Multi-stream LSTM-HMM decoding and histogram equalization for noise robust keyword spotting

Martin Wöllmer · Erik Marchi · Stefano Squartini · Björn Schuller

Introduction

Automatic speech recognition (ASR) has found many applications in recent years, including dictation software, navigation systems, mobile phones, and broadcast news transcription. Generally, high ASR accuracies can be obtained whenever the application scenario allows for processing well-articulated, neutral, or read speech that is captured by suited acoustic and language models (Anan-thakrishnan and Narayanan 2007). More challenging is the recognition of spontaneous, emotionally colored, and noisy speech, which leads to higher error rates due to pronunci-ation and intonation variance or speech signal disturbances. Yet, since modern speech-based computer interfaces need to handle challenges like different speaking styles, emo-tional coloring of speech, and background noise in order to enable reliable and natural human-machine communication, novel ASR techniques that reach acceptable recognition performance in spite of demanding conditions are in the focus of current research (Lathoud et al. 2005; Mesot and Barber 2007; Windmann and Haeb-Umbach 2008; Wöllmer et al. 2009). An important discipline within the field of ASR is keyword spotting (Ketabdar et al. 2006) or spoken term detection (Mamou et al. 2007). Particularly in human-machine interaction scenarios that do not aim at unrestricted natural language understanding but rather focus on inferring the user's intention from a limited set of keywords, the requirement of extracting the full transcription of the spo-ken content can often be dropped. That is why in chal-lenging spontaneous speech processing applications the accurate detection of certain keywords is often more important than large vocabulary continuous speech recog-nition (LVCSR) (McTear 2002; Wöllmer et al. 2010).

Past research on noise-robustness in ASR has attempted to optimize many components of a speech recognition

M. Wöllmer · B. Schuller
Institute for Human-Machine Communication, Technische Universität München, Theresienstr. 90, 80333 München, Germany

M. Wöllmer
e-mail: woellmer@tum.de

E. Marchi · S. Squartini
3MediaLabs, A3LAB, DIBET, Dipartimento di Ingegneria Biomedica, Elettronica e Telecomunicazioni, Università Politecnica delle Marche, 60131 Ancona, Italy
e-mail: s.squartini@univpm.it

system with respect to the applicability in noisy conditions. Prior to acoustic feature extraction, techniques for speech enhancement (Hussain and Campbell 1998) aim at denoising the speech signal via methods such as Wiener filtering or spectral subtraction (Lathoud et al. 2005). A well-known problem which has to be considered during speech enhancement is that improved intelligibility of speech does not necessarily imply lower ASR error rates. Enhancement algorithms that are applied *after* feature extraction are referred to as feature enhancement techniques and attempt to reconstruct the 'clean' speech features from the observed noisy features, e.g., using normalization techniques like cepstral mean subtraction (CMS), cepstral mean and variance normalization (CMVN), and histogram equalization (HEQ) (Hilger and Ney 2006), Bayesian estimation approaches for feature warping (Squartini et al. 2011), or model-based feature enhancement algorithms applying Gaussian mixture models, Hidden Markov Models (HMM), or Switching Linear Dynamic Models (SLDM) (Droppo and Acero 2004; Deng et al. 2007; Schuller et al. 2008). By contrast, model adaptation approaches try to adapt the acoustic models to noisy conditions, either via transforms of models trained on clean data or by including noisy training material in the learning process (matched- or multi-condition training) (Schuller et al. 2009). Thereby most studies on noise robust ASR focus on small-vocabulary tasks, such as connected digit recognition as defined in the AURORA task (Hirsch and Pearce 2000). In this article, we focus on keyword spotting as needed for spontaneous human-machine interaction and investigate the recognition performance when adapting both, features and acoustic models to noisy conditions. More specifically, we combine Histogram Equalization with multi-condition training and examine the effect on keyword detection in conversational, emotionally colored speech. HEQ is a very effective feature enhancement technique since—in contrast to CMS and CMVN—it normalizes all moments of the probability distribution of the feature vector components and thus compensates non-linear distortions caused by noise.

