KI - Künstliche Intelligenz (2020) 34:143–164
https://doi.org/10.1007/s13218-020-00632-3

TECHNICAL CONTRIBUTION

eXplainable Cooperative Machine Learning with NOVA

Tobias Baur[1] · Alexander Heimerl[1] · Florian Lingenfelser[1] · Johannes Wagner[1] · Michel F. Valstar[2] · Björn Schuller[1] · Elisabeth André[1]

Received: 30 September 2019 / Accepted: 2 January 2020 / Published online: 19 January 2020

Abstract
In the following article, we introduce a novel workflow, which we subsume under the term "explainable cooperative machine learning" and show its practical application in a data annotation and model training tool called NOVA. The main idea of our approach is to interactively incorporate the 'human in the loop' when training classification models from annotated data. In particular, NOVA offers a collaborative annotation backend where multiple annotators join their workforce. A main aspect is the possibility of applying semi-supervised active learning techniques already during the annotation process by giving the possibility to pre-label data automatically, resulting in a drastic acceleration of the annotation process. Furthermore, the user-interface implements recent eXplainable AI techniques to provide users with both, a confidence value of the automatically predicted annotations, as well as visual explanation. We show in an use-case evaluation that our workflow is able to speed up the annotation process, and further argue that by providing additional visual explanations annotators get to understand the decision making process as well as the trustworthiness of their trained machine learning models.

Keywords Annotation · Cooperative machine learning · Explainable AI

1 Motivation

In various research disciplines (Behavioural Psychology, Medicine, Anthropology,...) the annotation of social behaviours is a common task. This process includes manually identifying relevant behaviour patterns in audio-visual material and assigning descriptive labels. Generally speaking, segments in the signals are mapped onto a set of discrete classes, e.g., a certain type of gesture, a social situation (e.g., conflict), or the emotional state of a person.

In Affective Computing, a subset of these events—the so called *social signals*—are used to augment the spoken part of a message with non-verbal information to enable a more natural human–computer interaction. [54, 55][1]. To automatically detect social signals from raw sensory input (e.g., speech signals) machine learning (ML) can be applied. That is, sensory input is transformed into a compact set of relevant features and a classifier is trained on manually labelled examples to optimise a learning function. Once trained, the classifier can be used to automatically predict labels on unseen data.

However, since humans transmit non-verbal messages through a number of channels (voice, face, gestures, etc.) and due to the complex interplay between these channels (think, for instance, of a faked versus a real smile, which depends on subtle contractions of the muscles at the corner of the eyes as well as the timing of the muscle activations [53]), progress in SSP is directly linked to the availability of large and well transcribed multi-modal databases rich of human behaviour under varying context and different environmental settings [16]. Common challenges in creating such datasets lie in the high degree of naturalness required of the recording scenarios, how well one recording scenario generalises to other settings, the number of human raters needed to reach a consensus on labels, and of course the sheer amount of data. Thinking of the many hours of labelled data that are required, it is clear that gathering large amounts of annotated training samples seems like an infeasible task, in respect to time, cost and effort.

Even though there exists a vast resource of raw data, which is nowadays pervasive in digital format and relatively

✉ Tobias Baur
baur@hcm-lab.de

[1] Augsburg University, Universitätstr. 6a, Augsburg, Germany

[2] University of Nottingham, Nottingham, UK

[1] To give an example of a social signal, think of a situation where we say something in a sarcastic voice to indicate that we actually mean the opposite.

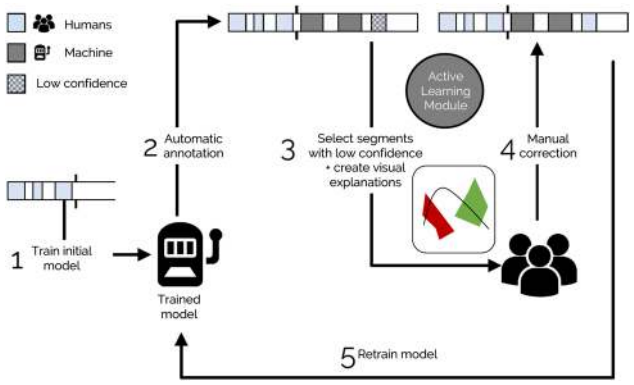

Fig. 1 The scheme depicts the general idea behind Cooperative Machine Learning (CML): (1) an initial model is trained on partially labelled data. (2) The initial model is used to automatically predict unseen data. (3) Labels with a low confidence are selected, and additionally visual explanations are generated and (4) they get manually revised by the annotator. (5) The initial model is retrained with the predicted/revised data

easy and inexpensive to collect, e.g., from public resources such as social media, the problem of gathering relevant annotations still needs to be overcome. One approach is the *Active Learning* (AL) [67] algorithm that interactively query the user to manually label certain data points. The core idea of AL is to extract the most informative instances from a pool of unlabelled data based on a specific query strategy [46]. These selected instances are then passed to human annotators and finally—after labelling—a model is derived from this subset. This, of course, reduces the labelling effort. In addition, it has two more positive side effects. First, it speeds up the training since fewer instances have to be processed. Second, it helps improving the maximum accuracy, as it reaches a more coherent learning model (focussing on the most relevant cases).

In this article, we subsume learning approaches that efficiently combine human intelligence with the machine's ability of rapid computation under the term *Cooperative Machine Learning* (CML) [15, 66]. In Fig. 1 we illustrate our approach, which creates a loop between a machine learned model and human annotators: an initial model is trained (1) and used to predict unseen data (2). An active learning module then decides which parts of the prediction are subject to manual revision by human annotators. A human annotator additionally has the option to create a visual explanation for each decision of the model (3+4). Afterwards the initial model is retrained using the new labelled data (5). Now the procedure is repeated until all data is annotated. By actively incorporating human expert knowledge into the learning process it becomes possible to interactively guide and improve the automatic prediction. Hence, the approach bears the potential to considerably cut down manual efforts. For instance, the system may quickly learn to label some simple behaviours, which already facilitates the work load for human annotators at an early stage. Then over time, it could learn to cope with more complex social signals as well, until at some point it is able to finish the task in a completely automatic manner. Such an iterative approach may even help bridging the gap between quantitative and qualitative coding, which still defines a great challenge in many fields in social science [9]. Additionally we implemented recent explainable AI techniques to not only identify the parts where the machine is not confident about it's predictions, but to also be able to provide visual explanations about the model's decision criteria.

In this paper, we introduce a next-generation annotation tool called NOVA, which implements the described workflow that interactively incorporates the 'human in the loop' [24, 26]. In particular, NOVA offers semi-automated annotations and provides visual feedback to inspect and correct machine-generated labels by incorporating eXplainable AI (XAI) techniques. In that sense, our work combines three recent topics of ML: *Explainable Artificial Intelligence*, as the transparency of the decision process is increased via visualisation of the predictions; *Semi-Supervised Active Learning*, since labels with low confidence are highlighted to guide the user towards relevant parts; and finally, *Interactive Machine Learning* [25], because human intelligence and machine power can cooperate and improve each other.

We subsume our overall approach under the term *eXplainable Cooperative Machine Learning* (XCML). We see the main contributions of this work as follows:

- In Sect. 2, we propose a novel two-step CML strategy: as long as only few labelled instances are available the system is applied to local fractions of the database. Later, as more labelled instances become available, larger parts can be predicted.
- In Sect. 3, we evaluate the proposed strategy on an audio-based annotation task by simulating the incremental injection of additional information during training. Results show that the proposed strategy significantly reduces manual coding efforts.
- In Sect. 4, we introduce an open-source tool for collaborative and machine-aided labelling (NOVA). A walkthrough is presented to demonstrate the collaborative annotating capability of the system.
- In Sect. 5, we describe how explainable AI techniques may extend the proposed Cooperative Machine Learning workflow, to not only speed up the process, but to also give better understanding to users of such a system how well their model performs and why it fails or succeeds.
- In Sect. 6, we discuss experiences, limitations and chances of applying Cooperative Machine Learning in the annotation process from the perspective of end-users.

For the sake of clarity related work will be given separately for each section.

2 Cooperative Machine Learning

Interactive machine learning [2, 18] aims to involve users actively in the creation of models for recognition tasks. Most IML approaches integrate automated data analysis and interactive visualisation tools in order to enable users to inspect data, process features and tune models. One main aspect of interactive machine learning is the goal of integrating end-users in the training process of Machine Learning models, making this process easier accessible to non-Machine Learning experts. In this section, we focus on approaches that facilitate the acquisition of annotated data sets and introduce a novel methodology for applying *Cooperative Machine Learning* (CML) to speed up annotation of social signals in large multi-modal databases and to involve the user actively in the Machine Learning loop.

2.1 Related Work

A common approach to reduce human labelling effort is the selection of instances for manual annotation based on active learning techniques. The basic idea is to forward only instances with low prediction certainty or high expected error reduction to human annotators [47].

An art of its own right thereby is how to estimate which are these most informative ones. A whole range of options to choose from exist, such as calculation of 'meaningful' confidence measures, detecting novelty (e.g., by training auto-encoders and seeing for the deviation of input and output when new data runs through the auto-encoder), estimating the degree of model change the data instance would cause (i.e., seeing whether knowing the label of a data point would make a change to the model at all), or trying to track 'scarce' instances, i.e., trying to find those data instances that are rare in terms of the expected label.

Further more sophisticated approaches aggregate the results of machine learning and crowdsourcing processes to increase the efficiency of the labelling process. Kamar et al. [27] made use of learned probabilistic models to fuse results from computational agents and human labellers. They showed how to allocate tasks to coders in order to optimise crowdsourcing processes based on expected utility. Zhang et al. [64] developed an agreement-based annotation technique that dynamically determines how many human annotators are required to label a selected instance. The technique considers individual rater reliability and inter-rater agreement to decide on a combination of raters to be allocated to an instance. Active learning has shown great potential in a large variety of areas including document mining [50], multimedia retrieval [62], activity recognition [49] and emotion recognition [65].

Most studies in this area focus on the gain obtained by the application of specific active learning techniques. However, little emphasis is given to the question of how to assist users in the application of these techniques for the creation of their own corpora. While the benefits of integrating active learning with annotation tasks has been demonstrated in a variety of experiments, annotation tools that provide users with access to active learning techniques are rare. Recent developments for audio, image and video annotation that make use of active learning include CAMOMILE [39] and iHEARu-PLAY [23]. However, systematic studies focusing on the potential benefits of the active learning approach within the annotation environment from a user's point of view have been performed only rarely [10, 29].

While techniques that enable systems to learn from human raters have become widespread, little attention has been paid to usability challenges of the remaining tasks left to end-users [2]. Rosenthal et al. [42] investigated which kind of information should be provided to users in order to reduce annotation errors in a setting for active learning. They found out that contextual information and predictions of the learning algorithms were in particular useful for the annotation of activity data. In contrast, uncertainty information had no effect on the accuracy of the labels, but just indicated to the labellers that classification was hard. Amershi [4] investigated how to empower users to select samples for training by appropriate visualisation techniques. They found that a representative overview of best and worst matching examples is of higher value than a set of high-certainty images and conjecture that high-certainty images do not provide much information to the learning processing due to their similarity to already labelled images. In another paper by Amershi et al. [3] the authors suggest an interactive visualization technique to assess model performance by sorting samples according to their prediction scores. In their tool the user can directly inspect samples to retrieve additional information and annotate them for better performance tracking. This way, the tool allows users to monitor the performance of individual samples while the model is iteratively retrained.

The approaches above supported users in the annotation and selection of samples for training. As an alternative, graphical user interfaces have been developed that enable users to create their own annotated examples for training models. Typically, the labels are given by instructions or stimuli to be provided to the users to evoke particular behaviours. An example includes SSI/ModelUI [57]. It presents users with a graphical user interface that allows them to test different machine learning algorithms on labelled data. Labels are acquired by stimuli which may include textual instructions, but also images or videos. However, users have

to determine themselves which kind of stimuli and data are most useful to create and tune models.

Summing up, it may be said that many studies experientially investigate the potential of novel techniques to minimise human labour. In addition, few studies were run to actually label novel data, rather than test whether such method could save effort. Also note that the prevailing choice is merely active learning rather than the combination with semi-supervised learning, i.e., cooperative machine learning. Relatively little attention has been paid, however, to the question of how to make these techniques available to human labellers. There is a high demand for annotation tools that integrate cooperative machine learning in order to reduce human effort—in particular in the area of social signal processing where human raters typically disagree on the labels [36].

