The Prosody Module

Viktor Zeißler, Johann Adelhardt, Anton Batliner, Carmen Frank, Elmar Nöth, Rui Ping Shi and Heinrich Niemann

Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
{zeissler,adelhardt,batliner,frank,noeth,shi,
niemann}@informatik.uni-erlangen.de,

Summary. In multimodal dialogue systems, several input and output modalities are used for user interaction. The most important modality for *human computer interaction* is speech. Similar to human human interaction, it is necessary in the human computer interaction that the machine recognizes the spoken word chain in the user's utterance. For better communication with the user it is advantageous to recognize his internal emotional state because it is then possible to adapt the dialogue strategy to the situation in order to reduce, for example, anger or uncertainty of the user.

In the following sections we describe first the state of the art in emotion and user state recognition with the help of prosody. The next section describes the *prosody module*. After that we present the experiments and results for recognition of user states. We summarize our results in the last section.

1 The State of the Art

Prosody refers to the segments of speech larger than phonemes, e.g., syllables, words, phrases, and whole utterances. These segments are characterized with properties like pitch, loudness, duration, speaking rate, and pauses. The machine can analyze these properties and detect prosodic events, such as accents and phrase boundaries, as well as decide the mood or emotion in which a human expresses a certain utterance (Adelhardt et al., 2003). With the help of prosody it is consequently possible to get more knowledge about the user who is "talking" with the system, as has been, for instance, shown in some studies (Dellaert et al., 1996; Amir and Ron, 1998; Li and Zhao, 1998; Petrushin, 2000).

User states are an extension of the well-known term of emotion with some internal states of a human like, e.g., "*hesitant*", that are important in the context of human computer interaction (HCI). This extension of emotion refers to the interaction of users with the system, for instance, if the user shows hesitance or uncertaincy because he does not know how the machine can help him. For details, see also Streit et al. (2006).

One problem with the recognition of user states is the difficulty of data collection. In most cases actors "create" emotions according to some certain scenario, but recognizers trained with these *actor data* are not applicable for emotion detection with naive speakers. An alternative method to collect data for training an emotional recognizer is the so-called Wizard-of-Oz experiment (WOZ, see Schiel and Türk (2006)).

In the research of emotion recognition through prosody, generally three base features are used: fundamental frequency (F0 or pitch), duration, and energy. Furthermore, these base features can be combined with several other features. In Dellaert et al. (1996) actor data of five speakers are used to detect four emotions — joy, sadness, rage, and fear. The authors use several F0 features, speaking rate and statistics about the whole utterance. In Amir and Ron (1998) actor data of more than 20 persons are used to detect joy, anger, grief, fear, and disgust. The authors use, e.g., F0, energy and derived features based mainly on the whole utterance. In Li and Zhao (1998) joy, anger, fear, surprise, sadness, and neutral are detected with actor data of five speakers. The authors use, e.g., formants, F0, energy, and derived short-term features as well as several derived long-term features. In Petrushin (2000) data of 30 persons showing 5 emotions, joy, rage, sorrow, fear, and neutral, in their speech are classified. The authors use features like F0, energy, speaking rate, formants, and bandwidth of formants. In addition, minima, maxima, average, and regression of the features above are used.

Different classification techniques can be applied to emotion classification. In Dellaert et al. (1996) maximum likelihood Bayes classification, kernel regression, and k-nearest neighbor are used for classification. In Amir and Ron (1998) the authors use two methods for classification. The first method computes wordwise emotion scores averaged over the utterance, which are compared against each other to determine the emotion. The second method suggests framewise classification followed by the final decision based on majority voting of the frames in each emotional class. In Li and Zhao (1998) the choice of the features for classification is based on principal component analysis (PCA), while classification results from vector quantization, Gaussian mixture models, and *artificial neural networks* (ANN) are also used in Petrushin (2000).

Another contribution that is very important to our work stems from Huber (2002). The author uses wordwise as well as turnwise prosodic features and linguistic information for the classification of emotion (for the features, see Sect. 2.4 and Nöth et al. (2000); Kießling (1997)).

There are some other methods for emotion detection based on evaluation of linguistic information. One possibility is keyword spotting, where the utterance is checked against certain words (Lee et al., 2002; Arunachalam et al., 2001). Another method is the use of *part of speech* features (POS) introduced in Batliner et al. (1999).