A further aspect examined in this article is the incorporation of long short-term memory (LSTM) modeling into the speech decoding process. LSTM networks were originally introduced in (Hochreiter and Schmidhuber 1997) and allow for context-sensitive sequence labeling. Unlike conventional recurrent neural networks (RNN) or multilayer perceptrons (MLP), LSTM networks are able to model a self-learned amount of contextual information and were recently proven to boost performance in Tandem ASR systems (Wöllmer et al. 2009, 2010, 2011). LSTM is well-suited for modeling coarticulation effects in human speech and can be applied instead of Fernandez et al. (2007) and Wöllmer et al. (2010) or in combination with Wöllmer et al. (2011) triphone modeling. In this article, we investigate the keyword spotting performance of a multi-stream model that decodes both, conventional Mel-frequency cepstral coefficient (MFCC) features and phoneme estimates generated by an LSTM network and at the same time—unlike in our previous research (Wöllmer et al. 2010)—applies an in-domain language model for high-level context modeling. Our contribution builds on a preliminary study on the effect of multi-condition training in a challenging keyword spotting scenario (Wöllmer et al. 2011) and shows how further performance gains can be reached when HEQ is included in the front-end of the recognition system. Moreover, we evaluate our proposed multi-stream LSTM-HMM not only on clean data (as in Wöllmer et al. 2011), but also consider noisy test data in order to get an impression of the model's robustness in comparison to standard HMM-based decoding.

For our keyword detection evaluations, we consider the SEMAINE scenario involving highly spontaneous and emotional speech recorded during natural human-agent conversations. Thus, we employ the SEMAINE database[1] which was recorded to provide in-domain training material for the SEMAINE system (Schröder et al. 2008)—a conversational agent with emotional competences.

The article is structured as follows: "The SEMAINE scenario" section briefly explains the SEMAINE system tailored for emotional human-agent conversations. In "Histogram equalization" section we review the principle of HEQ which will be used in combination with multi-condition training for noise-robust keyword detection. In "Multi-stream LSTM-HMM decoding" section we outline the idea of LSTM and introduce our multi-stream LSTM-HMM decoder. Finally, in "Experiments" section we show experimental results before we draw conclusions in "Conclusion" section.

The SEMAINE scenario

The techniques for noise robust keyword detection which are described in this article will be evaluated with respect to a specific application scenario involving spontaneous and highly emotional speech. We will focus on optimizing our keyword spotter for the SEMAINE system which was developed within the SEMAINE project.[2] Unlike most task-oriented dialogue systems, the *Sensitive Artificial Listeners* representing the SEMAINE system (Schröder et al. 2008) focus on aspects of communication that are emotion-related and non-verbal (as in Memon and Treur 2010, for example). The system is designed for a one-to-

[1] http://www.semaine-db.eu.

[2] http://www.semaine-project.eu.

one dialogue situation in which one user is conversing with one of four available virtual agent characters. Besides speech, the (multimodal) interaction involves head movements and facial expressions. The SAL characters have to recognize a limited set of emotionally relevant keywords, non-linguistic vocalizations such as *laughing* or *sighing*, and the prosody with which the words are spoken. Based on the interpreted input from audio and video, the system has to show appropriate listener behavior, e.g., multimodal *backchannels*, decide when to *take the turn*, and select a suitable phrase in order to maintain the conversation.

The four SAL characters roughly represent areas in the *arousal-valence* space: 'Spike' is angry (high arousal, low valence), 'Poppy' is happy (high arousal, high valence), 'Obadiah' is sad (low arousal, low valence), and 'Prudence' is matter-of-fact (moderate arousal, moderate valence). During the conversations, the virtual characters aim to induce an emotional state in the user that corresponds to *their* typical emotional state.

Histogram equalization

To maintain an acceptable keyword recognition performance when the SEMAINE system is used in noisy conditions, efficient feature enhancement and model adaptation techniques are needed. Since HEQ was shown to be one of the most effective and versatilely applicable feature enhancement techniques (see Schuller et al. 2009, for example), this article will investigate the effect of combining multi-condition training with HEQ for improving the noise robustness of the SEMAINE keyword spotter. HEQ is also a popular technique for digital image processing where it aims to increase the contrast of pictures. In speech processing, HEQ can be used to extend the principle of cepstral mean subtraction and mean and variance normalization to all moments of the probability distribution of the feature vector components (Hilger and Ney 2006; de la Torre et al. 2005). It enhances noise robustness by compensating non-linear distortions in speech representation caused by noise and therefore reduces the mismatch between test and training data.