2.2 Two-Fold Strategy

The cooperative machine learning strategy we propose here is a two-fold one. It is divided into a *Session Completion* (SC) step during which information of a fraction of a single session is used to complete the remaining part of the session, and a *Session Transfer* (ST) step during which information from a set of labelled sessions is used to predict a set of unlabelled sessions. In our understanding, a session defines a single continuous and self-contained recording. The sessions of a database can be captured on different dates and sites involving different subjects.

The division is motivated by the lack of labelled data in the beginning of an annotation process, which usually does not allow to build models that are robust enough to generalise well to the unseen parts. This is especially true if the recording conditions and the involved subjects vary between the individual sessions. Nevertheless, already small fractions of labelled data can be sufficient to build models that are able to make reliable predictions on data that resembles the instances that have been seen so far. An example is data recorded from the same subject under comparable conditions—something we can generally expect from snapshots of the same session. Even if these models are too "weak" to make reliable predictions for the whole dataset, they can help to speed up the early annotation process. In the following, we refer to a classifier trained on samples of a single session as a *session-dependent* classifier. Once enough sessions have been completed, a *session-independent* model can be trained and used to accomplish remaining sessions.

To ensure the quality of the recognition, manual verification of the outcome of the classification might be necessary. This procedure can be accelerated by rating the predictions, e.g., by adding confidence values to the predicted instances. Instead of reviewing everything annotators can concentrate on parts with *low confidence*, i.e., labels that have been predicted with a high uncertainty.[2] The proposed strategy can be summarised as follows:

1. *Session Completion* Manually assign labels to a fraction of a session and train a session-dependent classifier. Apply it to complete the remaining fraction. Based on the confidence values generated by the model query manual revision.
2. *Session Transfer* Take all (with aid of step 1) fully labelled sessions and train a session-independent classifier. Apply it to predict annotations for remaining sessions. Again, based on the generated confidence values decide which parts require manual adjustment.

So far we have distinguished between session-dependent and session-independent classification. Depending on the corpus to which the strategy is applied, this may not necessarily be the best practise. For instance, if a dataset is composed of recordings that are too short to apply the first step we can adapt the strategy and initially complete recordings belonging to the same subject. Once we have labelled data from a sufficient number of individual subjects, we continue by training a subject-independent model and apply it to the remaining recordings. Likewise, we can use the described strategy across several databases, too. In that case we would concentrate on individual databases first and afterwards obtain a database-independent model that we use to label the remaining databases.

2.3 Implementation

To efficiently apply the described strategy, we would like to know the *sweet spot* for applying the *Session Completion* and the *Session Transfer* step. On the one hand, if we apply it too early the model becomes unstable and predictions will be poor. On the other hand, if we annotate more data than necessary we give away precious time. To avoid any of the described situations, we are interested in finding a good trade-off between machine performance and human effort. Unfortunately, we cannot easily guess what is the ideal moment to hand over the task to a machine. This is because the amount of training data that is required to build a robust model depends on a number of factors, such as the homogeneity of the data, the discrimination ability of the extracted features, the number of subjects and classes, and not least at the complexity of the recognition problem. Alternatively, instead of trying to determine a sweet spot beforehand (and possibly miss it), we could iteratively test

[2] In a multi-class classification task uncertainty can e.g., be derived from the distance a predicted sample has to the decision boundaries of the other classes.

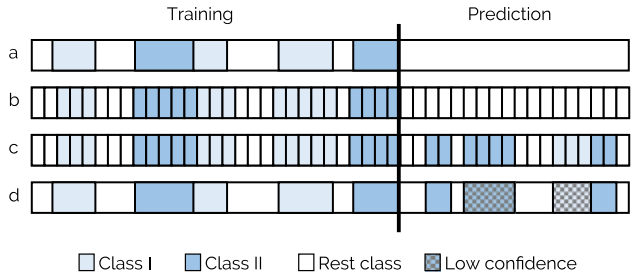

Fig. 2 Visualisation of the cooperative machine learning strategy by means of the SC step: (a) the end point of the last segment of the manual annotation defines where the training fraction ends and prediction begins (b) Labelled segments are mapped onto frames and empty frames are assigned to a rest class. (c) A model is build from the frames in the training fraction and used to predict the frames in the prediction fraction. (d) Successive frames with the same class label are combined, the rest class is removed and segments with a low confidence are highlighted

the applicability of the strategy and stop when the performance seems promising.

Therefore, we opted to make the described strategy an integral part of our tool (see Sect. 4). This allows annotators to visually examine the results at any time and to individually decide whether more labelling is required or not. However, this means that the time it takes to run the CML strategy becomes a crucial factor. Generally, it should not take longer than a few seconds or the annotation process will be interrupted (this is especially true for the session completion step). To reach this goal, we should reuse as much information as possible. One possibility is to apply classification on a small sliding window (frames) and use a rather simple (e.g., linear) classifier. The former means that features have to be extracted only once (or can be even pre-extracted); the latter ensures a fast training.

In the following, we describe the workflow for discrete annotations, i.e., we deal with multi-class problems. In case of the SC step we receive the raw signal stream (e.g., an audio signal) of the current session and a partly finished annotation composed of labelled segments with a discrete start and end point. The segments can be of variable length and there may be gaps between two successive segments. By applying the following procedure we then predict the segments for unlabelled fraction of the session (see also Fig. 2):

1. If not provided, extract frame-wise features for the whole session.
2. Find the frame that coincides with the end point of the last label in the annotation and split the feature sequence into a training fraction (preceding frames) and a prediction fraction (successive frames).
3. In the training set assign frames that overlap with a labelled segment by at least 50% to the corresponding class. In case of several candidates keep the dominant one (most overlap). Assign remaining frames to a rest class.
4. Learn a classifier using all frames from the training fraction.
5. Use the classifier to label the prediction fraction by assigning to each frame the class with the highest probability.
6. Combine successive frames belonging to the same class and keep the average probability of the combined frames as confidence. Remove frames that belong to the rest class. Optionally, apply thresholds to remove small segments and fill gaps.
7. Add the predicted segments to the original annotation and mark segments with a low confidence.

The ST step works in the same way with the difference that whole sessions are used to train the classifier, which is applied to predict whole sessions afterwards.

3 Evaluation

In this section we turn to some experiments in which we examine the practical effect of the proposed Cooperative Machine Learning (CML) strategies of Sect. 2. We do this by means of a database including natural human-human interaction and simulate a situation where the detection system is applied to predict unlabelled fractions of the dataset. Using the original and predicted parts of the corpus to train a final detection model we evaluate the robustness and efficiency of the CML approach.

3.1 Database and Problem Description

We first introduce the dataset we used for evaluating our approach. The NOvice eXpert Interaction (NOXI) database [8] is a corpus of screen-mediated face-to-face interactions that features natural interactions between human dyads in an expert-novice knowledge sharing context. In a session one participant assumes the role of an expert and the other participant the role of a novice. The corpus was created as a part of the ARIA-Valuspa [52] (Artificial Retrieval of Information Assistants—Virtual Agents with Linguistic Understanding, Social skills, and Personalised Aspects) project, and therefore has a strong focus on medial face-to-face interactions in an Expert-Novice setting. Figure 3 shows two users during interaction.

One purpose of NOXI is to study interruption strategies. For instance, when a listener decides to ask a question or comment to what the speaker was saying and therefore starts an attempt to take over the speech turn. The simplest way to detect such situations is by looking for spots where the voice of the two participants is overlapping. If afterwards a

Fig. 3 Snapshots of user interaction (left) and observer screen (right) during a recording session for the NOXI database

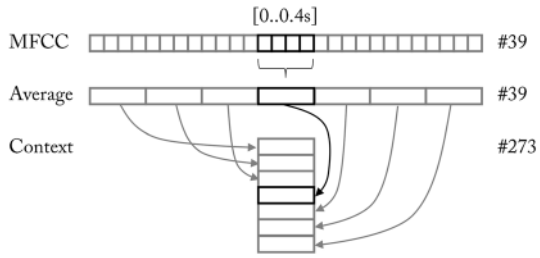

Fig. 4 Illustration of the feature extraction step. First, four MFCC frames with a dimension of 39 are averaged to reduce the sample rate of the signal to 25 Hz. Afterwards, neighbouring frames are added, here three frames from the left and three frames from the right. This results in a final feature vector of size 273

speaker change occurs we can assume that the interrupting party successfully took over the turn. Otherwise we can treat it as a failed attempt. However, an interposed utterance is not necessarily a signal to interrupt the speaker. It can also be an expression of approval or interest, denoted as *backchannels*. Likewise, not every speaker pause signals a floor change if, for instance, the speaker needs time to think what to say next. To bridge these pauses speakers usually utter a *filler* sound. Hence, to correctly identify speaker interruptions we have to separate backchannels and fillers from other speech parts.

In the following, we present a detection system that is trained to automatically identify backchannels and fillers in speech. First, we evaluate the system following a classic machine learning approach to measure the performance of the system. Afterwards, we examine if and to what extent the system is able to speed-up the manual annotation process in the CML loop.

3.2 Detection System

Though in our experiments we concentrate on the detection of speech and fillers/ backchannels, we opt for a detection system that is as generic as possible. This will allow us to apply it to other classification problems, too. Also, speed performance plays a crucial role as we do not want to interfere with the annotation process. In the following we start by describing the proposed generic detection system.

Due to its modularity and capability of fast online incremental processing we rely on the OpenSmile audio feature extraction tool [17]. However, we refrain from using a large statistical feature set like the ComParE (Computational Paralinguistics Evaluation) set, which assembles 6373 features by brute-force combination of Low Level Descriptors (LLDs) with Functionals [45]. This kind of feature sets are usually applied on chunks of several seconds length (e.g., a whole utterance). In our scenario, however, we opt for a frame-based feature set extracted over a small moving window that can be reused across successive training steps. Also, we should keep in mind that especially in the beginning of the annotation process the size of the training sets can be small. In that case a smaller feature set will lower the risk of overfitting.

Mel-Frequency Cepstral Coefficients (MFCC) provide a compact representation of the short-term power spectrum. Not only have they a long tradition in speech recognition systems [40] and speaker verification tasks [20], but have also been successfully applied in the field of social signal processing, e.g., emotional speech recognition [7, 31, 33, 38, 44, 56] and laughter detection [28, 32, 51]. For our tests, we calculate 13 Mel-Frequency Cepstral Coefficients (including the 0th coefficient) and their first- and second-order frame-to-frame difference (delta–delta).

According to standard practice we use a moving window of 25 ms with a frame step of 10 ms. Afterwards we reduce the stream to a frame step of 40 ms by averaging always four frames. This ensures that the sample rate of the feature stream is consistent with the video frame rate of 25 Hz. Though not relevant for the current study, it will be handy if we want to integrate visual features in the future. Yet, 40 ms are small enough to detect start and end point of voiced segments sufficiently accurate. Since the length of the filler events we want to detect may be longer than 40 ms, we optionally concatenate neighbouring frames from both sides of the current frame—in the following denoted as context size n. A context of size 3, e.g., means that the current frame is extended by three frames from the left and three frames from the right. This increases the number of features by a factor of $2 \cdot n + 1$. Figure 4 illustrates the feature extraction step.

As classification model we use a linear Support Vector Machine (SVM) provided by LIBLINEAR—a Library for Large Linear Classification [19]. Since the implementation does not use kernels, training time is significantly reduced even for large input sets composed of several ten thousand samples. For multi-class classification we select a L2-regularised logistic regression solver (option-s 0) and add a bias term of 0.1 (option-B 0.1). We keep default values for all other parameters. Since we expect unbalanced class

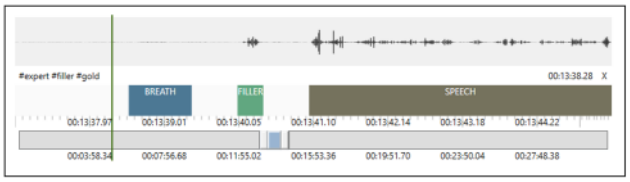

Fig. 5 Example of a manual annotation

distributions, we randomly remove samples to match the size of the class with the least number of samples. Finally, features values are scaled between -1 and 1 (when we test a sample we apply the scaling derived from the training set). Confidence values are scaled in a way such that individual class scores sum to 1.

3.3 Results

Having established a generic classification system we will now evaluate recognition performance on the NOXI corpus (see Sect. 3.1). We pick 18 sessions (German sub-corpus) and randomly split them into a training set including two-third of the sessions summing up to nearly 7 h of audio data. The remaining six sessions form the test set with an overall duration of almost 3.5 h.