In the SmartKom project, speech, gesture, and facial expressions are used for emotion recognition. In the further context of emotion recognition, there are several studies in the area of the term "affective computing," which has been established mainly by R. Picard. Affective computing combines several information channels to get the emotion of the user. An interesting introduction to this field is given in

Picard (1997). It covers, e.g., emotion in speech, facial expression, and physiological signals.

2 Module Description

The prosody module used in the SmartKom demonstrator is based on the Verbmobil prosody module described in detail in Batliner et al. (2000a) and Nöth et al. (2000). Compared to the Verbmobil version, several major changes have been made concerning both implementation and classification models. Since the SmartKom system provides a different approach for module communication (Herzog and Ndiaye, 2006), the module interface has been fully reimplemented. The classification core remains essentially the same, except for some minor changes, which increase the stability and performance of the module. All existing classification models for the recognition of prominent words, phrase boundaries, and questions have been retrained on the actual SmartKom WOZ dataset (Schiel and Türk, 2006). This makes it possible to achieve much better recognition results than those obtained with the old models on the current dataset (Sect. 3.1). Additionally, the user state classifier has been trained and integrated into the module.

In the following sections we first give a brief overview of the overall module structure and the data flow in the module (Sect. 2.1). The issues concerning the execution of the module in the SmartKom system and synchronization of input data streams are covered in Sect. 2.2 and 2.3, respectively. Afterwards the features used for classification are presented in Sect. 2.4, followed by the description of the prosodic classifiers in Sect. 2.5.

2.1 Architecture

The goal of the prosody module in SmartKom is to analyze the speech as a modality of the user input in order to detect the prosodic events as well as the most likely emotional state of the user (Streit et al., 2006). As shown in Fig. 1, the module has two main inputs: the speech signals from the *audio module* and the word lattices (*word hypothesis graphs*, WHGs) from the *speech recognizer*. After running through feature extraction and classification steps, the detected prosodic events are added to the original input WHG and the user state lattice is generated. In more detail, the subsequent processing steps are described below:

- *XML parser:* According to the SmartKom communication paradigm all data exchanges are done in the XML format. The incoming XML packets have to be parsed and filtered. Thus, we can check the data consistency, drop the irrelevant information, and convert the useful data to a compact internal representation.
- *Stream synchronization:* This component compares the time stamps of incoming packets to find the correct mapping between the WHGs and the speech data. It also ensures the amount of data is enough to trigger the feature extraction for a given WHG. For the detailed description see Sect. 2.3.

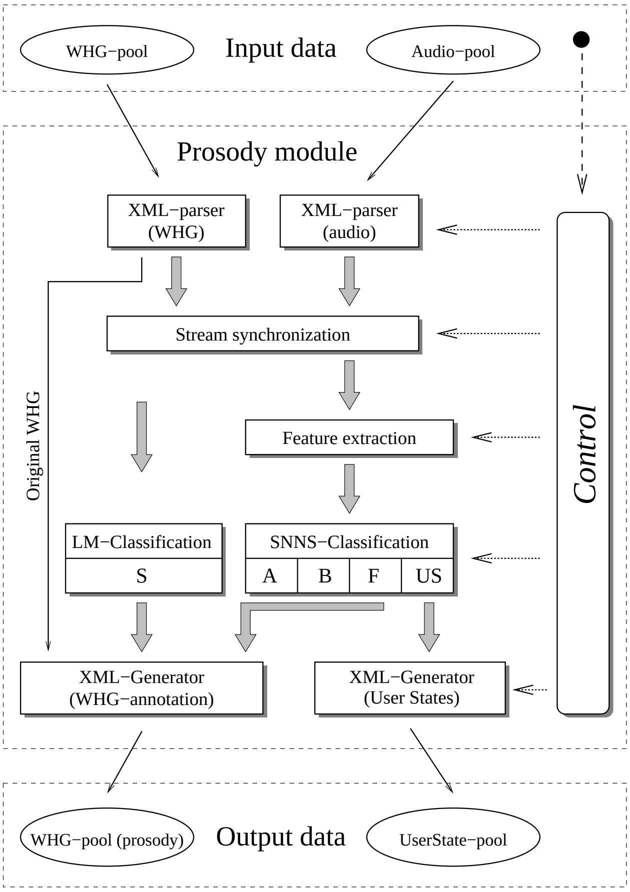

Fig. 1. Architecture of the prosody module

- *Feature extraction:* After the synchronization of the input data, a speech signal segment corresponding to each word in the WHG is located. The various F0, energy, and durational features of one segment are combined into a single feature vector to be used at the classification step.
- *ANN classification:* The ANN classification block consists of four independent classifiers used for the detection of:
  - phrase *A*ccents (A-labels),
  - phonetic phrase *B*oundaries (B-labels),
  - rising intonation for *Q*ueries (Q-labels),
  - *U*ser *S*tates (US-labels).