The main idea is to map the histogram of each component of the feature vector onto a reference histogram. The method is based on the assumption that the effect of noise can be described as a monotonic transformation of the features which can be reversed to a certain degree. As the effectiveness of HEQ is strongly dependent on the accuracy of the speech feature histograms, a sufficiently large number of speech frames has to be involved to estimate the histograms. An important difference between HEQ and other noise reduction techniques like Unsupervised Spectral Subtraction (Lathoud et al. 2005) is that no analytic assumptions have to be made about the noise process. This makes HEQ effective for a wide range of different noise processes independent of how the speech signal is parameterized.

When applying HEQ, a transformation

$$\tilde{x} = F(x) \tag{1}$$

has to be found in order to convert the probability density function $p(x)$ of a certain speech feature into a reference probability density function $\tilde{p}(\tilde{x}) = p_{ref}(\tilde{x})$. If x is a unidimensional variable with probability density function $p(x)$, a transformation $\tilde{x} = F(x)$ leads to a modification of the probability distribution, so that the new distribution of the obtained variable $\tilde{x}$ can be expressed as

$$\tilde{p}(\tilde{x}) = p(G(\tilde{x}))\frac{\partial G(\tilde{x})}{\partial \tilde{x}} \tag{2}$$

with $G(\tilde{x})$ being the inverse transformation of $F(x)$. To obtain the cumulative probabilities out of the probability density functions, we have to consider the following relationship:

$$\begin{aligned} C(x) &= \int_{-\infty}^{x} p(x')dx' \\ &= \int_{-\infty}^{F(x)} p(G(\tilde{x}'))\frac{\partial G(\tilde{x})}{\partial \tilde{x}'}d\tilde{x}' \\ &= \int_{-\infty}^{F(x)} \tilde{p}((\tilde{x}')d\tilde{x}' \\ &= \tilde{C}(F(x)). \end{aligned} \tag{3}$$

Consequently, the transformation converting the distribution $p(x)$ into the desired distribution $\tilde{p}(\tilde{x}) = p_{ref}(\tilde{x})$ can be expressed as

$$\tilde{x} = F(x) = \tilde{C}^{-1}[C(x)] = C_{ref}^{-1}[C(x)], \tag{4}$$

whereas $C_{ref}^{-1}(\ldots)$ is the inverse cumulative probability function of the reference distribution and $C(\ldots)$ is the cumulative probability function of the feature. To obtain the transformation for each feature vector component in our experiments, uniform intervals between $\mu_i - 4\sigma_i$ and $\mu_i + 4\sigma_i$ were considered to derive the histograms, with μ_i and σ_i representing the mean and the standard deviation of the ith feature vector component. For each component a Gaussian probability distribution with zero mean and unity variance was used as reference probability distribution.

Multi-stream LSTM-HMM decoding

A second strategy for enhancing keyword detection performance in challenging conditions as investigated in this article is the application of LSTM (Hochreiter and Schmidhuber 1997) for context-sensitive phoneme estimation. "Long short-term memory" section explains the basic principle of LSTM and provides insights into how a recurrent neural network can be extended to model long-range temporal context via LSTM. In "Multi-stream model" section, we show how an LSTM-based phoneme predictor can be incorporated into a continuous speech recognition system based on HMMs.

Long short-term memory

A simple and widely used technique for context-sensitive sequence labeling based on neural networks is the application of *recurrent* neural networks. RNNs are able to model a certain amount of context by using cyclic connections and can in principle map from the entire *history* of previous inputs to each output. Yet, the analysis of the error flow in conventional recurrent neural nets resulted in the finding that long-range context is inaccessible to standard RNNs since the backpropagated error either blows up or decays over time (vanishing gradient problem Hochreiter et al. 2001). This led to various attempts to address the problem of vanishing gradients for RNN, including non-gradient-based training (Bengio et al. 1994), time-delay networks (Schaefer et al. 2008; Lin et al. 1996; Lang et al. 1990), hierarchical sequence compression (Schmidhuber 1992), and echo state networks (Jaeger 2001). One of the most effective techniques is the Long Short-Term Memory architecture (Hochreiter and Schmidhuber 1997), which is able to store information in linear memory cells over a longer period of time. They are able to overcome the vanishing gradient problem and can learn the optimal amount of contextual information relevant for the classification task.

