for any task where using a Hidden Markov model is a straightforward option (of course, with the above-mentioned limitations in mind). These include, among others, speech synthesis (Amrouche et al., 2019; Csapó et al., 2016), handwriting recognition (Choudhury et al., 2019), sleep stage classification (Jian

et al., 2019), sign language recognition (Kumara et al., 2017) and calculating the clear-sky index for solar panel deployment (Shepero et al., 2019).

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.

Acknowledgments

Gábor Gosztolya was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and by the ÚNKP-21-5 New National Excellence Program by the Hungarian Ministry of Innovation and Technology. This study was partially funded by the National Research, Development and Innovation Office of Hungary via contract NKFIH FK-124413, by grant NKFIH-1279-2/2020 of the Hungarian Ministry of Innovation and Technology, and by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program (MILAB), Hungary.

References

Alaíz-Rodríguez, R., Guerrero-Curieses, A., Cid-Sueiro, J., 2011. Class and subclass probability reestimation to adapt a classifier in the presence of concept drift. Neurocomputing 74 (16), 2614–2623.

Amrouche, A., Abed, A., Falek, L., 2019. Arabic speech synthesis system based on HMM. In: Proceedings of ICEEE. Istanbul, Turkey. pp. 73–78.

Balikas, G., Partalas, I., Gaussier, E., Babbar, R., Amini, M.-R., 2015. Efficient model selection for regularized classification by exploiting unlabeled data. In: Proceedings of IDA. Saint Etienne, France. pp. 25–36.

Baur, T., Heimerl, A., Lingenfelser, F., Wagner, J., Valstar, M.F., Schuller, B., André, E., 2020. EXplainable cooperative machine learning with NOVA. KI – Künstl. Intell. 34, 143–164.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305.

Brueckner, R., Schmitt, M., Pantic, M., Schuller, B., 2017. Spotting social signals in conversational speech over IP: A deep learning perspective. In: Proceedings of Interspeech. pp. 2371–2375.

Brueckner, R., Schuller, B., 2013. Hierarchical neural networks and enhanced class posteriors for social signal classification. In: Proceedings of ASRU. pp. 362–367.

Brueckner, R., Schuller, B., 2014. Social signal classification using deep BLSTM recurrent neural networks. In: Proceedings of ICASSP. pp. 4856–4860.

Buda, M., Maki, A., Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. 106 (Oct), 249–259.

Chan, Y.S., Ng, H.T., 2006. Estimating class priors in domain adaptation for word sense disambiguation. In: Proceedings of ACL. Sydney, Australia. pp. 89–96.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of EMNLP. Doha, Qatar. pp. 1724–1734.

Choudhury, H., Mandal, S., Prasanna, S.R.M., 2019. Exploiting forced alignment of time-reversed data for improving HMM-based handwriting segmentation. Expert Syst. Appl. 121 (May), 158–169.

Csapó, T.G., Németh, G., Cerňak, M., Garner, P.N., 2016. Modeling unvoiced sounds in statistical parametric speech synthesis with a continuous vocoder. In: Proceedings of EUSIPCO. Budapest, Hungary. pp. 1338–1342.

du Plessis, M.C., Sugiyama, M., 2012. Semi-supervised learning of class balance under class-prior change by distribution matching. In: Proceedings of ICML. Edinburgh, UK.

Esuli, A., Sebastiani, F., 2015. Optimizing text quantifiers for multivariate loss functions. ACM Trans. Knowl. Discov. Data 9 (4), 27.

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.

Forman, G., 2008. Quantifying counts and costs via classification. Data Min. Knowl. Discov. 17 (2), 164–206.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier networks. In: Proceedings of AISTATS. pp. 315–323.

Gosztolya, G., 2015. On evaluation metrics for social signal detection. In: Proceedings of Interspeech. Dresden, Germany. pp. 2504–2508.

Gosztolya, G., Grósz, T., Tóth, L., 2020. Social signal detection by probabilistic sampling DNN training. IEEE Trans. Affect. Comput. 10 (1), 164–177.

Gosztolya, G., Vincze, V., Tóth, L., Pákáski, M., Kálmán, J., Hoffmann, I., 2019. Identifying Mild Cognitive Impairment and mild Alzheimer's disease based on spontaneous speech using ASR and linguistic features. Comput. Speech Lang. 53 (Jan), 181–197.

Gupta, R., Audhkhasi, K., Lee, S., Narayanan, S.S., 2013. Speech paralinguistic event detection using probabilistic time-series smoothing and masking. In: Proceedings of InterSpeech. pp. 173–177.

Gupta, R., Audhkhasi, K., Lee, S., Narayanan, S.S., 2016. Detecting paralinguistic events in audio stream using context in features and probabilistic decisions. Comput. Speech Lang. 36 (1), 72–92.

Hand, D.J., Yu, K., 2001. Idiot's Bayes – not so stupid after all? Internat. Statist. Rev. 69 (3), 385–398.