  For each word and label the classifiers generate a likelihood score representing a confidence of the recognition. The scores of each classifier are normalized to resemble the real probabilities (summed up to 1), though their distribution might be quite different.

- *LM classification:* The recognized words taken from the WHG are used here to detect *S*yntactic phrase boundaries (S-labels) with a *Language Model classifier* described in Batliner et al. (2000a); Schukat-Talamazzini (1995).
- *XML generators:* The output of classifiers is used to annotate the original WHG stored at the parsing step. Technically, we generate the XML structure only for the annotation part and paste it to the original lattice. This improves the performance of the module because disassembling of the complex WHG structure and reassembling of a new WHG from scratch can be avoided. Additionally, the user state labels are used to generate the user state lattice for the current turn. After generation, both lattices are sent to the output pools.
- *Control:* The control component determines the order of execution of all other parts and also handles the data transfer.

Apart from the main input pools shown in Fig. 1 the prosody module also gets data from the lexicon and the configuration pool. The processing of lexicon data makes it possible to update the internal static lexicon with new entries. The configuration data are used to set the internal module parameters from the GUI.

2.2 Execution

After being started by the SMARTKOM system the prosody module goes through several initialization steps including reading the configuration files, setting up the classifiers and loading statistical models. To be able to interact with the rest of the system, the module then subscribes the needed communication resources and installs the pool handlers.

The SMARTKOM communication system is based on the so-called *Pool Communication Architecture* (PCA) described in Herzog and Ndiaye (2006). There are several I/O FIFOs called communication pools which run in an asynchronous manner. If any module puts data into a pool, all modules that have subscribed this pool will be notified and can access the data. There are two possibilities to get notification events. The module can wait for these events by calling a special function or it can install handlers to be called when the notification event arrives. The prosody module handles the controlling events, such as exit or restart commands in the former and the pool notification events in the latter way.

When there are new data in one of the input pools subscribed by the module, the installed pool handler is called and the control component becomes active. First of all, the appropriate XML parser is called to get the data from the pool. When a new WHG arrives, the control component tries then to find the matching speech signal region (synchronization component). If it succeeds, the module registers itself as *processing* to the SMARTKOM system and proceeds to the further steps: feature extraction, classification and the data output.

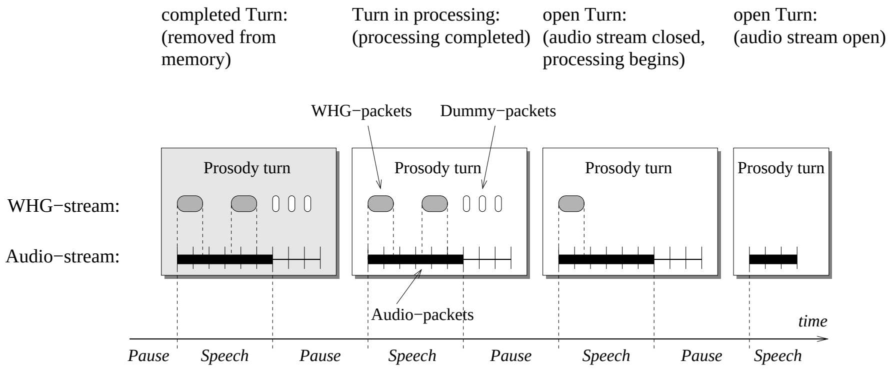

Fig. 2. Synchronization of audio and WHG streams in the prosody module

2.3 Data Synchronization

The essential part of the module managing the internal data flow is the stream synchronization component. Several problems must be solved to keep the data processing running smoothly from the technical point of view:

- memory allocation for the subsequently arriving data packets
- building the data structures needed for the following steps
- memory disallocation for the already processed data

Furthermore, some special features of the modules providing the input data should be considered. It concerns, for instance, the *silence detection* scheme. The audio module records the speech until silence is detected. The first and the last packet of the audio sequence between silences are labeled, so that such sequences can easily be identified. During silence dummy audio packets are generated in regular intervals. They are handled by the speech recognizer and can be ignored by the prosody module. The silence detection from the audio module regards only the signal energy and therefore only a robust detection of long pauses is possible. The speech recognizer has its own silence detection that also works for shorter and less distinct pauses. For every speech segment between such *short silences*, the speech recognizer generates a WHG and sends it to the output pool. Thus, there can be more than one WHG for a single audio sequence that needs to be properly aligned. Another factor to be considered is the processing of the dummy packets. The dummy packets from the audio module are transformed by the speech recognizer to dummy WHGs and sent to the WHG pool. These packets are needed for the robust detection of the user turns and dialogue management of SMARTKOM, therefore they should be passed through prosody module to the output pool “as is.”