An LSTM hidden layer is composed of multiple recurrently connected subnets which will be referred to as *memory blocks* in the following. Every memory block consists of self-connected *memory cells* and three multiplicative *gate* units (input, output, and forget gates). Since these gates allow for write, read, and reset operations within a memory block, an LSTM block can be interpreted as (differentiable) memory chip in a digital computer. Figure 1 shows the architecture of a memory block containing one memory cell.

If α_t^{in} denotes the activation of the input gate at time t *before* the activation function f_g has been applied and β_t^{in} represents the activation *after* application of the activation

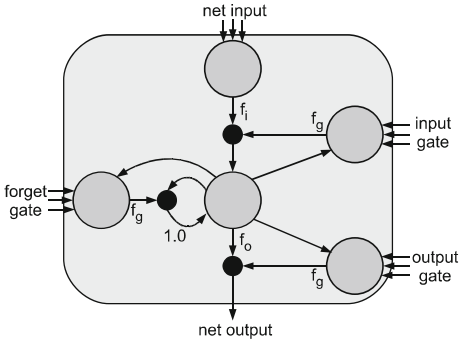

Fig. 1 LSTM memory block consisting of one memory cell: the input, output, and forget gates collect activations from inside and outside the block which control the cell through multiplicative units (depicted as *small circles*); input, output, and forget gate scale input, output, and internal state respectively; f_i, f_g, and f_o denote activation functions; the recurrent connection of fixed weight 1.0 maintains the internal state

function, the input gate activations (forward pass) can be written as

$$\alpha_t^{\text{in}} = \sum_{i=1}^{I} \eta^{i,\text{in}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{in}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{in}} s_{t-1}^c \tag{5}$$

and

$$\beta_t^{\text{in}} = f_g(\alpha_t^{\text{in}}), \tag{6}$$

respectively. The variable η^{ij} corresponds to the weight of the connection from unit i to unit j while 'in','for', and 'out' refer to input gate, forget gate, and output gate, respectively. Indices i, h, and c count the inputs x_t^i, the cell outputs from other blocks in the hidden layer, and the memory cells, while I, H, and C are the number of inputs, the number of cells in the hidden layer, and the number of memory cells in one block. Finally, s_t^c corresponds to the *state* of a cell c at time t, meaning the activation of the linear cell unit.

Similarly, the activation of the forget gates before and after applying f_g can be calculated as follows:

$$\alpha_t^{\text{for}} = \sum_{i=1}^{I} \eta^{i,\text{for}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{for}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{for}} s_{t-1}^c \tag{7}$$

$$\beta_t^{\text{for}} = f_g(\alpha_t^{\text{for}}). \tag{8}$$

The memory cell value α_t^{c} is a weighted sum of inputs at time t and hidden unit activations at time $t-1$:

$$\alpha_t^{\text{c}} = \sum_{i=1}^{I} \eta^{i,\text{c}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{c}} \beta_{t-1}^h. \tag{9}$$

To determine the current state of a cell c, we scale the previous state by the activation of the forget gate and the input $f_i(\alpha_t^{\text{c}})$ by the activation of the input gate:

$$s_t^c = \beta_t^{\text{for}} s_{t-1}^c + \beta_t^{\text{in}} f_i(\alpha_t^c). \quad (10)$$

The computation of the output gate activations follows the same principle as the calculation of the input and forget gate activations, however, this time we consider the *current* state s_t^c, rather than the state from the previous time step:

$$\alpha_t^{\text{out}} = \sum_{i=1}^{I} \eta^{i,\text{out}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{out}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{out}} s_t^c \quad (11)$$

$$\beta_t^{\text{out}} = f_g(\alpha_t^{\text{out}}). \quad (12)$$

Finally, the memory cell output is determined as

$$\beta_t^c = \beta_t^{\text{out}} f_o(s_t^c). \quad (13)$$

Note that the initial version of the LSTM architecture contained only input and output gates. Forget gates were added later (Gers et al. 2000) in order to allow the memory cells to reset themselves whenever the network needs to *forget* past inputs. In our experiments we exclusively consider the enhanced LSTM version including forget gates.

The overall effect of the gate units is that the LSTM memory cells can store and access information over long periods of time and thus avoid the vanishing gradient problem. For instance, as long as the input gate remains closed (corresponding to an input gate activation close to zero), the activation of the cell will not be overwritten by new inputs and can therefore be made available to the net much later in the sequence by opening the output gate.