To evaluate the proposed detection system we need to establish a ground truth. We use NOVA (will be introduced in Sect. 4) to manually annotate voiced parts in the audio files. To not introduce a machine bias none of the CML strategies described in Sect. 2 are applied. Manual annotation is accomplished by three experienced annotators, each completing six sessions. Table 1 lists the applied annotation scheme. Since labels are assigned to voiced sounds the remaining parts implicitly define the rest class SILENCE. Because of the better audio quality we use the head set recordings. However, it turned out that the close-talk recordings tended to pick up breathing sounds, so we introduce an additional BREATH class to prevent false alarms during silenced parts. Backchannels, fillers, laughter, and other voiced sounds such as grunts and coughs, are gathered in a single class denoted as FILLER. Speech segments that are neither backchannels nor fillers are labelled as SPEECH. An example of an annotation is shown in Fig. 5. We asked the raters to measure how long it took to annotate the sessions. In total they spent a little more than 14 h, which results in an average time of 47 min per session.

Next, we split the annotations in frames of 40 ms length and extract MFCC features, which results in 946,783 frames (exact class distribution are given in Table 1). We sample the training set down to 22,918 samples per class and train a linear SVM model. Results are summarised in Table 2. We report classwise recognition accuracy and Unweighted Average (UA) recall (average across classes). For a direct comparison with the INTERSPEECH 2013 Social Signals Paralinguistic Challenge we also consider the Area Under the Curve (AUC) measure. A 85% AUC for the FILLER class (best case) shows that results are comparable to Schuller et al. [45]. We take this as a prove that our detection system does a reasonable job on the examined task.

As seen in Table 2 increasing the number of concatenated frames has a positive effect on the recognition accuracy ($\sim$ 10%). Especially the FILLER class benefits from a larger frame context (25% improvement), which we explain with the fact that fillers are usually short and isolated speech episodes surrounded by silence. In Fig. 6 we notice a saturating effect for more than 10 frames, meaning we don't gain any additional improvements by adding more context. Also, we must not forget that in an online recognition system each additional frame that we look into the future introduces extra delay. For this reason, we decided to stick with a stacked context of 5 introducing a lag of 0.2 s, which we found still tolerable.

To give an impression how the system performs in terms of speed we report measurements on an Intel(R) Core(TM) i7-3930K. In our tests extracting MFCC-based features with a context of size 5 and a frame step of 0.04 s took 0.9 s for one minute of mono audio sampled at 48 kHz. Extrapolated to 10.5 h of interaction it requires less than 10 min to extract features for the whole German subset. Since features are reused this defines a one-time effort. Training a linear classifier on the training set (91,672 frames after class balancing) took on average 50 s. Frame-wise prediction on the test set (306,206 frames) only ~2.9 s. Such values suggest that the proposed detection system is fast enough to be embedded into the annotation process without causing serious interruptions (even if several hours of data are used as input/output).

3.4 CML Simulation

Finally, we want to know how the proposed detection system performs in combination with the proposed CML strategies. In Sect. 2 we have defined the *sweet spot* as the moment when additional annotation efforts no longer improve the stability of the classification model. Practically, this defines the ideal point to hand the task over to the machine. To experientially determine the sweet spot for the given problem, we incrementally inject information into the training process. In the following, we simulate this procedure by splitting the original training set into two parts: we assume that n sessions have been manually labelled (subset L), whereas the remaining sessions are yet unlabelled (subset U). Now, we derive three classifiers c, c' and c'' (see Fig. 7):

- c Train with the labels of L.
- c' Use c to predict the labels of U and retrain with the predicted labels.

Table 1 Annotation scheme and frame number per class

Class	Description	Train	%	Test	%
SPEECH	Speech (except filler and backchannels)	265,466	41.4	126,183	41.2
BREATH	Breathing (except unvoiced laughter)	22 918	3.6	3929	1.2
FILLER	Backchannels, fillers, laughter, and other voiced sounds	26,665	4.2	8592	2.8
SILENCE	Implicit rest class representing unvoiced parts	325,528	50.8	167,502	54.7
$\sum$		640,577		306,206	

Table 2 Classwise recall and area under the curve (in brackets) in % with respect to the context *n*

n	0	1	2	5	10	15
SPEECH	64.7 (95)	67.6 (96)	69.5 (96)	73.7 (97)	74.6 (97)	74.3 (97)
BREATH	82.5 (95)	84.3 (96)	85.1 (97)	87.2 (98)	87.9 (98)	88.2 (98)
FILLER	46.6 (69)	54.1 (74)	59.1 (77)	66.1 (82)	71.9 (84)	74.1 (85)
SILENCE	82.9 (92)	83.1 (93)	83.9 (94)	85.5 (95)	84.0 (96)	82.8 (96)
UA (UAAUC)	69.2 (88)	72.3 (90)	74.4 (91)	78.1 (93)	79.6 (94)	79.8 (94)

UA unweighted average, *UAAUC* UA of AUC

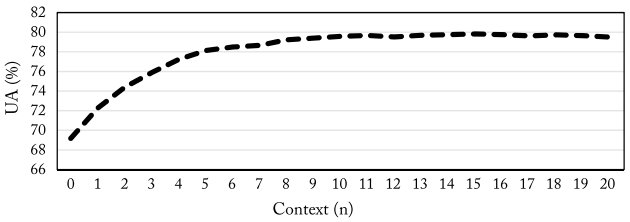

Fig. 6 Classwise UA recall in % with respect to the context size *n*

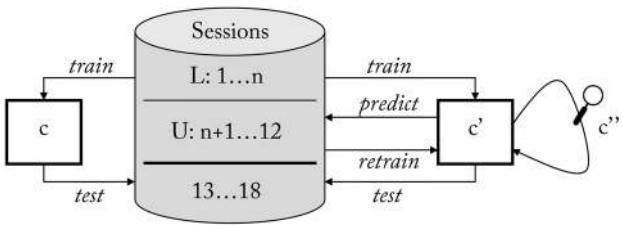

Fig. 7 In the default condition a classifier *c* is evaluated after training with labelled sessions (L) only. In case of c' unlabelled sessions (U) are predicted and used to retrain the model. And in case of c'' predicted labels are reviewed and possibly corrected before retraining takes place

c'' Before retraining inspect the predicted labels if their confidence is below a threshold *t* and correct them if necessary.

c' simulates the case where the annotation process is stopped at some point and the labelled fraction of the database is used to predict the remaining parts. Note that in this case *all* predicted labels are included during the final training step, i.e. no automatic selection strategies and no additional manual efforts are applied.

c'' simulates the case where parts of the prediction are inspected (here the selection is based on the class confidence). To assess the additional manual effort we measure what we call the *Inspection Rate* (IR), which is the fraction of frames below the confidence, and the *Correction Rate* (CR), which is the fraction of frames that are finally assigned a different label.

Table 3 summarises the performance of *c*, c' and c'' on the test set (the same as before). In each row we assume that *n* sessions of the original training set have been labelled (e.g., $n = 4$ means that *L* consists of sessions 1 to 4 and *U* consists of sessions 5–12). Based on the results we can gain some interesting insights. Let us therefore assume we aim for a classification model that is at maximum one percent worse than the reference model trained on all sessions, i.e. has an Unweighted Average (UA) recall of at least 77.1% (throughout the tests we have applied a stacking context of 5).

The performance of classifier *c* shows that ten of the twelve sessions are sufficient to yield a 77.8% recognition accuracy. Hence, to achieve our goal we can stop after labelling ten sessions and skip the last two. Now, what happens if we extend the training set with predicted labels (no selection or manually correction yet)? Checking the results of c' we see that again ten sessions are required to achieve the desired accuracy. In fact, extending the training set with purely predicted data generally has no positive effect on the recognition performance. Although disappointing at first glance this is actually not too surprising. Obviously we cannot expect to improve a model unless we inject some new knowledge, which is not the case if we add predictions without inspection. This is as if we asked a student to revise his own test, which is pointless unless we point out some of his mistakes first.

Table 3 Recognition results on the test when incrementally injecting information into the training process using the three classifiers c, c', c'' (see remarks in text)

	c	c'	c'' ($t = 0.5$)			c'' ($t = 0.75$)		
n	UA (%)	UA (%)	UA (%)	IR (%)	CR (%)	UA (%)	IR (%)	CR (%)
1	67.2	70.1	74.1	38	14	**77.4**	**87**	**25**
2	72.3	70.5	74.3	51	17	78.0	82	25
3	73.0	71.6	76.0	36	12	77.9	66	18
4	74.4	73.2	76.4	36	12	78.0	59	18
5	76.2	75.8	76.9	31	11	78.0	51	16
6	76.3	76.4	**77.1**	**27**	**9**	78.0	45	14
7	76.4	76.4	77.2	25	9	78.2	40	13
8	77.0	76.3	78.0	15	5	78.0	26	7
9	76.8	76.9	78.1	11	4	78.1	19	6
10	**77.8**	**77.8**	78.0	5	2	77.9	10	2
11	78.1	77.9	78.1	1	0	78.1	4	1
12	78.1	78.1	78.1	–	–	78.1	–	–

In case of c'' t defines the confidence threshold for inspecting predicted labels. In each row we start with n labelled sessions. Bold values refer to recognition rates that are comparable to the reference model (see remarks in text). Results are obtained with the detection system described earlier using a stacking context of 5

Hence, some manual efforts are needed here. And indeed: after correcting frames with a confidence below 0.5 (that is 9% of all frames in the remaining subset) c'' yields 77.1% already after 6 sessions. To achieve this we actually had to review 27% of predicted frames. If we assume that the remaining six sessions make up approximately half of the frames this corresponds to $\frac{1}{8}$ of the full training set, i.e. in total we have to examine $\frac{5}{8}$ ($= \frac{1}{2} + \frac{1}{8}$) of the training data. As mentioned earlier the average time to annotate a session was 47 min. Hence, we can reckon a saving of approximately 3.5 h (5.9 h instead of 9.4 h). Obviously, this significantly speeds up the annotation process.

Apparently, the more work we are willing to spend on the correction of predicted labels the earlier we receive a stable classification model. In fact, if we lift the correction threshold to 0.75 we observe that c'' now yields 77.4% already after the first session. However, this is achieved at the expense of a more than three times higher inspection rate (87%), which means that we have to view almost $\frac{9}{10}$ of the corpus (precisely $0.87\frac{9}{10} + \frac{1}{10}$). Hence, it can be a better strategy to complete a couple of sessions first and in return apply a smaller correction threshold afterwards leaving less data for inspection.

4 NOVA Tool

The results of the previous section encouraged us to integrate the proposed Cooperative Machine Learning (CML) approach into our annotation tool NOVA. This way we give annotators the possibility to immediately inspect and if necessary correct predicted annotations. Though an earlier version of the tool existed (see [6]), we extended it to achieve a seamless integration of the collaborative annotation process. NOVA is open-source and can be downloaded from http://github.com/hcmlab/nova.

4.1 Related Work

NOVA's interface has been inspired by existing annotation tools. For instance, EUDICO Linguistic Annotator (ELAN) [63], Annotation of Video and Language (ANVIL) [30], and EXMARALDA (Extensible Markup Language for Discourse Annotation) [43]. These tools offer layer-based tiers to insert time-anchored labelled segments, that is *discrete* annotations. *Continuous* annotations, on the other hand allow an observer to track the content of an audiovisual stimulus over time based on a continuous scale. A tool that allows labellers to trace emotional content in real-time on two dimensions (activation and evaluation) is FEELTRACE [12]. Its descendant GTRACE (General Trace) [13] allows the user to define their own dimensions and scales. Other tools to accomplish continuous descriptions are CARMA (Continuous Affect Rating and Media Annotation) [21] and DARMA (Dual Axis Rating and Media Annotation) [22]. An interesting approach for gathering crowd-sourced annotations is iHEARu-PLAY [23], that allows labelling audio material on a valence-arousal scale in form of a browser-game. Whereas most tools are restricted to describe audiovisual data by a single user, RepoVizz [37] is an integrated online system to collaboratively annotate streams of heterogeneous data (audio, video, motion capture, physiological signals, etc.). Datasets are stored in an online database, allowing users to interact with the data remotely through a web browser.