To reflect the SMARTKOM dialogue turns, the stream synchronization component works also with turn-based structures. Correspondingly, the *prosodic turn* is defined as a continuous speech sequence between the silences detected by the audio module. The speech data are collected from the incoming audio packets and stored in an array

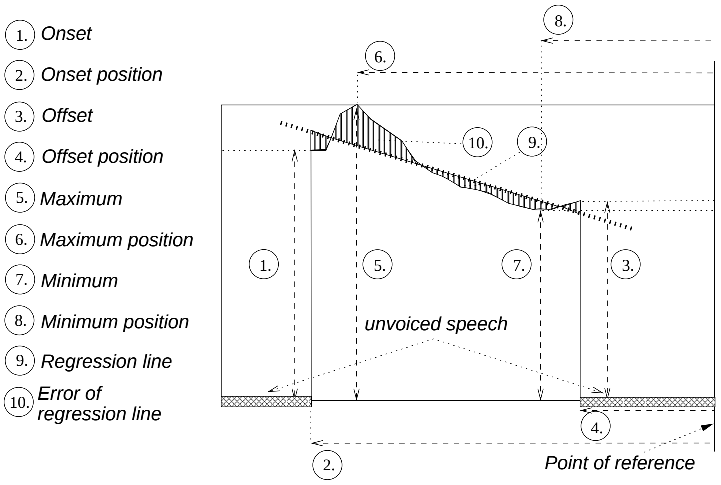

Fig. 3. Example of features describing the pitch contour (Batliner et al., 2000a)

in the turn structure. With the begin of a new audio sequence, the next turn structure is created to collect the incoming audio and WHG streams. When a new WHG packet arrives, the time stamps are compared to decide which turn the WHG belongs to. The processing scheme is illustrated in Fig. 2. Although this makes it necessary to hold open and manage several turns at a time, it has obviously the following advantages:

- The speech data belonging to one turn are stored in a single array and do not need to be pasted together from different packets. It also means smaller overhead at the feature extraction step.
- The memory management is done in the following manner: After the last WHG in a turn is processed, the complete turn, including all WHG, speech, and all intermediate data, is deleted and there is no need for costly garbage collection.

2.4 Feature Extraction

To recognize the prosodic characteristics of a speech signal we use the features describing such prosodic cues as energy, pitch, and duration, but also some linguistic information, in particularly POS features. The feature extraction is performed in two steps. First, we compute two *basic prosodic features*: integrated short time energy and pitch values framewise with a step size of 10 ms (Kießling, 1997). At the second step we take the WHG to determine the word boundaries and compute for each word a set of *structured prosodic features* (Nöth et al., 2000; Kießling, 1997). Every feature of the set has the following three configuration parameters:

- *Extraction interval:* To incorporate the word context information, the features can be computed on the subsequent words within a five-word interval beginning two words before and ending two words after the actual word. For example, we can take a two-word interval beginning one after the word in focus and ending two words after it. We use no more than a two-word range for such intervals. The larger intervals in our experiments brought no further improvement.

- *Basic prosodic cue:* Every feature can be computed using one of the following information cues: energy and pitch (computed at the first step), duration (taken from the WHG according to the chosen extraction interval), and linguistic information (taken from the dictionary).
- *Feature type:* For every prosodic cue there are a number of different features that can be subdivided into two main groups: *contour describing* and *integrated* features. The features of the first group including onset/offset/maximum/minimum values and their positions, regression coefficient, and regression error are shown in Fig. 3. They can be applied to the pitch and some of them to the energy contour. The second group includes mean and normalized values of the energy and duration cues. To eliminate the influence of microprosodic and global factors, such as the intrinsic word duration and the speech rate (when applied to duration cue), we use twofold normalization, described in Zeißler et al. (2002).

