Using Phonetic Patterns for Detecting Social Cues in Natural Conversations

Johannes Wagner, Florian Lingenfelser, Elisabeth André

University of Augsburg, Lab for Human Centered Multimedia, Germany

[wagner,lingenfelser,andre]@hcm-lab.de

Abstract

Laughter and fillers like "uhm" and "ah" are social cues expressed in human speech. Detection and interpretation of such non-linguistic events can reveal important information about the speakers' intensions and emotional state. The INTERSPEECH 2013 Social Signals Sub-Challenge sets the task to localize and classify laughter and fillers in the "SSPNet Vocalization Corpus" (SVC) based on acoustics. In the paper at hand we investigate phonetic patterns extracted from raw speech transcriptions obtained with the *CMU Sphinx* toolkit for speech recognition. Even though *Sphinx* was used out of the box and no dedicated training on the target classes was applied, we were able to successfully predict laughter and filler frames in the development set with $\sim 87\%$ accuracy (unweighted average Area Under the Curve (AUC)). By accumulating our features with a set of standard features provided by the challenge organizers results increased above 92%. When applying the combined set to the test corpus we achieved 87.7% as highest score, which is 4.4% above the challenge baseline.

Index Terms: Computational Paralinguistics, Social Signals, Phonetic Features

1. Introduction

Laughter and fillers like "uhm" and "ah" are social cues expressed in human speech. Detection and interpretation of such non-linguistic events can reveal important information about the speakers' intensions and emotional state [1]. Laughter, for instance is often a sign of amusement or joy, though it may also show scorn or embarrassment [2]. Fillers, on the other hand, are indicators to hold the floor or signal that what was said will be revised by the speaker [3]. The paper at hand deals with the automatic detection of laughter and fillers in natural conversations at frame level. We therefore investigate phonetic patterns extracted from raw speech transcriptions in the neighbourhood of a frame.

1.1. Related Work

Previous work has focused both on the study of the acoustic and phonetic characteristics of human laughter [2, 4, 5, 6] and the development of automated laughter detectors [7, 8, 9, 10, 11, 12, 13, 14, 15]. Such studies provide valuable information to guide algorithms for the automated detection of laughter.

Besides studies on the acoustic and phonetic properties of laughter, a variety of attempts has been made to automatically recognize laughter and other affect bursts from audio data. A common approach is to train Hidden Markov Models (HMMs) on data for specific non-speech fillers. For example, Kennedy and Hauptmann [7] extended the Sphinx speech recognizer by the ability to recognize non-word sounds, such as dog barking or laughter, by training a small set of HMM parameters specifically on these sounds. This approach bears the advantage that it can be easily integrated into an HMM based speech recognizer. However, since there is a large variety of laughter, it is hard to detect laughter based on a single model only. For these reason, later approaches tried to identify different laughter categories and built specific models for them. For example, Tanaka and Campbell [14] investigated four phonetic categories of laughter (nasal, voiced, chuckle, or ingressive) and trained HMMs on them using spectral features. With this approach, the single laughter categories could be recognized with an accuracy rate of 86.79%. Schuller and colleagues [13] tested various classifiers using spectral features to distinguish four kinds of isolated non-verbal vocalization (laughter, breathing, hesitation, and consent). In their experiment, HMMs achieved an accuracy rate of 92.3% and outperformed Hidden Conditional Random Fields (HCRFs) and Support Vector Machines (SVMs).

A more recent paper by Scherer and colleagues [15] showed that the performance of a laughter classifiers depends on whether it is applied in online or offline mode. SVMs gave the best results for offline mode with given segmentations. However, in online mode, they were outperformed by Echo State Networks (ESNs) and HMMS because they failed to find on- and offsets of laughter. A particular challenge in laughter detection is overlapping laughter in conversational speech. Truong and Trouvain [16] conducted a study using four corpora in order to identify spectral features that differentiate overlapping and non-overlapping laughter.

Overall, it may be said that a large number of acoustic and phonetic features has been investigated in the context of social cue detection. However, to our knowledge, the use of phonetic patterns obtained from an automated transcription process has not yet been explored so far.