Hansen, N., Kern, S., 2004. Evaluating the CMA evolution strategy on multimodal test functions. In: Proceedings of PPSN. pp. 282–291.

Hansen, N., Ostermeier, A., 2001. Completely derandomized self-adaptation in evolution strategies. Evol. Comput. 9 (2), 159–195.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hopkins, D.J., King, G., 2010. A method of automated nonparametric content analysis for social science. Am. J. Political Sci. 54 (1), 229–247.

Jian, D., nan Lu, Y., Ma, Y., Wang, Y., 2019. Robust sleep stage classification with single-channel EEG signals using multimodal decomposition and HMM-based refinement. Expert Syst. Appl. 121 (May), 188–203.

Johnson, J.M., Khoshgoftaar, T.M., 2019. Survey on deep learning with class imbalance. J. Big Data 6.

Kantharaju, R., Ringeval, F., Besacier, L., 2018. Automatic recognition of affective laughter in spontaneous dyadic interactions from audiovisual signals. In: Proceedings of ICMI. pp. 220–228.

Kar, P., Li, S., Narasimhan, H., Chawla, S., Sebastiani, F., 2016. Online optimization methods for the quantification problem. In: Proceedings of KDD. San Francisco, CA, USA. pp. 1625–1634.

Kennedy, L., Ellis, D., 2004. Laughter detection in meetings. In: Proceedings of the NIST Meeting Recognition Workshop at ICASSP. Montreal, Canada. pp. 118–121.

King, G., Lu, Y., 2008. Verbal autopsy methods with multiple causes of death. Statist. Sci. 23 (1), 78–91.

Kumara, P., Gauba, H., Roy, P.P., Dogra, D.P., 2017. Coupled HMM-based multi-sensor data fusion for sign language recognition. Pattern Recognit. Lett. 86 (Jan), 1–8.

Mann, H.B., Whitney, D.R., 1947. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 18 (1), 50–60.

Mohamed, A., Dahl, G.E., Hinton, G., 2012. Acoustic modeling using deep belief networks. IEEE Trans. Audio Speech Lang. Process. 20 (1), 14–22.

Neuberger, T., Beke, A., 2013. Automatic laughter detection in Hungarian spontaneous speech using GMM/ANN hybrid method. In: Proceedings of SJUSK. pp. 1–13.

Neuberger, T., Beke, A., Gósy, M., 2014. Acoustic analysis and automatic detection of laughter in Hungarian spontaneous speech. In: Proceedings of ISSP. pp. 281–284.

NIST, 2006. NIST Spoken Term Detection 2006 Evaluation Plan. http://www.nist.gov/speech/tests/std/docs/std06-evalplan-v10.pdf.

Odell, J., 1995. The Use of Context in Large Vocabulary Speech Recognition (Ph.D. thesis). University of Cambridge.

Panzner, M., Cimiano, P., 2016. Comparing hidden Markov models and long short term memory neural networks for learning action representations. In: Proceedings of MOD. Volterra, Italy. pp. 94–105.

Pokorny, F.B., Peharz, R., Roth, W., Zöhrer, M., Pernkopf, F., Marschik, P.B., Schuller, B., 2016. Manual versus automated: The challenging routine of infant vocalisation segmentation in home videos to study neuro(mal)development. In: Proceedings of Interspeech. San Francisco, CA, USA. pp. 2997–3001.

Saerens, M., Latinne, P., Decaestecker, C., 2002. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. Neural Comput. 14 (1), 21–41.

Salamin, H., Polychroniou, A., Vinciarelli, A., 2013. Automatic detection of laughter and fillers in spontaneous mobile phone conversations. In: Proceedings of SMC. pp. 4282–4287.

Schmitt, M., Cummins, N., Schuller, B., 2019. Continuous emotion recognition in speech – Do we need recurrence? In: Proceedings of Interspeech. Graz, Austria. pp. 2808–2812.

Schuller, B., 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.

Shepero, M., Munkhammar, J., Widén, J., 2019. A generative hidden Markov model of the clear-sky index. J. Renew. Sustain. Energy 11 (4).

Tóth, L., 2013. Phone recognition with deep sparse rectifier neural networks. In: Proceedings of ICASSP. pp. 6985–6989.

Tóth, L., Kocsor, A., Csirik, J., 2005. On naive Bayes in speech recognition. Int. J. Appl. Math. Comput. Sci. 15 (2), 287–294.

Truong, K.P., van Leeuwen, D.A., 2007. Automatic discrimination between laughter and speech. Speech Commun. 49 (2), 144–158.

Wheatley, B., Doddington, G., Hemphill, C., Godfrey, J., Holliman, E., McDaniel, J., Fisher, D., 1992. Robust automatic time alignment of orthographic transcriptions with unconstrained speech. In: Proceedings of ICASSP. pp. 533–536.