To sum up, we compute a total set of 91 prosodic and 30 POS features, described in detail in Batliner et al. (2000b). We use different subsets for classification of different prosodic events. The total set of 121 features is used for the detection of phrase accents and phrase boundaries. With a subset of 25 F0 features we classify the rising intonation (queries). For the classification of the user states, we use only 91 prosodic features.

2.5 Classification

For the prosodic classification we use a *multilayer perceptron*, a special kind of artificial neural network (ANN). We normalize all the input features to the mean of zero and the variance of one. The normalization coefficients are estimated in advance on the whole training dataset. To find an optimal training configuration, we need to know the following parameters: network topology, training weight for the RPROP training algorithm (Riedmiller and Braun, 1993), and random seed for the initialization. In preliminary tests we found out that complex topologies with two or more hidden layers do not improve the results for our dataset in comparison to a simple perceptron with one hidden layer. Hence, we restrict the number of hidden layers to one and look only for the optimal number of nodes in the hidden layer. We then evaluate different combinations of these parameters and choose the configuration with the best result on the validation set.

As primary classification method we use the wordwise classification. For each word ω_i we compute a probability $P(s \mid \omega_i)$ to belong to one of the given user states s. The probability maximum then determines the classification result. Further, we use these probabilities to classify the whole utterance, assuming the conditional independence between word classification events (Huber et al., 1998). The utterance probabilities are computed according to the following equation:

Table 1. The classwise averaged (Cl) and total (Rr) recognition rates (in %) yielded for a two-class problem on phrase accents, phrase boundaries, and rising intonation

Classifiers	SmartKom				Verbmobil			
Tested on	SK-train		SK-test		SK-test		Verbmobil	
	CL	RR	CL	RR	CL	RR	CL	RR
Prominent words	81.2	81.0	77.1	77.0	72.8	72.8	79.2	78.4
Phrase boundaries	90.7	89.8	88.5	88.6	82.2	82.7	81.5	85.8
Rising intonation	75.6	72.0	79.2	66.4	57.4	81.1	60.1	89.7

$$P(s \mid \omega_1, \omega_2, \ldots, \omega_n) \approx \prod_{i=1}^{n} P(s \mid \omega_i) \; . \qquad (1)$$

3 Experiments

3.1 Detection of Prosodic Events

Like in Verbmobil, in SmartKom we have three different prosodic events to detect: phrase accents, phrase boundaries, and rising intonation. We use the WOZ dataset, since it contains the labeling we need (Schiel and Türk, 2006). For each event we use a two-class mapping of the existing labels described in Batliner et al. (2000a): one class for the strong occurrence of the event and the other for its absence.

In our experiments we use a part of the WOZ dataset consisting of 85 public wizard sessions. It includes 67 minutes of speech data (the total session length is about 2.5 hours) collected from 45 speakers (20 m/25 w). We divide the data into training and test sets built from 770 and 265 sentences, respectively. The test set contains only speakers unseen in the training set. The recognition rates on both sets are given in columns 1 to 4 of Table 1. To ensure the newly trained networks have a better performance than those of the old Verbmobil classifiers, we also tested the Verbmobil networks on the test data. The results are given in columns 5 and 6 of Table 1 (For comparison, the results of the Verbmobil networks on the Verbmobil dataset are given in columns 7 and 8).

Comparing the results on test and training sets we see small (but significant) differences. Nonetheless, we conclude that the trained classifier has rather good generalization capability for the recognition of prosodic events on the SmartKom data. On the other hand, we observe rather big differences if comparing it with the results of Verbmobil classifiers, especially in the detection of rising intonation. It illustrates the necessity to retrain the old classifiers on the actual dataset.

3.2 Detection of User States on the WOZ Data

For the detection of user states on the WOZ data we use the same subset of 85 sessions as described above. There was no special *prosodic* labeling of user states, thus we used a so-called *holistic* labeling based on the overall impression of the labeler,

Table 2. Original labels, used class mappings and the number of words in each class