1.2. Approach Using Phonetic Patterns

Of particular interest to our work is a recent study by Dutoit and Urban [6] who performed a phonetic transcription of laughter. They found that individuals use rather different subsets of laughter phones. On average, about thirty phones per laughter occurred in the corpus they investigated. Since several sounds could not be found in the International Phonetic Alphabet, Dutoit and Urban used additional labels for the manual transcription of laughter, such as groan, snore, and grunt. Even though Dutoit and Urban focus on laughter synthesis, their work provides useful insights for laughter detection as well. However, the transcription of laughter has been conducted manually in their work.

Regarding fillers, it is common approach to augment speech recognition with according models to detect so called Out Of Vocabulary (OOV) [17]. Today's speech recognizers usually come with pre-trained models for such sounds.

The objective of our work is therefore to investigate how far we can get with a speech recognizer that provides us with pho-

Phonemes
AA AE AH AO AW AY B CH D DH EH ER EY F G HH IH IY JH K L M N NG OW OY P R S SH T TH UH UW V W Y Z ZH SIL
Fillers
+BREATH+ +NOISE+ +COUGH+ +GARBAGE+ +SMACK+ +UH+ +UM+ +UHUM+

Table 1: Complete listing of all phonemes and fillers included in the dictionary used with the *Sphinx* recognizer.

netic labels that have not been specifically trained on samples of laughter or fillers from the analyzed corpus. Instead we apply the speech recognizer out of the box and investigate whether we will find distinctive patterns for laughter and filler segments. We evaluate our approach by means of the INTERSPEECH 2013 Social Signals Sub-Challenge, which sets the task to localize and classify laughter and fillers within the "SSPNet Vocalization Corpus" (SVC). SVC is composed of $2'763$ audio clips (11 seconds length each) collected in 60 phone calls involving 120 subjects. For more information on the data set please refer to the challenge paper [18].

2. Methodology

In the following we describe the phonetic features we extract and the classification method that will be used for evaluation.

2.1. Transcription

First of all, we need a phonetic transcription for the raw audio files. We decided to employ a speech recognizer where phoneme detection is applied as a preliminary step. In speech recognition the next step would be to map detected phoneme strings to words in a dictionary, before finally sentences will be created according to a specific grammar. During the latter steps parts of speech that do not fit applied models are removed or mapped to the most probable sequence of words. However, for our purpose we actually require the raw phonetic transcriptions as we expect them to contain important hints about potential fillers and laughter.

For the following experiments we apply the *CMU Sphinx* toolkit for speech recognition, an open source project by Carnegie Mellon University[1]. We take version 3 of the *Sphinx* recognizer as it offers direct access to the phonetic transcription. Therefore we switch the detecting mode to `allphone` and choose one of the pre-trained recognition models that comes with *Sphinx*. It should be noted that it was not our purpose to choose a speech model specially suited for detecting fillers or laughter (in fact the chosen model does not contain a laughter class), nor did we re-train the model using data of this or any other corpus. We rather used *Sphinx* and one of its model as a sort of a black box to see how far we can get with it[2].

A complete list of phonemes and fillers the chosen model is able to detect is given in Table 1. Having set up *Sphinx* we can now pass a raw audio file to get a list of phonemes along with timestamps and probability scores.

[1] `http://cmusphinx.sourceforge.net/`

[2] Basically, we just followed the description at `http://cmusphinx.sourceforge.net/wiki/phonemerecognition` and used the suggested model file `interp_nodx.arpa.dmp`.

2.2. Phonetic Feature Extraction

Having retrieved a phonetic transcription for each audio file in the database we now proceed by calculating for each frame a set of features from the retrieved sequences. We choose a frame length of 10 ms as this will later on allow us to combine our feature set with the one provided by the challenge organizers.

Next, we need a compact representation for the frames describing the distribution of phonemes within the surrounding of a frame. As illustrated in Figure 1 we calculate a histogram, which stores the frequency of each phoneme within a certain range. We define this range by the frame itself plus n frames to the left and n frames to the right. Hence, by varying parameter n we can control the context we want to use when extracting the features. Obviously, the length of the feature vector equals the number of phonemes (and fillers) in the dictionary: in our case 48 features (40 phonemes and 8 fillers). Finally, we normalize feature values by dividing them by the context length ($2 \times n + 1$), so that feature vectors sum up to 1. In the following we denote this feature set as *pho-1*.