In recent years, the LSTM technique has been successfully applied for a variety of pattern recognition tasks, including phoneme classification (Graves and Schmidhuber 2005), speech-based emotion recognition (Wöllmer et al. 2010), handwriting recognition (Graves et al. 2008), and driver distraction detection (Wöllmer et al. 2011).

Multi-stream model

To exploit LSTM-based phoneme recognition for keyword or continuous speech recognition tasks, the LSTM network can be combined with an HMM architecture that performs time warping in order to map from framewise phoneme estimates to words. The LSTM-HMM decoder applied in this article builds on preliminary experiments presented in Wöllmer et al. (2011) by employing a multi-stream architecture that simultaneously models low-level MFCC features and LSTM phoneme predictions. Since the SEMAINE system requires fully incremental real-time speech processing, this article focuses on *unidirectional* LSTM networks that exclusively consider past, and not future context (unlike the system presented in Wöllmer et al. (2011) which applies *bidirectional* LSTM and thus is not causal).

The general procedure for building our multi-stream LSTM-HMM system is as follows: First, we train an LSTM network on framewise phoneme targets so that it outputs a vector of phoneme likelihoods at every time step. Using this vector of posterior probabilities, we create a discrete feature that encodes the identity of the most likely phoneme (see Eq. 14). Finally, we train a continuous-discrete HMM system that uses both, MFCC features and the discrete BLSTM feature as observations. During testing, the role of the multi-stream HMM is to robustly map from (error-prone) LSTM phoneme estimates to sequences of words.

The structure of our multi-stream decoder can be seen in Fig. 2: s_t and x_t represent the HMM state and the acoustic (MFCC) feature vector, respectively, while b_t corresponds to the discrete phoneme prediction of the LSTM network (shaded nodes). Squares denote observed nodes and white circles represent hidden nodes. In every time frame t the HMM uses two independent observations: the MFCC features x_t and the LSTM phoneme prediction feature b_t. The vector x_t also serves as input for the LSTM, whereas the size of the LSTM input layer i_t corresponds to the dimensionality of the acoustic feature vector. The vector o_t contains one probability score for each of the P different phonemes at each time step. b_t is the index of the most likely phoneme:

$$b_t = \arg\max_j (o_{t,1}, \ldots, o_{t,j}, \ldots, o_{t,P}) \quad (14)$$

In every time step the LSTM generates a phoneme prediction according to Eq. 14 and the HMM models $x_{1:T}$

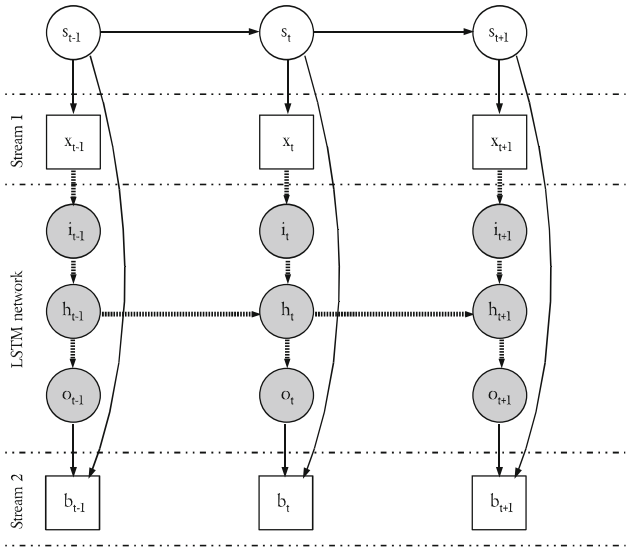

Fig. 2 Architecture of the multi-stream LSTM-HMM decoder: s_t: HMM state, x_t: acoustic feature vector, b_t: LSTM phoneme prediction feature, i_t, o_t, h_t: input, output, and hidden nodes of the LSTM network; *squares* correspond to observed nodes, *white circles* correspond to hidden nodes, *shaded circles* represent the LSTM network

and $b_{1:T}$ as two independent data streams. With $y_t = [x_t; b_t]$ being the joint feature vector consisting of continuous MFCC and discrete LSTM observations and the variable a denoting the stream weight of the first stream (i.e., the MFCC stream), the multi-stream HMM emission probability while being in a certain state s_t can be written as

$$p(y_t|s_t) = \left[\sum_{m=1}^{M} c_{s_t m}\mathcal{N}(x_t; \mu_{s_t m}, \Sigma_{s_t m})\right]^a \times p(b_t|s_t)^{2-a}. \quad (15)$$