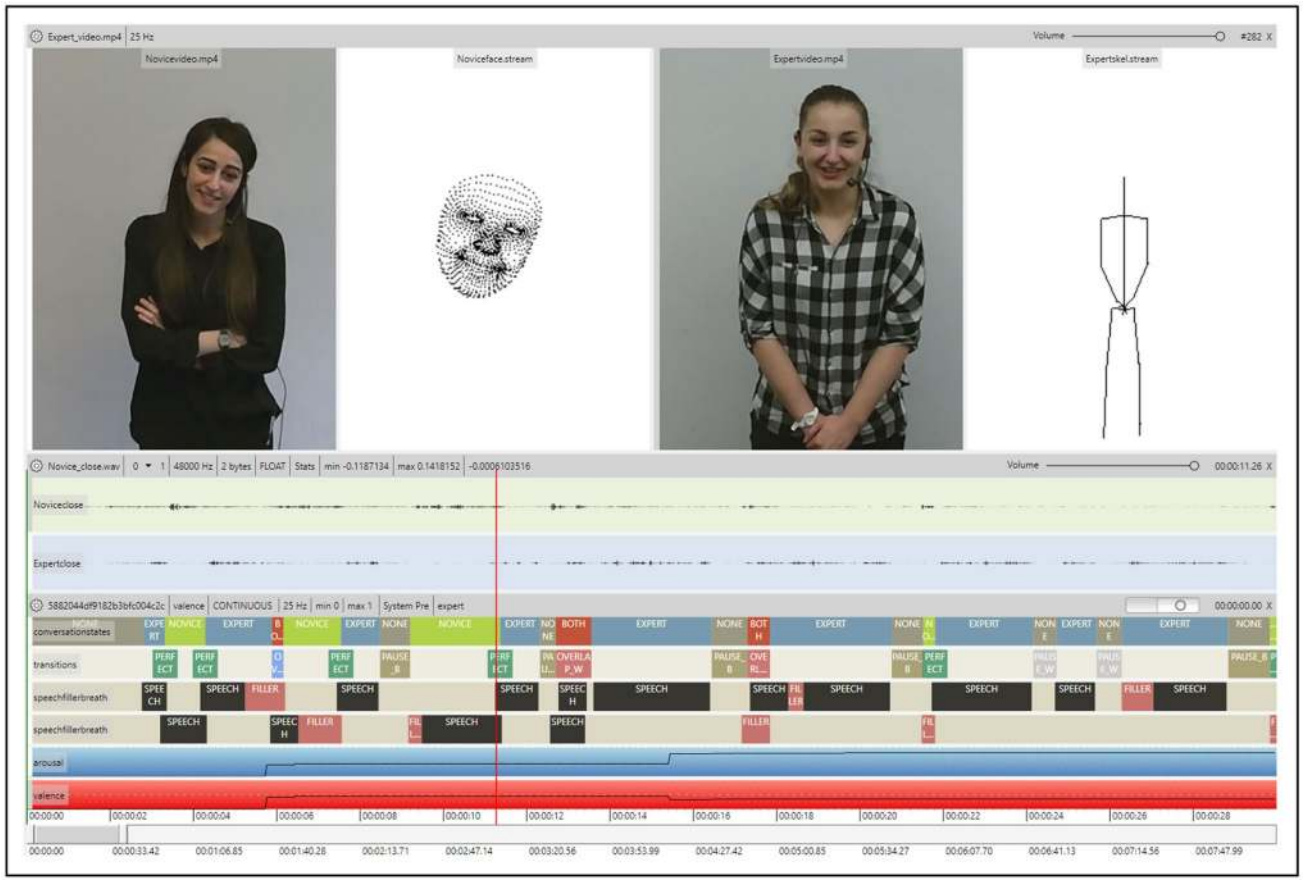

Fig. 8 NOVA allows it to visualise various media and signal types and supports different annotation schemes. From top down: full-body videos along with skeleton and face tracking, and audio streams of two persons during an interaction. In the lower part several discrete and continuous annotation tiers are displayed. Annotations can be edited on a static fraction of the recording or interactively during playback

Though the mentioned tools are of great help to create annotations at a high level of detail, the tools offer none or only little automation. Since labelling of several hours of interaction is an extremely time consuming task, methods to automate the coding process are highly desirable. To this end NOVA has been advanced with features to create collaborative annotations and to apply CML strategies out of the box (see Sect. 2). To support a truly collaborative workflow between several annotators and the machine a database back-end is provided to store, exchange, and combine annotation work.

4.2 General Interface

The NOVA user interface has been designed with a special focus on the annotation of long and continuous recordings involving multiple modalities and subjects. Unlike other annotation tools, the number of media files that can be displayed at the same is not limited and various types of signals (video, audio, facial features, skeleton, depth images, etc.) are supported. Further, multiple types of annotation schemes (discrete, continuous, transcriptions, geometric, etc.) can be selected to describe the visualised content (see Fig. 8 for an example). Several statistics are available to process the annotations created by multiple coders. For instance, statistical measures such as Cronbach's α [14] or Cohen's κ [11] can be applied to identify inter-rater agreement and annotations from multiple raters can be merged from the interface.

4.3 Annotation Types

The coding process of multi-modal data depends on the phenomenon we want to describe. For example, we would prefer a discrete annotation scheme to label behaviour that can be classified into a set of categories (e.g. head nods and head shakes), so that all annotators use the same "vocabulary", whereas variable dimensions like activation and evaluation are better handled on continuous tiers. For tasks like language transcriptions, which consist of hundreds of

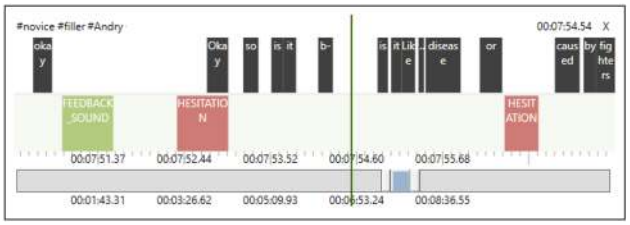

Fig. 9 Example of a discrete (bottom) and free (top) annotation tier. The start- and endpoint of a label can be directly changed with the mouse (even during playback). The name of a label can be changed through a dialogue by using pre-defined 'hot keys'

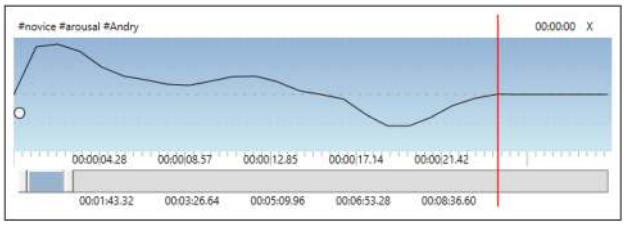

Fig. 10 Example of a continuous annotation tier. A value within a predefined range is assigned at a constant interval

individual words, we want to assign labels with free text. Finally, we might also want to annotate geometric points in visual material, for example if we want to learn about movements of the face.

To meet the different needs, NOVA supports four kinds of annotations:

1. *Discrete annotations* consist of a list of labelled time segments. Each segment has a start and end point and holds a label name. Segments can vary in length, may overlap and possibly have a gap to adjacent segments. Label names are not arbitrary but chosen from a set of predefined (*classes*).
2. *Free annotations* are similar to discrete annotations, but allow annotators to assign free label names. This is obviously useful if an annotation task can not easily be reduced to a few classes (for example in case of speech transcriptions). See Fig. 9 for an example of a free tier.
3. *Continuous annotations* are continuous in time and space. Instead of names, numerical values (*scores*) are assigned at a constant interval defined by a selectable sample rate.

   A live mode is available that allows annotators to interactively change the score values by moving the mouse/a gamepad or using the up and down keys to the desired level. See Fig. 10 for an example of a continuous tier. This is especially useful for regression tasks in machine learning, or for describing emotions and attitudes.
4. *Geometric annotations* are meant for annotation tasks where neither discrete nor continuous annotations are useful. Imagine we want to train a model to recognise facial landmarks, for example to calculate the FACS automatically.

4.4 Annotation Schemes

Each annotation type comes along with its own annotation scheme. For example, for discrete annotations a scheme contains information such as the annotation's name, the background colour of the tier and the labels allowed on the tier, respectively their colours. Once such a scheme is loaded, the annotator can only chose between these predefined labels. As described before, for FREE annotations, labels are not predefined and can be chosen freely during the coding process. Continuous and geometric schemes contain information such as the sample-rate, the minimum and maximum ranges, and for geometric annotations the number of points per frame. Using annotation schemes allows multiple annotators to create comparable annotations, and helps avoiding errors and misunderstandings.

4.5 Database Backend

To support a collaborative annotation process, NOVA maintains a database back-end, which allows users to load and save annotations from and to a MongoDB[3] running on a central server. This gives involved annotators the possibility to immediately commit changes and follow the annotation progress of the others. MongoDB is an open-source and cross-platform NoSQL database. We have chosen it in favour of a relational database due to its simplicity and fast read/write operations.

We opt for a design that not only allows to read and write annotations, but manages all relevant meta data of a corpus, too. Generally, each corpus is represented by a single database including several collections (the analogous to tables in relational databases). The collections are (see also Fig. 11):

- *Meta* Meta information about a database, including the data server location, and a description
- *Sessions* Stores general information for each recording session, such as location, language and date.
- *Annotators* Stores names and meta information of the involved annotators (human or machine!).
- *Roles* Stores the different roles subjects can take on during a recording session (e.g., listener vs speaker).
- *Streams* Stores the recorded stream files. Each file is assigned to a media type, a session, a subject and a role.

[3] https://www.mongodb.com/.

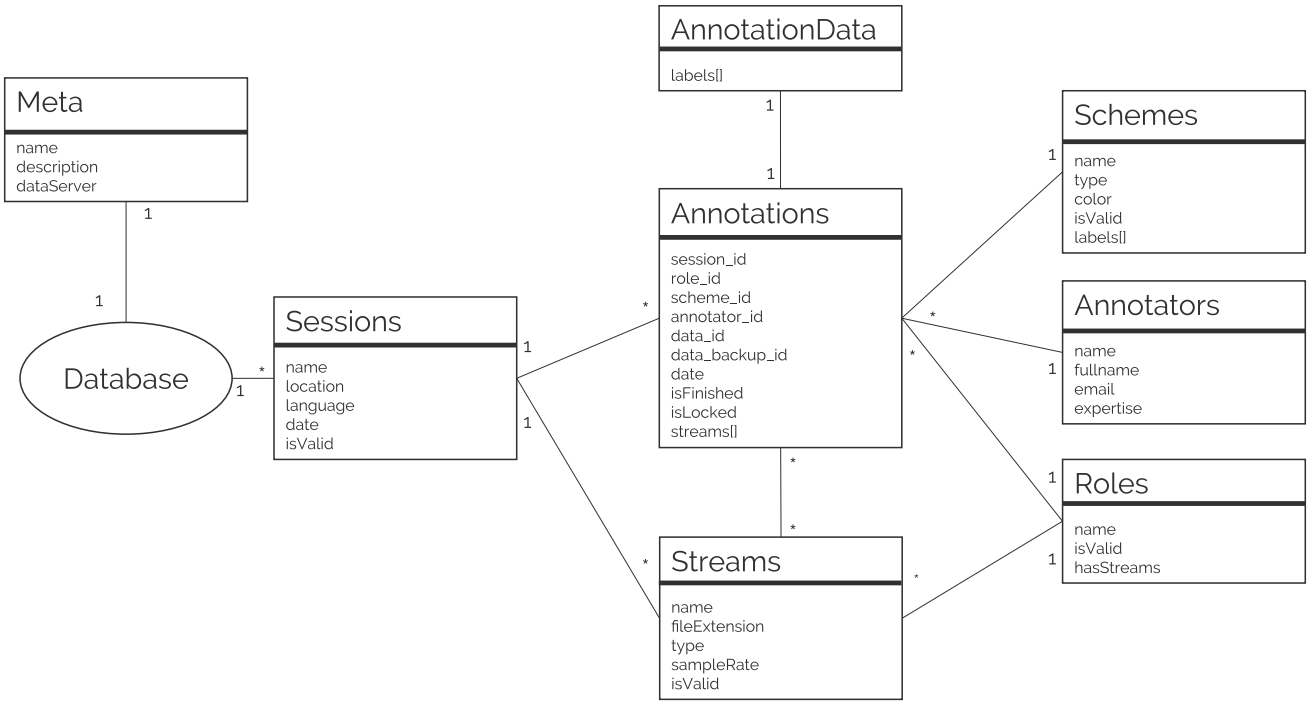

Fig. 11 Overview of NOVA's database structure. Annotations and meta information on subjects, sessions, etc. are stored in different collections. NOVA includes necessary tools to maintain and populate a database

An url is included that points to the location where the file can be downloaded.