Since a histogram stores only the frequencies at which phonemes occur, while no information about the temporal ordering is kept, we might lose important temporal cues, e. g. on the rhythmic of laughter or that a filler is represented by a certain order of phonemes. To capture those cues we implement a second type of feature: instead of calculating a single histogram over the whole context, we extract two – one for the left context (including the current frame) and one for the right context. In this way we hope to measure differences in the phonetic distributions on the left and right context of a frame. Obviously, this doubles the number of features in the set. In the following this feature set is called *pho-2*.

2.3. Classification

To allow for a fair comparison with the baseline features we stick to the classification scheme suggested by the challenge organizers. That is we also use the WEKA data mining toolkit [19] and employ a linear kernel Support Vector Machines (SVM) with Sequential Minimal Optimisation (SMO). For simplicity we keep the complexity parameter C fixed at 10^{-1}, which gave best results for the baseline features. To train the classifier we first extract features on the complete training set ($1'735'770$ frames) and downsample them in the exact same way as the challenge organizers ($79'572$ frames). We then evaluate the trained model using the development set ($547'789$ frames). As proposed by the challenge organizer we measure the Unweighted Average Area Under the Curve (UAAUC) for the laughter and filler classes on frame level.

3. Results and Discussion

3.1. Phoneme Frequencies

Before we discuss the quality of the phonetic features in terms of recognition results let us first have a look at the distributions of phonemes among target classes. If it is possible to distinguish laughter and filler from garbage frames using the proposed features we should expect some differences in the frequency at which certain phonemes occur with respect to the classes.

Table 2 lists counts for the 5 most frequent phonemes per target class. We see that the most frequent phoneme occurring within garbage is the phoneme for silence (`SIL`). This reflects the fact that large portions of the audio files actually contain silence, e. g. when the operator is speaking. However, `SIL` turns

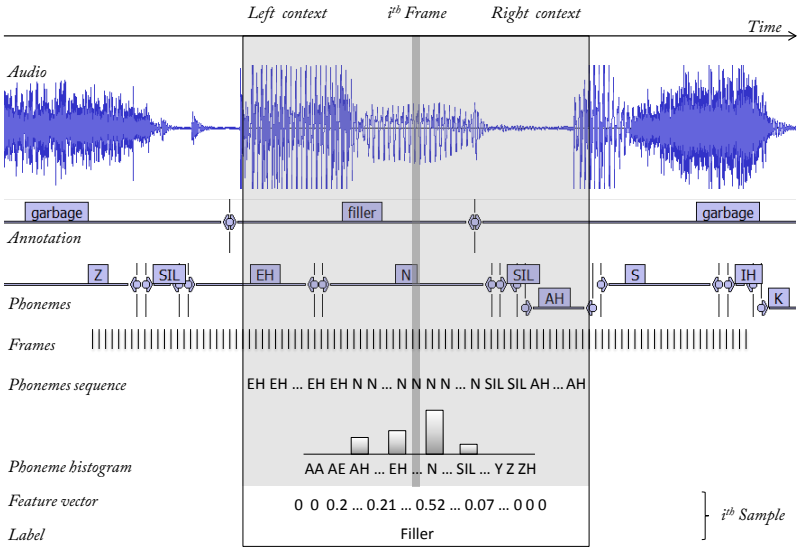

Figure 1: Extraction of phonetic features: first, for each frame the sequence of phonemes is determined by mapping the phonetic transcription on frames of 10 ms length. The context size n defines how many frames to the left and right are taken into account. Then, a histogram is built by counting the occurrence of each phoneme in the sequence. Finally, the relative phoneme frequencies are stored as features labelled by the class of the centre frame.

garbage		laughter		filler	
SIL	7406	+C+	672	N	894
IH	6521	SIL	157	SIL	608
N	5457	N	117	AE	523
AH	5316	T	106	EH	398
T	5049	IH	94	D	339

Table 2: Counts for the 5 most frequent phonemes per target class on segment level in the training set. Note that due to lack of space +COUGH+ has been shortened to +C+.