<table>
<tr><th colspan="2" rowspan="2">Original Labels</th><th colspan="5">Mappings</th></tr>
<tr><th>7 Classes</th><th>5 Classes</th><th>4 Classes</th><th>3 Classes</th><th>2 Classes</th></tr>
<tr><td>Joyful (strong)</td><td>113</td><td>113</td><td rowspan="2">805</td><td colspan="2" rowspan="3">884</td><td rowspan="6">11,491</td></tr>
<tr><td>Joyful (weak)</td><td>692</td><td>692</td></tr>
<tr><td>Surprised</td><td>79</td><td colspan="2">79</td></tr>
<tr><td>Neutral</td><td>9236</td><td colspan="4">9236</td></tr>
<tr><td>Pondering</td><td>848</td><td rowspan="2">1371</td><td colspan="2" rowspan="2">1371</td><td rowspan="4">2030</td></tr>
<tr><td>Hesitant</td><td>523</td></tr>
<tr><td>Angry (weak)</td><td>483</td><td>483</td><td colspan="2" rowspan="2">659</td><td rowspan="2">659</td></tr>
<tr><td>Angry (strong)</td><td>176</td><td>176</td></tr>
</table>

taking both speech and facial expression into account: During the annotation the labeler could observe the test person and hear the recorded utterances simultaneously. Then he was supposed to detect the changes in the user state of the test person and assign the corresponding label to the actual time interval (Schiel and Türk, 2006). The disadvantage of this method is obvious: If the observed user state change is judged only after the user's facial expression, gesture, or/and the situational context, there will be no prosodic equivalent of the assigned label, and this fact will in turn have negative influence on the classifier training and recognition results.

After the preprocessing of the annotations, we have one of the different user state labels assigned to each word. Thus, we apply only a word-based classifier as described in Sect. 2.5. In our experiments we used different mappings to get 2-, 3-, 4-, 5-, and 7-class problems. The original labels, the used class mappings, and the number of words in each class are shown in Table 2. The main problem for the automatic classification is a strong unequal distribution of the different classes. In case we want to train a classifier for a two-class problem, we have 659 words labeled as angry vs. 11,491 words with other labels. To get stable recognition results under such unfavorable conditions we conduct only *leave-one-speaker-out* (LOSO) tests on the whole dataset with neural networks (ANN) and *leave-one-out* (LOO) tests with *linear discriminant analysis* (LDA) from the SPSS software (Norusis, 1998). The results are shown in two last columns of Table 3.

Above we pointed out that there was only a holistic labeling available, taking into account both speech and facial expression at the same time: *u*ser *s*tates, *h*olistic (USH). In a second pass, other annotators labeled all nonneutral user states based purely on facial expressions: *u*ser *s*tates *f*acial (USF); details can be found in Batliner et al. (2003). For further classification experiments, we divided the whole database into two disjunct subsets, based on the agreement of USH and USF for the four "basic" user states *positive, neutral, hesitant,* and *angry*: *agreeing* (USH $=$ USF) and *not agreeing* (USH $\neq$ USF). LOO classification was this time done with LDA and decision trees (J48) (Witten and Frank, 2000), and all 91 prosodic and 30 POS features. Recognition rates for *not agreeing* cases were lower, and for *agreeing* cases higher than for all cases taken together. For example, for seven classes, LDA yields a

Table 3. Classwise averaged recognition rates (in %) yielded in Lo(S)O tests on the WOZ data for five different class mappings. The results of two different classifiers, ANN and LDA, are given in last two columns

No. of classes	User states							Results	
								ANN	LDA
7	Joyful (strong)	Joyful (weak)	Surprised	Neutral	Hesitant	Angry (strong)	Angry (weak)	30.8	26.8
5	Joyful		Surprised	Neutral	Hesitant	Angry		36.3	34.2
4	Positive			Neutral	Hesitant	Angry		34.5	39.1
3	Positive			Neutral	Problem			42.7	45.5
2	Not angry					Angry		66.8	61.8

classwise averaged recognition rate of 29.7%, J48 47.5; for the two classes *not angry* vs. *angry*, the figures are 68.9% for LDA and 76.5% for J48. These results indicate a mutual reinforcment of the two modalities, which, in turn, leads to a more stable annotation if holistic and facial impressions converge.

3.3 Detection of user states on the Multimodal Emogram (MMEG) data

In experiments described below we use the data from the MMEG collection introduced in Adelhardt et al. (2006). In this dataset we have sentences labeled with one of four user states: *joyful*, *neutral*, *helpless*, and *angry* (Streit et al., 2006). From all collected speech data with good quality we choose randomly 4292 sentences for the training set and 556 for the test set (*test1*). Notice that we have here approximately 60% of sentences with the same syntactic structure in both training and test sets that were definitely dependent on the used user state. For example, all sentences built after the pattern *"I don't like <some TV genre>"* belong to the user state *angry*. To ensure that we really recognize user states and not the different syntactic structures of the sentences, we additionally select 1817 sentences for another test set (*test2*). The second test set contains only sentences with a syntactic structure independent of the labeled user state, for instance, sentences consisting of an isolated name of a TV genre or special expressions (Adelhardt et al., 2006). Thus, for this set, the syntactic structure of the sentence could not be used as a key to a proper recognition.