- *Schemes* Stores the available annotation schemes.
- *Annotations* Stores the headers of created annotations. An annotation is linked to an annotator, an annotation scheme, a role and a session. Optionally, a list of stream files is referenced to store which information should be displayed during the annotation process.
- *AnnotationData* Contains the actual annotation data (segments or scores) for an annotation. Additionally a Backup is stored for each annotation, allowing the user to go back to the previous version.

As soon as several users collaborate on a common database it becomes crucial to implement adequate security policies. For instance, we want to prevent a situation in which a user accidentally overwrites the annotation of another user. Therefore, standard users can only edit and delete their own annotations. They can, however, load annotations of other users. In that case the annotation is copied and stored under their username. Only users, privileged with admin rights may edit and delete annotations of other users. They can also assign newly created annotations to specific users. This way, an admin can divide up forthcoming annotation tasks among the pool of annotators.

Beside human annotators, a database may also be visited by one or more "machine users". Just like a human operator they can create and access annotations. Hence, the database also functions as a mediator between human and machine. To control the annotation progress we have introduced a 'isFinished' flag that signals if an annotation requires further fitting or is finished. A second flag 'isLocked' marks whether an annotation is editable or not.

NOVA provides instruments to create and populate a database on a MongoDB server from scratch. This gives users the possibility to apply the tool on their own corpora. At any time new annotators, schemes and additional sessions can be added. No specific knowledge about databases is required.

4.6 ML Backend

For best possible performance tasks related to machine learning (ML) are outsourced and executed in a background process. As ML framework we use our open-source Social Signal Interpretation (SSI) framework.[4] SSI has been successfully applied to a couple of recognition problems in the past, see e.g., [34, 35, 51, 58, 59, 61]. Since SSI is primarily designed to build online recognition systems, a trained model can be directly used to detect social cues in real-time [60].

[4] http://openssi.net.

Though, SSI is developed in C++, it offers a simple XML interface to define feature extractors and classifiers. For instance, the definition of the MFCC features from Sect. 3.2 looks as follows:

```
<chain>
  <meta frameStep="10ms"
        rightContext="15ms" />
  <filter>
    <item create="OSPreemphasis" />
  </filter>
  <feature>
    <item create="OSMfccChain"
          option="mfccdd" />
  </feature>
</chain>
```

When applied to a stream, the signal values are first run through a pre-emphasis filter before MFCC features are extracted over a sliding window of 25 ms with a frame step of 10 ms (timings can be overwritten in NOVA). To configure the MFCC extraction (e.g., the number of coefficients) a separate option file is created (here 'mfccdd'). However, SSI supports other features sets, too. For instance, it allows to run scripts from the widely used OpenSmile toolkit [17]. And it provides feature sets for other type of signals. For instance, a wrapper for the OpenFace tool [5] is available to extract of facial points and action units from video streams.

Likewise, the classification model from Sect. 3.2 is defined as follows:

```
<trainer>
  <meta balance="under" />
  <normalize>
    <item method="Scale" />
  </normalize>
  <model create="LibLinear"
         option="linsvm" />
</trainer>
```

Here, SSI is configured to balance the number of class samples by removing samples from overrepresented classes and scale features into a common interval. As training model a linear SVM will be used. However, SSI also supports a Python interface for using other classification models as well, e.g. Google's neural network framework TensorFlow,[5] as well as native implementations in C/C++. However, as we argumented before, we suggest to use fast classifiers because for users interacting with such a system, latency is a crucial aspect as we deal with short training iterations.

4.7 Basic CML Walk-through

We will conclude this section with a walk-through that demonstrates NOVA's CML tools. In this section we will go through the CML workflow, and turn to the extensions of eXplainable CML in the next section. We assume that a database has been created and populated with several sessions of audio recordings from one or more users. In our case, we work on the NOXI [8] and apply an annotation scheme containing the labels BREATH, FILLER and SPEECH. Note that number and names of the classes is defined by the underlying annotation scheme, which be easily adapted by the user to fit any other labelling problem.

As a first step, we extract MFCC features for the German sessions in the NOXI database. The dialogue is shown in Fig. 12. It allows us to choose a source stream and a feature extraction method (only methods that can be applied to the selected stream will be listed). Optionally, we can overwrite the default frame step and context sizes. Extraction can be accelerated by running several sessions in parallel (here 8).

In a next step, we can now pick an annotation scheme and apply it to the previously extracted feature streams. Figure 13 shows the interface that allows us to select the input and choose a classification model (only models are shown that fit the selected input). Optionally, we can set a left and right context to concatenate neighbouring feature frames (see Sect. 3.2). Afterwards the trained model is stored and can now be applied to predict unlabelled data.

To predict annotations, both CML strategies from Sect. 2.2 are available. In case of *Session Transfer* a dialogue similar to the one in Fig. 14 is shown. However, this time we select a previously trained model and use it to predict the selected sessions. In case of the *Session Completion* step, the annotation is completed by temporarily training a model using only the labels available from current tier. An example before and after the completion is shown in Fig. 14. The screenshot shows that labels with a low confidence are highlighted with a pattern. This way crucial parts are quickly found and can be revised if necessary.

To assess the prediction accuracy of a model, a dialogue similar to Fig. 13 is available. Here, we can pick a trained model and the sessions we want to use for evaluation (only sessions with an according annotation are listed). The model is now applied to predict labels for the selected sessions and the output is compared to the existing annotations. The result is presented in form of a confusion matrix as shown in Fig. 15. A confusion matrix provides information on the overall recognition performance, as well as, accuracies for individual classes and which class pairs are often confused. Note that this feature is only available for the Session

[5] https://www.tensorflow.org/.

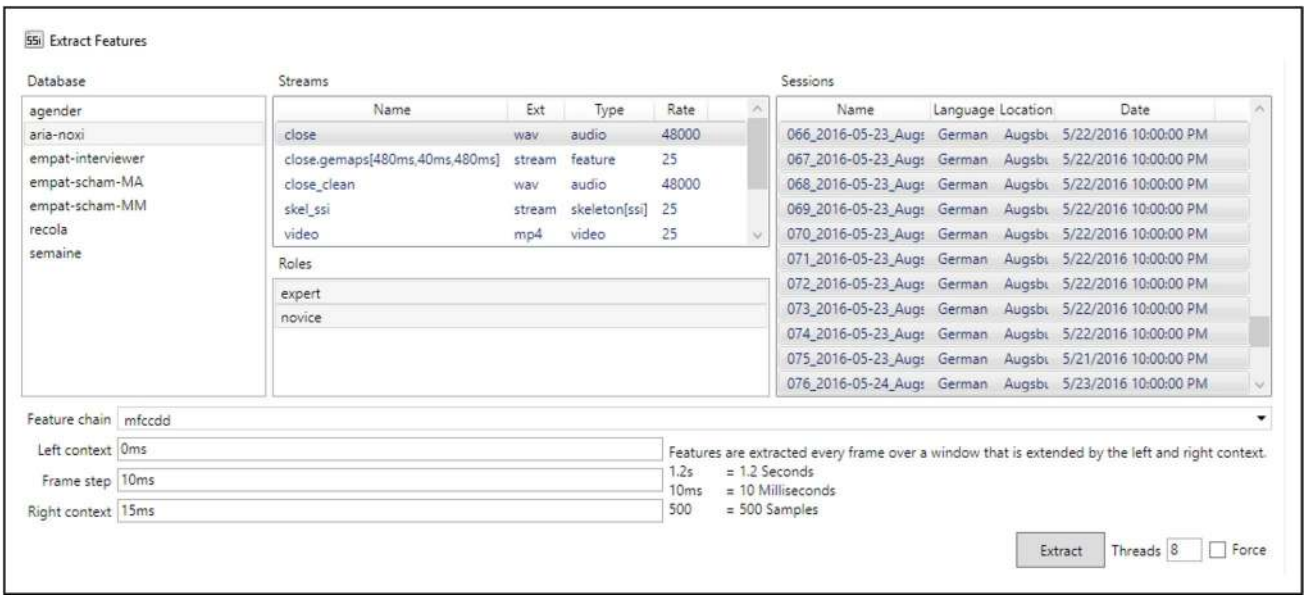

Fig. 12 Screenshot of the feature extraction dialogue. The user chooses a stream (here audio) and an according feature extraction method (here mfccdd). Feature extraction is applied for the selected roles and sessions

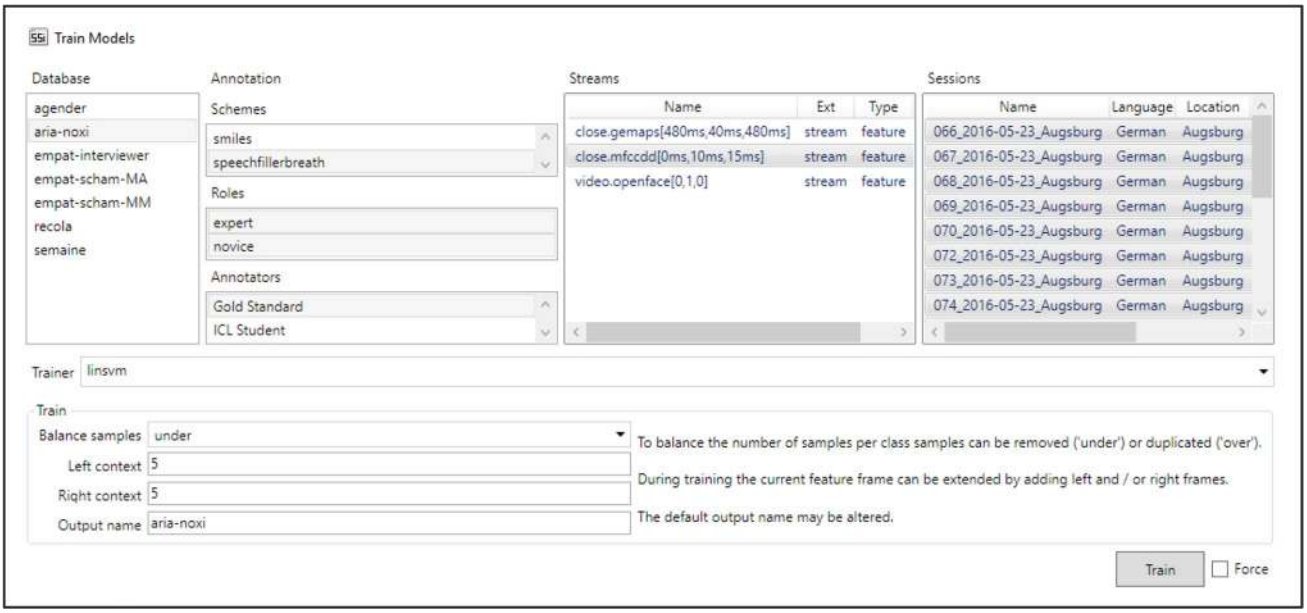

Fig. 13 Screenshot of the model training dialogue. The user selects a coding scheme, a role and an annotator (here Gold Standard). Sessions for which an according annotation exists are now displayed and a stream can be selected to define the input for the learning step. Finally, a model (here linsvm) is chosen and the training begins

Transfer step, respectively a classical Supervised Learning approach, as we need to have ground truth labels to compare our model with.