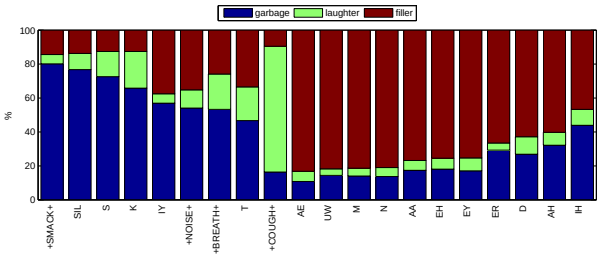

Figure 2: Relative frame frequency per class for the 20 most frequent phonemes (90.1% of total mass) on the downsampled training set (no context).

out to be a frequent phoneme within laughter and filler segments, as well. Probably because both vocalizations are likely to be surrounded by silence as they interrupt the normal speech flow. Interestingly, none of the pre-trained filler phonemes (see Table 1) seems to occur very often during filler vocalizations. However, we should recall that the speech recognizer has not been trained on the corpus we are analysing. Apparently, the pre-trained filler models simply fail to recognize fillers on our corpus.

Regarding laughter our findings are actually in line with the study by [6] even though they conducted a manual transcription using a phoneme set adapted to laughter while we started from the phoneme set provided by the Sphinx speech recognizer. The SIL phoneme, which was our most dominant phoneme after the cough sound, was the phoneme most frequently observed by them in laugher exhalation phases. Also, the non-central vowel I and the nasal phoneme N were among their most frequent phonemes. Dutoit and Urbain found the large number of the plosive T a bit surprising, an observation which could, however, also be confirmed by our study. We did not find a similar large number of the breathy phoneme HH as reported in [6] because the Sphinx dictionary we used included a filler +BREATH+ onto which breathy sounds were mapped.

So far we have observed annotations on segment level. This might be misleading, because even if a phoneme occurs regularly within two classes the duration at which it occurs may still differ. Figure 2 compares occurrences of the 20 most frequent phonemes (90.1% of total mass) among classes on frame level. To have a fair comparison numbers are obtained for the downsampled training set where the number of frames per class has been balanced. In fact, we can observe clear preferences of some of the phones for certain classes. For instance, phoneme +SMACK+ has a 80% probability to occur during garbage frames. Likewise the phonemes AE or M are popular filler phones, probably reflecting the sequence {AE M}, which is a popular filler pattern. The most prominent phoneme correlating with laughter is still +COUGH+. It seems that the filler model for coughing actually shares very similar phonetic properties with laughter events in our corpus.

3.2. Recognition Results

The tendency of certain phonemes to correlate with certain classes give grounds to believe that we can successfully train a classifier and predict target classes above chance level. As described in Section 2 we use two phonetic feature sets *pho-1* and *pho-2* to train a Support Vector Machines (SVM) classifier on the downsampled training set while stepwise varying the context size n. Results obtained for the development set in terms

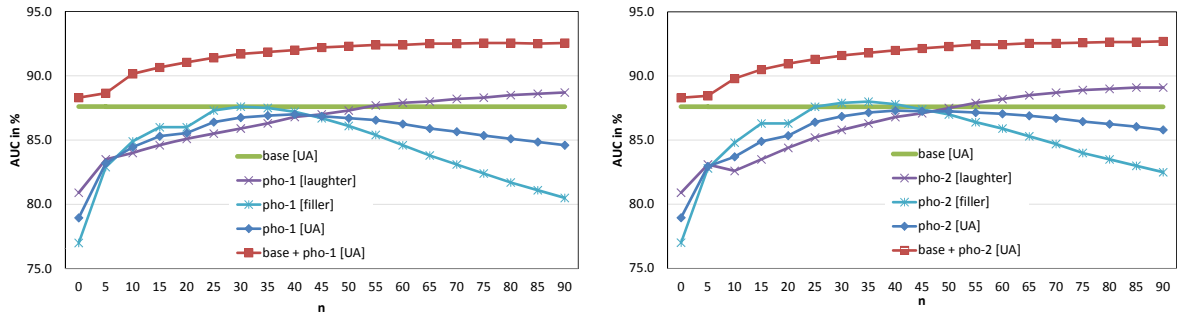

Figure 3: Area Under the Curve (AUC) in % for phonetic features *pho-1* (left) and *pho-2* (right) as a function of the context length n. UA = unweighted average.