To train the classifier we had first to find out the optimal feature set. We tried different subset combinations of F0-based features, all prosodic features (91 set), and linguistic POS features in both context-dependent and independent form. In context-independent feature sets we used only the features computed for the word in question. For all configurations we trained the neural networks and tested them on the test sets (see the results of *test1* vs. *test2* in Table 4). The classwise averaged recognition rates for the four class problems (in percent) are shown in Table 4. We computed both word based and sentence-based recognition rates as indicated in the second column.

In Table 4 we notice that the POS features yield remarkable improvement only on the *test1* set; the results on the *test2* set get worse (see columns 3 and 5). That means they reflect to a great extent the sentence structure and therefore cannot be properly

Table 4. Classwise averaged recognition rates (in %) for four user states classified with ANN using five different feature sets. For each test set the wordwise and the sentencewise recognition rates are given

Test set	Type	Without context			With context	
		F0 feat. 12 feat.	All pros. 29 feat.	Pros.+POS 35 feat.	All pros. 91 feat.	Pros.+POS 121 feat.
Test1	Word	44.8	61.0	65.7	**72.1**	86.6
	Sentence	54.0	64.5	72.1	**75.4**	81.4
Test2	Word	36.9	46.8	46.5	**54.5**	52.7
	Sentence	39.8	47.5	48.1	**55.1**	54.2

Table 5. Confusion matrix of user state recognition with the ANN on the best feature set (91 features) using LOSO. Both the wordwise and sentencewise results are computed (in %)

Reference	Wordwise				Sentencewise			
User state	Neutral	Joyful	Angry	Hesitant	Neutral	Joyful	Angry	Hesitant
Neutral	**68.3**	12.5	12.6	6.6	**67.7**	12.0	16.5	3.8
Joyful	13.8	**65.8**	10.6	9.8	14.3	**66.4**	13.8	5.5
Angry	14.5	11.3	**64.7**	9.5	13.9	9.2	**70.8**	6.1
Hesitant	10.0	10.8	9.9	**69.3**	10.0	6.5	15.3	**68.2**

applied for the user state recognition in our case due to the construction of the corpus. The best results were achieved with the 91 prosodic feature set (75.4% *test1*, 55.1% *test2*, sentencewise). To verify these results with the speaker-independent tests, we additionally conducted a LOSO test using the 91 feature set. Here we achieved an average recognition rate of 67.0% wordwise and 68.3% sentencewise. The confusion matrix of this test is given in Table 5.

4 Conclusion

The prosody module used in the SmartKom demonstrator is based on the Verbmobil prosody module described in detail in Batliner et al. (2000a) and is extended with several new features. The module detects phrase accents, phrase boundaries, and rising intonation (prosodic marked queries) and includes new features concerning module communication, data synchronization, and a classifier for user state recognition. User state classification is done in two steps. In the first step we use word-based classification to compute a probability to assign one of several user states to each word. In the second step we process the probability of the whole utterance to decide one of the several classes.

For classification of prosodic events with the test set, we obtain a classwise averaged recognition rate of 77.1% for phrase accents, 88.5% for phrase boundaries, and 79.2% for rising intonation. For user state classification we collected our own data, due to the lack of training samples in WOZ data. Regarding the recognition of the user states, we noticed that POS features yield a remarkable improvement only for the *test1* set containing sentences with the same syntactic structure as the training set. For the *test2* set, which contains only one-word sentences and special expressions, POS features have worsened rather than improved the results. Because of the construction of the MMEG database, the POS features reflect to a great extent the structure of sentence and thus cannot be properly applied to the user state recognition in this case. In a speaker-independent test for user state classification without POS features we achieved an average recognition rate of 67.0% wordwise and 68.3% sentencewise.

References

J. Adelhardt, C. Frank, E. Nöth, R.P. Shi, V. Zeißler, and H. Niemann. Multimodal Emogram, Data Collection and Presentation, 2006. In this volume.

J. Adelhardt, R.P. Shi, C. Frank, V. Zeißler, A. Batliner, E. Nöth, and H. Niemann. Multimodal User State Recognition in a Modern Dialogue System. In: *Proc. 26th German Conference on Artificial Intelligence (KI 03)*, pp. 591–605, Berlin Heidelberg New York, 2003. Springer.