5 eXplainable AI (XAI) Extension

The application of CML provides the possibility to reduce the time needed for annotating while sustaining satisfactory classification accuracies. However not only

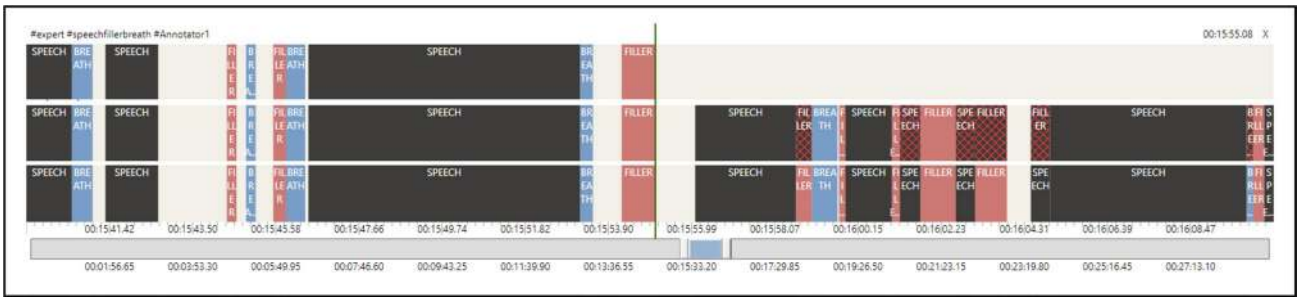

Fig. 14 Visualisation of partly finished annotation (upper tier) and the results after the tier is automatically completed (middle tier). Segments with a low confidence are marked with a red pattern. The lower tier shows the final result after manual correction

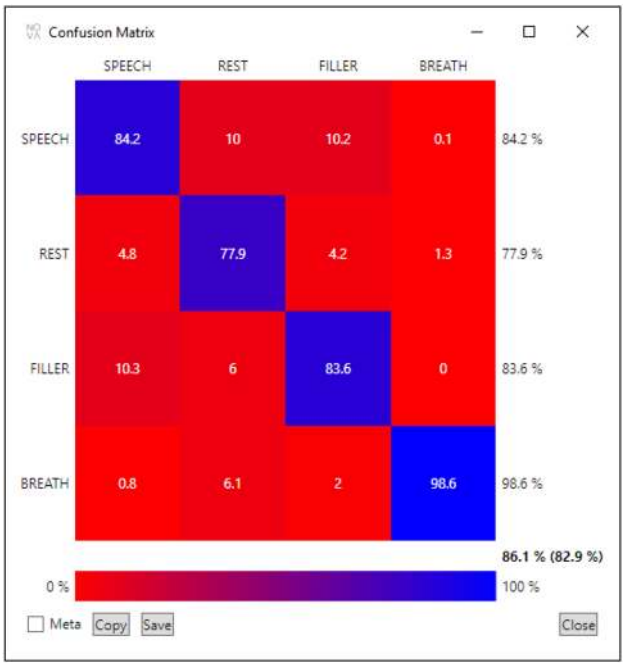

Fig. 15 A confusion matrix provides information about the recognition accuracy of individual classes and to what extent they are confused with other classes. For instance, here we see that speech frames are often falsely classified as fillers and vice versa. Hence, an annotator should put attention to these classes while revising the predictions. The REST class implicitly represents *silence* in this example

needed time and accuracy scores are important measures in modern machine learning, but also comprehensibility and transparency. The strive of making machine learning models more comprehensible for humans by providing explanations goes back to as early as the 1970s. Back then Shortliffe and Buchanan were stressing the need for explanations in rule-based expert systems [48]. During the CML step we introduced confidence values for the labels predicted by the model. Low confidence values highlight sections where the model is uncertain. Those values provide the user with a basic tool to gain insight about instances that the model has issues to correctly classify. However, to provide the user with a comprehensible machine learning experience we extended NOVA with the two explanation frameworks LIME [41] and iNNvestigate [1]. LIME is capable of providing explanations for various problem domains like text and image classification. The basic idea is to approximate an interpretable model around the original model. Their explanations come in the form of visual feedback, highlighting the sections that have been crucial for the prediction of a specific class. They showed that with the help of LIME it is easier for users to determine from a set of classifiers which one performs best for a given problem domain. This is especially useful when test-accuracy scores themselves are misleading. Moreover, they argue that LIME not only is useful for gaining additional insight about a model, but also users have been able to improve performance of classifiers by identifying unnecessary features and removing them based on the explanations generated by LIME. INNvestigate is a library that provides implementations of common analysis methods for neural networks, e.g. PatternNet and LRP. This extension allows an in-depth analysis of predictions with the help of visual explanations. The possibility to generate explanations can be beneficial for several use cases. In general whenever a model's prediction is wrong you can not only examine the prediction scores, but also take a visual explanation into account that has been generated by exploring the features most important for the classification. Moreover, this is not only the case for misclassifications. Explanations can also help to gain additional information when there are serious doubts on what the model really has learned. With the help of their explanation framework, Ribeiro et al. revealed in [41] that correct predictions are not necessarily based on semantic correct correlations.

In the previous sections we demonstrated that NOVA provides the possibility to complete unfinished annotations automatically and highlight uncertain predictions with a confidence score. The XAI extension allows now to further investigate those particular spots and gain additional insight

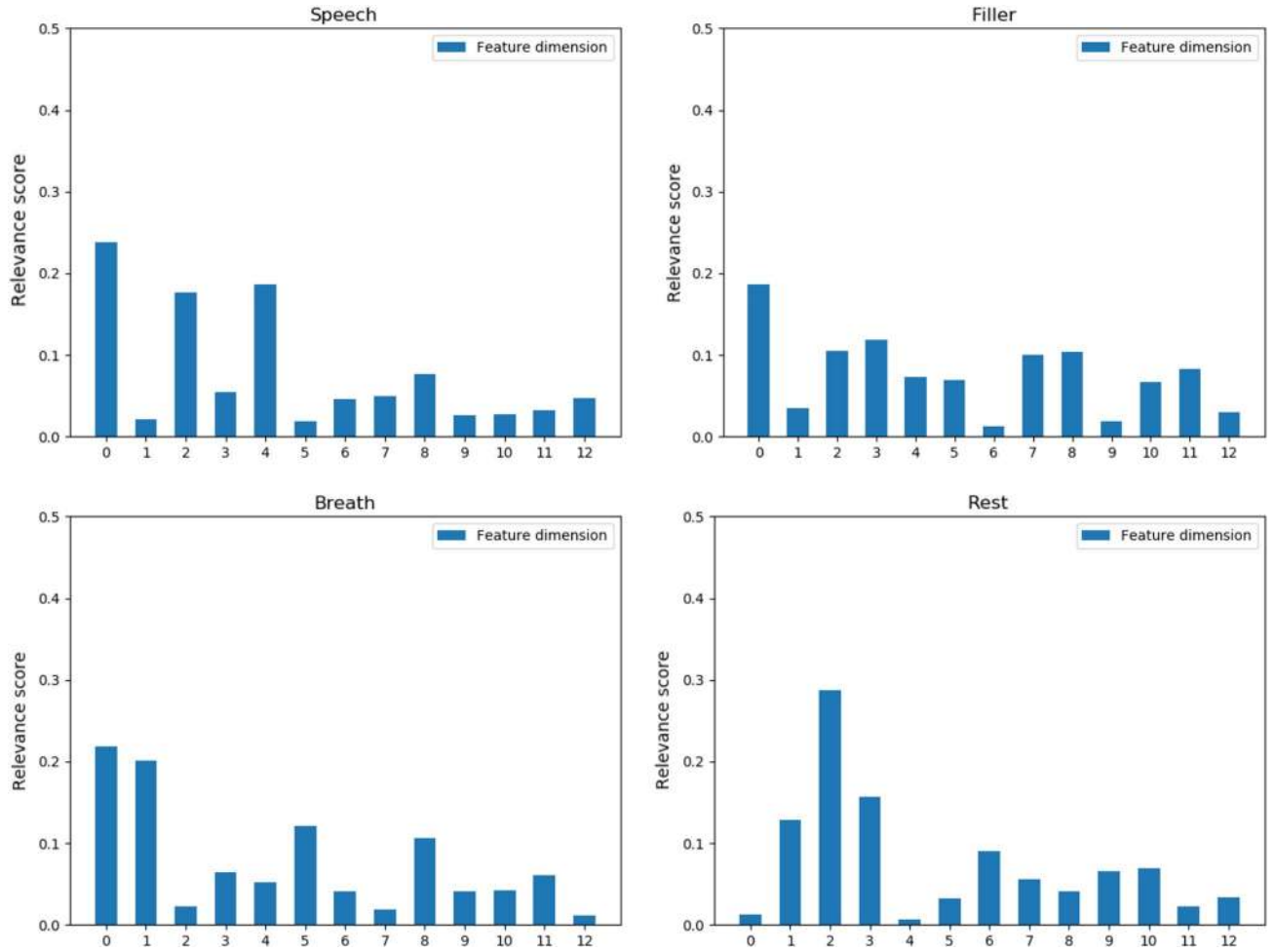

Fig. 16 Explanations for exemplary instances for the four classes speech, filler, breath, rest

on the classifier's decision making. For example, Fig. 16 shows the relevance scores for every feature dimension of the 13 MFCC features calculated for our previous classification use case. In particular the figure displays explanations generated for exemplary instances of the four classes, SPEECH, FILLER, BREATH and REST (Silence). The relevance score describes the importance of a specific feature in regard to the classification. In our use case the model was trained on Mel-Frequency Cepstral Coefficients which provide a compact representation of the short-term power spectrum of an audio signal.

Further, explainable AI techniques are not only of use to visualise the relevance of particular features, but may also be used to explain more sophisticated models. In particular, NOVA is able to provide visual explanations of Deep Learning models, such as convolutional neural networks that are trained on raw data. In order to describe the explanation capabilities of NOVA, we will turn to a visual classification task in the following example.

Fig. 17 Explanations for the top four classes generated in NOVA with the usage of LIME

Figure 17 shows an Instance of the NOVA interface for generating explanations with LIME. In the present example we demonstrate a use-case for visually categorising images of faces into base emotions. Explanations for the top four predicted classes are given, however, the number of considered classes may be changed by the user. Moreover, coherent with LIME, additional options can be altered like the number of samples or the number of features. Furthermore, for the generation of explanations the user can either choose from a list of models that have been trained with the help of NOVA for the given modality or drag and drop models from a different source.

In the displayed case in Fig. 17 the predicted top class has been happy, followed by anger, neutral and surprise. The green shapes (so called super-pixels) represent areas of the original image that have been important for the prediction. In contrast to that the red shapes describe areas that spoke against a particular prediction. As one would intuitively guess, an interesting area for recognizing whether a person is happy, is the space around the mouth to see if the person is smiling. Moreover, the same area is a strong evidence against the presence of anger, neutral and surprise, which is highlighted by a red area in the other images. Despite the fact that the used model predicted the correct class with an accuracy of 90.5% there is evidence in the explanations present that the model still has flaws. The fact that various areas of the background have been considered important for the prediction, even though there is no relevant information visible, shows that the model isn't perfectly optimized for the given use case.

Alongside the explanation generated by LIME, NOVA also offers the possibility to create explanations with iNNvestigate. The corresponding NOVA interface not only provides a variety of algorithms implemented in iNNvestigate, but also allows the user to decide between different visualization representations. Figure 18 displays an excerpt of some algorithms and visualizations for different facial expressions. The class that has been predicted by the model is written above of the original images. For Fig. 18 A all algorithms highlighted the central area of the face—including the eyes, nose and mouth—as important elements regarding the prediction. In case of the angry face (Fig. 18b) the visualizations 4 and 5 show a stronger emphasis on the forehead and eyebrow area which is what would be expected as the bending of the eyebrows is a common indicator for anger. Similar is true for Fig. 18c here especially visualization 2 and 3 highlight amongst other areas the forehead which displays an intensely furrowed brow. Before covering the last facial expression we want to emphasize the fact that similar to LIME, the algorithms used in 4 and 5 all highlighted to some degree areas in the background of the original image, which corroborates the hypotheses that the model isn't fully optimized and bases the prediction to some extent on irrelevant information.

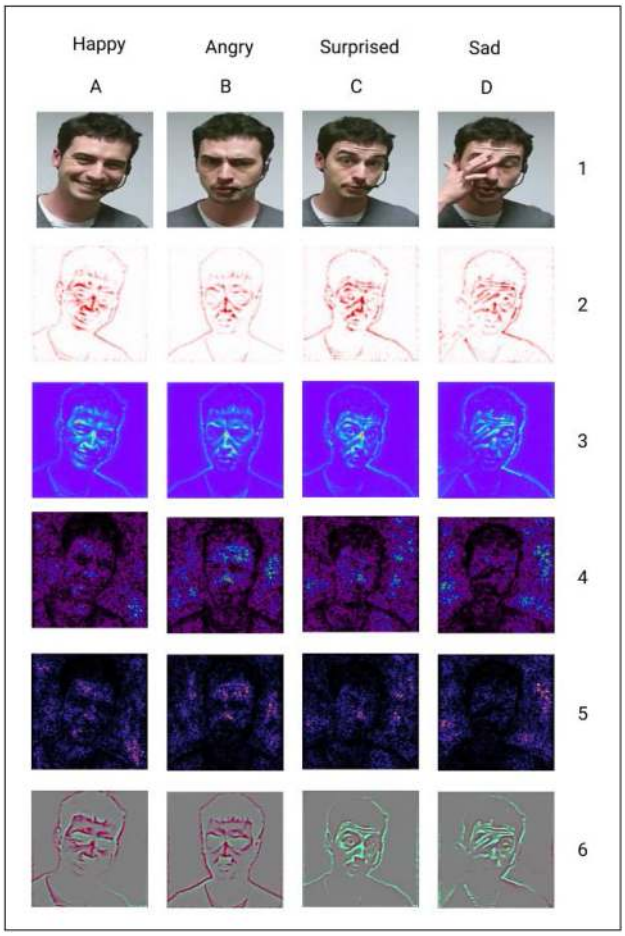

Fig. 18 Visual explanations generated with the iNNvestigate framework. Letters A to D represent the different predicted emotions noted above. The numbers on the right map onto the following approaches: 1, original image; 2, guided backpropagation; 3, deep Taylor; 4, LRP Epsilon; 5, LRP Z; 6, LRP Alpha Beta

Figure 18d displays an interesting case in terms of prediction and generated explanation. Just by visually exploring the image one could easily agree that the person is sad because he just might have shed a tear and is trying to wipe it away with his hand. Also the explanations generated by the different algorithms stress the areas covering the hand and eyes. However, if one would examine the moment short before and after the specific frame it would become obvious that the person has not been sad at all and probably has just rubbed its eye. This way it becomes evident that for a correct interpretation it is vital to also consider context information. NOVA offers, besides the generation of explanations through state of the art algorithms, the possibility to investigate relevant information before and after a specific frame being part of a video or feature stream.