of Unweighted Average Area Under the Curve (UAAUC) for the laughter and filler classes on frame level are summarized in Figure 3. For comparison the baseline (*base*) provided by the challenge organizers is also given, as well as, results obtained when combining baseline and phonetic features (*base* + *pho-1* and *base* + *pho-2*). Baseline features were extracted using TUM's open-source openSMILE feature extractor [20] (see [18] for the types of features).

From the graphs we can conclude that using solely phonetic features and a context size around 40 frames almost matches results for the baseline features ($\sim 87\%$). It turns out that laughter is generally better detected than fillers and that results still improve for context lengths of 70 and more, whereas for the filler class a maximum is observed at $n = 30$. Probably because the average segment length is smaller for fillers ($0.50s$) than for laughter ($0.91s$). The latter comes close to the average duration reported by Bachorowski ($0.87s$) [4]. Though *pho-2* generally works slightly better than *pho-1* the results achieved for both feature sets are more or less the same. Only, when $n > 50$ it seems that fillers are slightly better detected ($\sim 2\%$) if separate histograms are used for the left and right context. We take this as a hint that capturing more temporal information would actually benefit the detection of fillers.

Finally, we observe an improvement of about 5% when phonetic features and baseline features are combined to a single set. Again, the choice of the context length has an significant influence on the quality of the phonetic feature set and should be large enough to capture sufficient information about neighbouring phonemes. In our experiments results become stable for $n > 50$, which corresponds to a window length of more than one second (which is actually roughly the average length of a laughter segment). Results on the test corpus are summarized in Table 3. For the best configuration (*base* + *pho-2* and $n = 80$) we achieve 87.7%, which is 4.4% above the baseline.

	UAAUC in %					
	base		*base* + *pho-1* ($n = 70$)		*base* + *pho-2* ($n = 80$)	
	devel	*test*	*devel*	*test*	*devel*	*test*
laughter	86.2	82.9	88.2	89.1	93.3	89.4
filler	89.0	83.6	83.1	85.8	92.0	85.9
mean	87.6	**83.3**	92.5	**87.5**	92.7	**87.7**

Table 3: Recognition results for different feature sets on development and test set: *base* = baseline features, *base* + *pho-1* = baseline features and phonetic features extracted on single histogram, *base* + *pho-2* = baseline features and phonetic features extracted on two independent histograms.

4. Conclusion

In the paper at hand we explored the use of phonetic patterns to detect social cues in natural conversations, namely laughter and fillers like "uhm" and "ah". Therefore, we applied the *CMU Sphinx* speech recognizer to receive phonetic transcriptions of the raw audio files and investigated how far we can get with the phonetic labels. Note that the recognition model has not been specifically trained on samples of laughter or fillers from the analyzed corpus.

When looking at which phonemes correlated most frequently with the laughter class we made a number of findings that are in line with the study by [6]. For instance, the non-central vowel `I` and the nasal phoneme `N` were also among their most frequent phonemes. A phoneme frequently correlating with garbage was `SIL`, which is not surprising as large parts of the files actually contain silence. Phonemes `AE` and `M` turned out to be frequent filler phones, probably reflecting the sequence [`AE M`], which is a popular filler pattern.

Finally, we evaluated our approach by means of the INTERSPEECH 2013 Social Signals Sub-Challenge [18], which sets the task to localize and classify laughter and fillers. Therefore, we employed a linear kernel Support Vector Machines (SVM). When choosing a context length long enough to capture sufficient information about the neighbouring frames (> 1 second) we gained an unweighted average AUC of $\sim 87\%$ on the development set. We could further improve results by more than 5% when merging our features with those provided by the challenge organizers. On the test corpus we achieved 87.7% as highest score, which is a 4.4% above the baseline. The results show that social cues can be reliable detected by observing phonetic patterns, even if they were not explicitly included during the training of the speech recognition model.