N. Amir and S. Ron. Towards an Automatic Classification of Emotions in Speech. In: *Proc. ICSLP-98*, vol. 3, pp. 555–558, Sydney, Australia, 1998.

S. Arunachalam, D. Gould, E. Andersen, D. Byrd, and S. Narayanan. Politeness and Frustration Language in Child-Machine Interactions. In: *Proc. EUROSPEECH-01*, pp. 2675–2678, Aalborg, Denmark, September 2001.

A. Batliner, A. Buckow, H. Niemann, E. Nöth, and V. Warnke. The Prosody Module. In: W. Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 106–121, Berlin Heidelberg New York, 2000a. Springer.

A. Batliner, R. Huber, H. Niemann, E. Nöth, J. Spilker, and K. Fischer. The Recognition of Emotion. In: W. Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 122–130, Berlin Heidelberg New York, 2000b. Springer.

A. Batliner, M. Nutt, V. Warnke, E. Nöth, J. Buckow, R. Huber, and H. Niemann. Automatic Annotation and Classification of Phrase Accents in Spontaneous Speech. In: *Proc. EUROSPEECH-99*, vol. 1, pp. 519–522, Budapest, Hungary, 1999.

A. Batliner, V. Zeißler, C. Frank, J. Adelhardt, R.P. Shi, E. Nöth, and H. Niemann. We Are Not Amused – But How Do You Know? User States in a Multi-Modal Dialogue System. In: *Proc. EUROSPEECH-03*, vol. 1, pp. 733–736, Geneva, Switzerland, 2003.

F. Dellaert, T. Polzin, and A. Waibel. Recognizing Emotion in Speech. In: *Proc. ICSLP-96*, vol. 3, pp. 1970–1973, Philadelphia, PA, 1996.

G. Herzog and A. Ndiaye. Building Multimodal Dialogue Applications: System Integration in SmartKom, 2006. In this volume.

R. Huber. *Prosodisch-linguistische Klassifikation von Emotion*, vol. 8 of *Studien zur Mustererkennung*. Logos, Berlin, Germany, 2002.

R. Huber, E. Nöth, A. Batliner, A. Buckow, V. Warnke, and H. Niemann. You BEEP Machine – Emotion in Automatic Speech Understanding Systems. In: *TSD98*, pp. 223–228, Brno, Czech Republic, 1998.

A. Kießling. *Extraktion und Klassifikation prosodischer Merkmale in der automatischen Sprachverarbeitung*. Shaker, Aachen, Germany, 1997.

C.M. Lee, S.S. Narayanan, and R. Pieraccini. Combining Acoustic And Language Information For Emotion Recognition. In: *Proc. ICSLP-2002*, pp. 873–876, Denver, CO, 2002.

Y. Li and Y. Zhao. Recognizing Emotions in Speech Using Short-Term and Long-Term Features. In: *Proc. ICSLP-98*, vol. 6, pp. 2255–2258, Sydney, Australia, 1998.

M.J. Norusis. *SPSS 8.0 Guide to Data Analysis*. Prentice Hall, Upper Saddle River, NJ, 1998.

E. Nöth, A. Batliner, A. Kießling, R. Kompe, and H. Niemann. Verbmobil: The Use of Prosody in the Linguistic Components of a Speech Understanding System. *IEEE Transactions on Speech and Audio Processing*, 8(5):519–532, 2000.

V.A. Petrushin. Emotion Recognition in Speech Signal: Experimental Study, Development, and Application. In: *Proc. ICSLP-2000*, vol. IV, pp. 222–225, Beijing, China, 2000.

R.W. Picard (ed.). *Affective Computing*. MIT Press, Cambridge, MA, 1997.

M. Riedmiller and H. Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: *Proc. IEEE Intl. Conf. on Neural Networks*, pp. 586–591, San Francisco, CA, 1993.

F. Schiel and U. Türk. Wizard-of-Oz Recordings, 2006. In this volume.

E.G. Schukat-Talamazzini. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Braunschweig, Germany, 1995.

M. Streit, A. Batliner, and T. Portele. Emotion Analysis and Emotion Handling Subdialogs, 2006. In this volume.

I.H. Witten and E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques With Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000.

V. Zeißler, E. Nöth, and G. Stemmer. Parametrische Modellierung von Dauer und Energie prosodischer Einheiten. In: *Proc. KONVENS 2002*, pp. 177–183, Saarbruecken, Germany, 2002.