Figure 19 shows a possible setup when working with NOVA. In the presented screenshot an annotation and the corresponding video is loaded. The frame of interest is

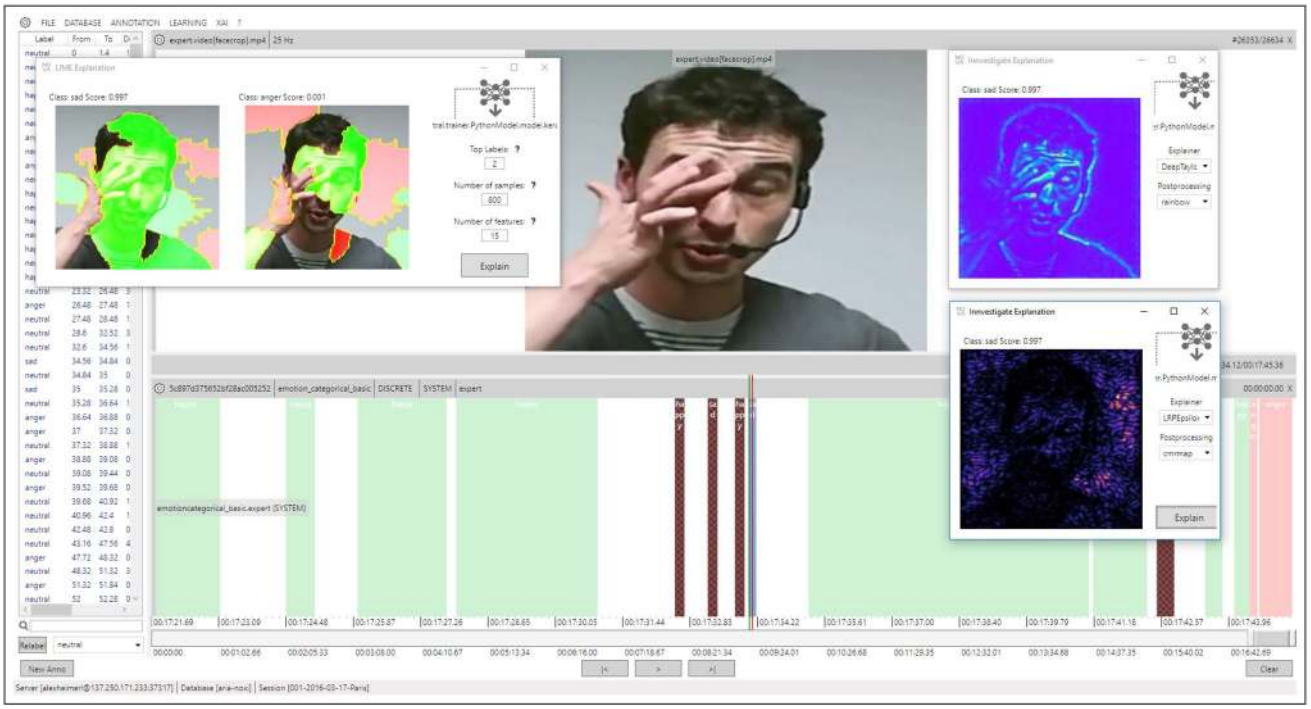

Fig. 19 An instance of the NOVA user interface with visual explanations for a particular frame

the earlier discussed alleged sad facial expression. To gain additional insight explanations with LIME and iNNvestigate have been generated.

6 Discussion and Outlook

In Sect. 3 we have presented a technical evaluation of our proposed Cooperative Machine Learning (CML) strategy. Results show that CML bears great potential to significantly reduce human labelling effort. However, it does not necessarily mean that results gained in a simulation can be transferred to human labellers without further ado. Hence, in the following, we want to discuss the experiences of users who have been applying the CML strategies with the NOVA tool.

So, what exact gain can we expect when giving a tool like NOVA into the hands of human labellers? Unfortunately, a general answer to this question does probably not exist. Our experiences show that the amount of time we may save depends on a couple of variables, which may vary from one case to another.

Probably, the largest uncertainty comes from the nature of the annotation problem itself and the ability of the applied machine learning (ML) techniques to cope with it. For instance, let us assume the task of labelling voiced parts in audio. If the recordings have low background noise and speech is really the only prominent signal, a simple feature like loudness may already allow us to train a robust model on few samples, yet generalizing well on unseen data. In this case, the time saving (compared to a completely manual approach) can be tremendous. On the other hand, if the speech files are noisy and contain other audible sounds—possibly overlapping with speech—the problem becomes immediately harder. As a consequence not only a more sophisticated feature set (and classification model) is needed, but more manual labelling effort is required to obtain a robust model. As a consequence, less time is saved. And we may even reckon the case where the problems becomes too hard to train a reliable model at all, so that the effort to manually revise the prediction may eat up initial savings. The possibility to exchange features and classifiers in NOVA is therefore an essential precondition to adapt to the problem at hand in the best possible way.

Another point to consider is the quality of the annotation that is desired. Can we live with some false prediction? Or do we aim for a high precision, yet do not mind a high number of false negatives? This, of course, depends very much on the purpose the data is labelled for. As a special flaw social signals often lack a ground truth. And when multiple raters are employed the agreement often turns out to be low. This makes it especially difficult to estimate the quality of a prediction. In the end, it depends a lot on the assessment of the user if he or she is pleased with the automatic completion. Here, NOVA's feature to immediately visualise the results is an important tool to let raters assess the quality of automatic predictions. Further the integrated XAI techniques offer a more transparent insight in the model's reliability.

Finally, comparing manual with semi-manual annotations is not as straight forward as it may seem. When observing automatic predictions we observed that on- and offset of the labels were often more precise than that of humans, which are usually rather fuzzy (unless they work at a very fine granular time scale, which is usually too time-consuming). Likewise, we found that short occurrences of a behaviour are easily overlooked by human labellers, especially as their attention drops with time. Hence, since machines show no signs of fatigue their predictions are often more consistent throughout a corpus compared to those of humans. Consequently, applying CML strategies may not just help saving time, but also lead to more accurate and consistent annotations.

The core idea behind cooperative machine learning (CML) is to create a loop, in which humans start solving a task (here labelling social signals) and over time a machine learns to automatically complete the task. In conventional approaches, this involves at least two parties: an end-user, who has knowledge about the domain, and a machine learning practitioner, who can cope with the learning system. However, to make the process more rapid and focused, Amershi et al. [2] demand that more control should be given to the end-user. To this end, our tool combines a traditional annotation interface with CML functions that can be applied out of the box requiring no knowledge on machine learning. We found it important to give coders the possibility to individually decide when and how to use them in the labelling process. And to assess the reliability of automatic predictions immediate visual feedback is provided, which gives annotators the chance to adapt their strategies at times. By interactively guiding and improving automatic predictions, an efficient integration of human expert knowledge and rapid mechanical computation is achieved. The reported experiments show that even end-users with little or no background in machine learning are able to benefit from the described machine-aided techniques.

We also observed that CML strategies not only have the potential to speed up coding, but can also have a positive influence on the annotator's coding style. Because of the preciseness machine-aided techniques introduce into the coding process the level-of-detail is improved while at the same time human efforts are reduced. Here, strategies to guide the attention of the annotator during inspection of the predicted labels become a crucial matter. As mentioned before Rosenthal et al. [42] investigated which kind of information should be provided to the user to minimise annotation errors. However, in their studies they concentrate on single images whereas in our case we deal with continuous recordings. To not overload the annotator with too many details we decided to uniformly highlight labels below an adjustable confidence threshold. Our simulations in Sect. 3 suggest that this approach helps to significantly reduce labelling efforts. However, the exact gain depends highly on the nature and complexity of annotation problem, the applied machine learning techniques, and not least the expertise and subjective attitude of the human coder.

7 Conclusion

The goal of the presented work is to foster the application of *Cooperative Machine Learning* (CML) strategies to speed up annotation of social signals in large multi-modal databases and additionally to give the user a better understanding of the trained models by incorporating explainable AI techniques. Well described corpora that are rich of human behaviour are needed in a number of disciplines, such as Social Signal Processing and Behavioural Psychology. However, populating captured user data with adequate descriptions can be an extremely exhausting and time-consuming task. To this end, we have presented strategies and tools to distribute annotation tasks among multiple human raters (to bundle as much human efforts as possible) and automatically complete unfinished fractions of a database (to reduce human efforts where possible).

In particular, we have proposed a two-fold CML strategy to support the manual coding process (Sect. 2). Applied to a fresh database it first concentrates on completing few individual sessions. A relatively small amount of labels is sufficient to build a session-dependent model, which—though not strong enough to generalise well across the whole database—can be used to derive local predictions. Afterwards, a session-independent classification model is created to finish the remaining parts of the database. During both steps, confidence values are created to guide the inspection of the predictions.

To prove the usefulness of the CML approach, we have presented results for a realistic use-case based on a database featuring natural interactions between human dyads. For our experiments in Sect. 3 we picked the task of detecting fillers in speech. Fillers are an important cue if one aims to study turn taking and interruption strategies. A fast and general audio detection system in combination with a linear classification model has been applied to more than 10 h of natural conversations yielding an average recognition performance of almost 80% (four classes: speech, breath, filler and silence). In a simulation we proved that labelling efforts can be significantly reduced using the proposed system. If applied in combination with a revision of instances with a low confidence value, manual inspection was reduced to $\frac{5}{8}$ of the database. In our case, this corresponds to a saving of approximately 2.5 h (4.1 h instead of 6.6 h).

It was important to us to bring the proposed approach into application. To this end, in Sect. 4 we introduced NOVA[6]—an

[6] http://github.com/hcmlab/nova.

open-source tool for collaborative and machine-aided labelling. Other than conventional annotation tools NOVA supports a fully collaborative workflow and allows it to distribute annotation tasks among multiple raters. The discussed CML strategies have been integrated and can be directly applied from the interface to speed up manual annotation. The generalisability of the proposed detection system will enable other researchers to adopt the approach for their own databases and annotation tasks in the future.

Further, we described how the proposed CML workflow in the NOVA tool can be extended in terms of transparency and comprehensibility by introducing explainable AI techniques into the cooperative machine learning workflow. We subsume this overall process as "explainable Cooperative Machine Learning".

In our future work, we also plan to extent the current workflow by automatically generating recommendations in which order sessions in a database should be processed. Poignant et al. [39] suggest the use of hierarchical clustering to select prototypical examples and prioritise them during the coding process. However, it is not straightforward to adapt their techniques to continuous recordings. Alternatively, in our case we can make use of the confidence values generated during label prediction. Using the average value the following strategy is conceivable: every time a session is finished, a model is built to predict remaining sessions and pick the one with the lowest score to complete next. This way we ensure that manual efforts get spent on data that has a high potential to improve the learner in the next iteration.

Acknowledgements Open Access funding provided by Projekt DEAL. This work has received funding DFG under Project Number 392401413, DEEP.