In future we may further improve results by extending phonetic transcriptions with additional phonemes, for instance the ones used by Dutoit and Urbain in their manual laughter transcriptions [6]. Filler detection would probably benefit from adding features that are better suited to capture the temporal sequence of phonemes.

5. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement $n°270780$.

6. References

[1] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image Vision Comput.*, vol. 27, no. 12, pp. 1743–1759, Nov. 2009.

[2] W. Ruch and P. Ekman, "The Expressive Pattern of Laughter," *Emotion qualia, and consciousness*, pp. 426–443, 2001.

[3] H. Clark, "Using uh and um in spontaneous speaking," *Cognition*, vol. 84, no. 1, pp. 73–111, May 2002.

[4] J. A. Bachorowski, M. J. Smoski, and M. J. Owren, "The acoustic features of human laughter." *J Acoust Soc Am*, vol. 110, no. 3 Pt 1, pp. 1581–97, 2001.

[5] J. Trouvain, "Segmenting phonetic units in laughter," in *Proc. of the 15th International Congress of Phonetic Sciences*, 2003, pp. 2793–2796.

[6] J. Urbain and T. Dutoit, "A phonetic analysis of natural laughter, for use in automatic laughter processing systems," in *Affective Computing and Intelligent Interaction - 4th International Conference, ACII 2011, Memphis, TN, USA, October 9-12, 2011, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 6974. Springer, 2011, pp. 397–406.

[7] P. E. Kennedy and A. G. Hauptmann, "Laughter extracted from television closed captions as speech recognizer training data," in *Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999, Budapest, Hungary, September 5-9, 1999*. ISCA, 1999.

[8] L. Kennedy and D. Ellis, "Laughter detection in meetings," in *NIST Meeting Recognition Workshop at ICASSP*, 2004, pp. 118–121.

[9] M. T. Knox and N. Mirghafori, "Automatic laughter detection using neural networks," in *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007*. ISCA, 2007, pp. 2973–2976.

[10] K. P. Truong and D. A. van Leeuwen, "Automatic discrimination between laughter and speech," *Speech Communication*, vol. 49, no. 2, pp. 144–158, 2007.

[11] K. Laskowski and T. Schultz, "Detection of laughter-in-interaction in multichannel close-talk microphone recordings of meetings," in *Machine Learning for Multimodal Interaction, 5th International Workshop, MLMI 2008, Utrecht, The Netherlands, September 8-10, 2008. Proceedings*, ser. Lecture Notes in Computer Science, vol. 5237. Springer, 2008, pp. 149–160.

[12] S. Petridis and M. Pantic, "Fusion of audio and visual cues for laughter detection," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, ser. CIVR '08. New York, NY, USA: ACM, 2008, pp. 329–338.

[13] B. Schuller, F. Eyben, and G. Rigoll, "Static and dynamic modelling for the recognition of non-verbal vocalisations in conversational speech," in *Proceedings of the 4th IEEE tutorial and research workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems*, ser. PIT '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 99–110.

[14] H. Tanaka and N. Campbell, "Acoustic features of four types of laughter," in *The 17th International Congress of Phonetic Sciences (ICPhS XVII)*. City University of Hongkong, 2011, pp. 1958–1961.

[15] S. Scherer, M. Glodek, F. Schwenker, N. Campbell, and G. Palm, "Spotting laughter in natural multiparty conversations: A comparison of automatic online and offline approaches using audiovisual data," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, no. 1, p. 4, 2012.

[16] K. P. Truong and J. Trouvain, "On the acoustics of overlapping laughter in conversational speech," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, 2012.

[17] M. Dunnachie, P. Shields, D. Crawford, and M. Davies, *Filler Models for Automatic Speech Recognition Created from Hidden Markov Models using the K-Means Algorithm*, 2009.

[18] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Proceedings of Interspeech*, Sep. 2013.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[20] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1459–1462.