References

1. Alber M, Lapuschkin S, Seegerer P, Hägele M, Schütt KT, Montavon G, Samek W, Müller K, Dähne S, Kindermans P (2018) Investigate neural networks! CoRR. arXiv:abs/1808.04260
2. Amershi S, Cakmak M, Knox WB, Kulesza T (2014) Power to the people: the role of humans in interactive machine learning. AI Mag 35(4):105–120
3. Amershi S, Chickering M, Drucker SM, Lee B, Simard P, Suh J (2015) Modeltracker: redesigning performance analysis tools for machine learning. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems. ACM, pp 337–346
4. Amershi S, Fogarty J, Kapoor A, Tan DS (2009) Overview based example selection in end user interactive concept learning. In: Proceedings of the 22nd annual ACM symposium on user interface software and technology, Victoria, October 4–7, 2009, pp 247–256
5. Baltrušaitis T, Robinson P, Morency LP (2016) Openface: an open source facial behavior analysis toolkit. In: 2016 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1–10
6. Baur T, Mehlmann G, Damian I, Lingenfelser F, Wagner J, Lugrin B, André E, Gebhard P (2015) Context-aware automated analysis and annotation of social human–agent interactions. ACM Trans Interact Intell Syst (TiiS) 5(2):11
7. Beritelli F, Casale S, Russo A, Serrano S, Ettorre D (2006) Speech emotion recognition using MFCCs extracted from a mobile terminal based on ETSI front end. In: International conference on signal processing, vol. 2
8. Cafaro A, Wagner J, Baur T, Dermouche S, Torres Torres M, Pelachaud C, André E, Valstar MF (2017) The noxi database: multimodal recordings of mediated novice–expert interactions. In: Proceedings of the 19th international conference on multimodal interaction. ACM **(in press)**
9. Chen NC, Kocielnik R, Drouhard M, Peña-Araya V, Suh J, Cen K, Zheng X, Aragon CR (2016) Challenges of applying machine learning to qualitative coding. In: CHI 2016 workshop on human centred machine learning
10. Cheng J, Bernstein MS (2015) Flock: hybrid crowd-machine learning classifiers. In: Proceedings of the 18th ACM conference on computer supported cooperative work and social computing. ACM, pp 600–611
11. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46
12. Cowie R, Douglas-Cowie E, Savvidou S, McMahon E, Sawey M, Schröder M (2000) 'feeltrace': an instrument for recording perceived emotion in real time. In: ISCA tutorial and research workshop (ITRW) on speech and emotion
13. Cowie R, McKeown G, Douglas-Cowie E (2012) Tracing emotion: an overview. Int J Synth Emot (IJSE) 3(1):1–17
14. Cronbach LJ (1951) Coefficient alpha and the internal structure of tests. Psychometrika 16(3):297–334
15. Dong M, Sun Z (2003) On human machine cooperative learning control. In: Proceedings of the 2003 IEEE international symposium on intelligent control, pp 81–86
16. Douglas-Cowie E, Campbell N, Cowie R, Roach P (2003) Emotional speech: towards a new generation of databases. Speech Commun 40(c):33–60
17. Eyben F, Weninger F, Gross F, Schuller B (2013) Recent developments in opensmile, the Munich open-source multimedia feature extractor. In: Proceedings of the 21st ACM international conference on multimedia, MM '13. ACM, New York, pp 835–838
18. Fails JA, Olsen Jr, DR (2003) Interactive machine learning. In: Proceedings of the 8th international conference on intelligent user interfaces, IUI '03. ACM, New York, pp 39–45
19. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874
20. Ganchev T, Fakotakis N, Kokkinakis G (2005) Comparative evaluation of various MFCC implementations on the speaker verification task. In: Proceedings of the SPECOM-2005, pp 191–194

21. Girard JM (2014) Carma: software for continuous affect rating and media annotation. J Open Res Softw 2(1):e5
22. Girard JM, Wright AGC (2016) DARMA: dual axis rating and media annotation **(submitted)**
23. Hantke S, Eyben F, Appel T, Schuller B (2015) iHEARu-PLAY: introducing a game for crowdsourced data collection for affective computing. In: 2015 International conference on affective computing and intelligent interaction (ACII). IEEE, pp 891–897
24. Holzinger A (2016) Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inform 3(2):119–131
25. Holzinger A (2018) From machine learning to explainable AI. In: 2018 World symposium on digital intelligence for systems and machines (DISA). IEEE, pp 55–66
26. Holzinger A, Plass M, Holzinger K, Crişan GC, Pintea CM, Palade V (2016) Towards interactive machine learning (IML): applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In: International conference on availability, reliability, and security. Springer, pp 81–95
27. Kamar E, Hacker S, Horvitz E (2012) Combining human and machine intelligence in large-scale crowdsourcing. In: International conference on autonomous agents and multiagent systems, AAMAS 2012, Valencia, June 4–8, 2012 (3 volumes), pp 467–474
28. Kennedy L, Ellis DPW (2004) Laughter detection in meetings. In: Proceedings of NIST meeting recognition workshop, Montreal, pp 118–121
29. Kim B, Pardo B (2017) I-SED: an interactive sound event detector. In: Proceedings of the 22nd international conference on intelligent user interfaces, IUI '17. ACM, New York, pp 553–557
30. Kipp M (2013) Anvil: the video annotation research tool. In: Handbook of corpus phonology. Oxford University Press, Oxford
31. Kishore KK, Satish KP (2013) Emotion recognition in speech using MFCC and wavelet features. In: International conference on advance computing conference (IACC), pp 842–847
32. Knox MT, Mirghafori N (2007) Automatic laughter detection using neural networks. In: INTERSPEECH 2007, 8th annual conference of the International Speech Communication Association, Antwerp, August 27–31, 2007, pp 2973–2976
33. Lee CM, Yildirim S, Bulut M, Kazemzadeh A, Busso C, Deng Z, Lee S, Narayanan S (2004) Emotion recognition based on phoneme classes. In: International conference on spoken language processing (ICSLP), pp 889–892
34. Lingenfelser F, Wagner J, André E (2011) A systematic discussion of fusion techniques for multi-modal affect recognition tasks. International conference on multimodal interfaces (ICMI), ICMI '11. ACM, New York, pp 19–26
35. Lingenfelser F, Wagner J, André E, McKeown G, Curran W (2014) An event driven fusion approach for enjoyment recognition in real-time. In: International conference on multimedia (MM), MM '14. ACM, New York, pp 377–386
36. Lotfian R, Busso C (2017) Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. IEEE Trans Affect Comput 10(4):471–483
37. Mayor O, Llimona Q, Marchini M, Papiotis P, Maestre E (2013) repoVizz: a framework for remote storage, browsing, annotation, and exchange of multi-modal data. In: Proceedings of the 21st ACM international conference on multimedia, MM '13. ACM, New York, pp 415–416
38. Neiberg D, Elenius K, Laskowski K (2006) Emotion recognition in spontaneous speech using GMMs. In: Conference of the International Speech Communication Association (INTERSPEECH)
39. Poignant J, Budnik M, Bredin H, Barras C, Stefas M, Bruneau P, Adda G, Besacier L, Ekenel HK, Francopoulo G, Hernando J, Mariani J, Morros R, Quénot G, Rosset S, Tamisier T (2016) The CAMOMILE collaborative annotation platform for multi-modal, multi-lingual and multi-media documents. In: Proceedings of the tenth international conference on language resources and evaluation LREC 2016, Portorož, May 23–28, 2016
40. Rabiner L, Juang BH (1993) Fundamentals of speech recognition. Prentice-Hall, Upper Saddle River
41. Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?": explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, August 13–17, 2016, pp 1135–1144
42. Rosenthal S, Dey AK (2010) Towards maximizing the accuracy of human-labeled sensor data. In: Proceedings of the 2010 international conference on intelligent user interfaces, February 7–10, 2010, Hong Kong, pp 259–268
43. Schmidt T (2004) Transcribing and annotating spoken language with EXMARaLDA. In: Proceedings of the international conference on language resources and evaluation: workshop on XML based richly annotated corpora, Lisbon 2004. ELRA, Paris, pp 879–896
44. Schuller B, Batliner A, Seppi D, Steidl S, Vogt T, Wagner J, Devillers L, Vidrascu L, Amir N, Kessous L, Aharonson V (2007) The relevance of feature type for the automatic classification of emotional user states: low level descriptors and functionals. In: INTERSPEECH. ISCA, pp 2253–2256
45. Schuller B, Steidl S, Batliner A, Vinciarelli A, Scherer KR, Ringeval F, Chetouani M, Weninger F, Eyben F, Marchi E, Mortillaro M, Salamin H, Polychroniou A, Valente F, Kim S (2013) The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In: INTERSPEECH 2013, 14th annual conference of the international Speech Communication Association, Lyon, August 25–29, 2013, pp 148–152
46. Settles B (2010) Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, vol 52, pp 55–66
47. Settles B (2012) Active learning: synthesis lectures on artificial intelligence and machine learning. Morgan and Claypool, San Rafael
48. Shortliffe EH, Buchanan BG (1975) A model of inexact reasoning in medicine. Math Biosci 23(3):351–379
49. Stikic M, Laerhoven KV, Schiele B (2008) Exploring semi-supervised and active learning for activity recognition. In: 12th IEEE international symposium on wearable computers (ISWC 2008), September 28–October 1, 2008, Pittsburgh, pp 81–88
50. Tong S, Koller D (2002) Support vector machine active learning with applications to text classification. J Mach Learn Res 2:45–66
51. Urbain J, Niewiadomski R, Bevacqua E, Dutoit T, Moinet A, Pelachaud C, Picart B, Tilmanne J, Wagner J (2010) Avlaughtercycle. J Multimodal User Interfaces 4(1):47–58
52. Valstar MF, Baur T, Cafaro A, Ghitulescu A, Potard B, Wagner J, André E, Durieu L, Aylett M, Dermouche S, Pelachaud C, Coutinho E, Schuller B, Zhang Y, Heylen D, Theune M, van Waterschoot J (2016) Ask Alice: an artificial retrieval of information agent. In: Proceedings of the 18th ACM international conference on multimodal interaction. ACM, pp 419–420
53. Valstar MF, Gunes H, Pantic M (2007) How to distinguish posed from spontaneous smiles using geometric features. In: Proceedings of the 9th international conference on multimodal interfaces. ACM, pp 38–45
54. Vinciarelli A, Pantic M, Bourlard H (2009) Social signal processing: survey of an emerging domain. Image Vis Comput 27(12):1743–1759
55. Vinciarelli A, Pantic M, Bourlard H, Pentland A (2008) Social signal processing: state-of-the-art and future perspectives of an emerging domain. In: International conference on multimedia (MM), Vancouver, pp 1061–1070
56. Vogt T, André E (2005) Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition.

In: International conference on multimedia and expo (ICME), pp 474–477
57. Wagner J, André E, Kugler M, Leberle D (2010) SSI/ModelUI—a tool for the acquisition and annotation of human generated signals. In: DAGA 2010. TU Berlin, Berlin
58. Wagner J, Lingenfelser F, André E, Kim J, Vogt T (2011) Exploring fusion methods for multimodal emotion recognition with missing data. Affect Comput 2(4):206–218
59. Wagner J, Lingenfelser F, André E, Mazzei D, Tognetti A, Lanatà A, Rossi DD, Betella A, Zucca R, Omedas P, Verschure PF (2013) A sensing architecture for empathetic data systems. In: Augmented human international conference (AH). ACM, Stuttgart, pp 96–99
60. Wagner J, Lingenfelser F, Baur T, Damian I, Kistler F, André E (2013) The social signal interpretation (ssi) framework: multimodal signal processing and recognition in real-time. In: Proceedings of the 21st ACM international conference on Multimedia, MM '13. ACM, New York, pp 831–834
61. Wagner J, Seiderer A, Lingenfelser F, André E (2015) Combining hierarchical classification with frequency weighting for the recognition of eating conditions. In: INTERSPEECH 2015, 16th annual conference of the International Speech Communication Association, Dresden, September 6–10, 2015, pp 889–893
62. Wang M, Hua XS (2011) Active learning in multimedia annotation and retrieval: a survey. ACM Trans Intell Syst Technol 2(2):10:1–10:21
63. Wittenburg P, Brugman H, Russel A, Klassmann A, Sloetjes H (2006) Elan: a professional framework for multimodality research. In: Proceedings of the fifth international conference on language resources and evaluation (LREC), pp 879–896
64. Zhang Y, Coutinho E, Schuller B, Zhang Z, Adam M (2015) On rater reliability and agreement based dynamic active learning. In: International conference on affective computing and intelligent interaction, ACII. Xi'an, pp 70–76
65. Zhang Y, Coutinho E, Zhang Z, Quan C, Schuller B (2015) Dynamic active learning based on agreement and applied to emotion recognition in spoken interactions. In: Proceedings of the 2015 ACM on international conference on multimodal interaction, ICMI '15. ACM, New York, pp 275–278
66. Zhang Z, Coutinho E, Deng J, Schuller B (2015) Cooperative learning and its application to emotion recognition from speech. IEEE/ACM Trans Audio Speech Lang Process 23(1):115–126
67. Zhu X (2005) Semi-supervised learning literature survey. Tech. rep., Computer Sciences, University of Wisconsin-Madison