Thus, the continuous MFCC observations are modeled via a mixture of M Gaussians per state while the LSTM prediction is modeled using a discrete probability distribution $p(b_t|s_t)$. Index m denotes the mixture component, $c_{st}m$ is the weight of the m'th Gaussian associated with state s_t, and $\mathcal{N}(\cdot; \mu, \Sigma)$ represents a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ. The distribution $p(b_t|s_t)$ is trained to model typical phoneme confusions that occur in the LSTM network.

The applied real-time LSTM phoneme predictor is publicly available as part of our on-line speech feature extraction engine openSMILE (Eyben et al. 2010).

Experiments

Both, LSTM-HMM decoding and HEQ combined with multi-condition training will be evaluated on the SEMAINE database which is detailed in "Databases" section, along with other corpora used for system training. In "Multi-condition training" section we will analyze the keyword spotting performance of a conventional single-stream HMM system applying models built via multi-condition training, while in "Histogram equalization" section we investigate the benefit of combining HEQ with multi-condition training to further reduce the mismatch between training and (noisy) test conditions. Finally, in "Multi-stream decoding" section, we focus on the effect of extending the single-stream HMM system to a multi-stream LSTM-HMM decoder as presented in "Multi-stream model" section.

Databases

The SEMAINE database was recorded in order to provide training material for the speech and vision-based input components of the SEMAINE system. For this purpose, the functionality of the virtual agent system was imitated by a human operator using a Wizard-of-Oz scenario. Thus, users were encouraged to show emotions while naturally speaking about arbitrary topics.

The transcribed part of the database consists of 19 recordings with different English speaking users and has a total length of 6.2 h. Models used for the experiments in "Experiments" section are trained on recordings 1–10 (speech material from both, user and operator) and tested on recordings 11–19 (only speech from the user). The vocabulary size of the SEMAINE corpus is 3.4 k.

In addition to the SEMAINE database, two other spontaneous speech corpora were used for acoustic and language model training: the SAL corpus and the COSINE corpus. The SAL database was recorded under similar conditions as the SEMAINE corpus, which makes it well-suited for our application scenario. It has already been used in a large number of studies on emotional speech (for more details on the SAL database, see Wöllmer et al. (2010), for example). The COSINE corpus (Stupakov et al. 2009) contains multi-party conversations recorded in real world environments and is partly overlaid with indoor and outdoor noise sources. It consists of ten transcribed sessions with 11.4 h of speech from 37 different speakers and has a vocabulary size of 4.8 k.

Multi-condition training

To improve keyword detection accuracy in noisy conditions, we investigated true positive and false positive rates when including noisy speech material in the training process. For all experiments, a part of the training material consisted of unprocessed versions of the SEMAINE database (recordings 1–10), the SAL corpus, and the COSINE database. This speech material will be referred to as *clean* in the ongoing (even though the COSINE corpus was partly recorded under noisy conditions). In addition to the 'clean' models, we evaluated different extensions of the training material by adding distorted versions of the SEMAINE and the SAL corpora. For this purpose, we superposed the clean speech with additive noise at different SNR levels: 15 dB, 10 dB, and 5 dB. We considered both, white Gaussian noise and babble noise from the NOISEX database. For evaluation, we used clean and distorted versions of the SEMAINE database (recordings 11–19). Since conversational agents such as the SEMAINE system are often used while other people talk in the background, the babble noise evaluation scenario is most relevant for our application. We considered a set of 173 keywords which are relevant for the dialogue management of the SEMAINE system. Further, we modeled three different non-linguistic vocalizations (*breathing, laughing*, and *sighing*). The training/test set distribution for *breathing, laughing*, and *sighing* was 124/54, 268/227, and 45/8, respectively. Keyword detection was based on simply searching for the respective words in the most likely ASR hypothesis. The applied trigram language model was trained on the SEMAINE corpus (recordings 1–10), the SAL database, and the COSINE database (total vocabulary size 6.1 k). The choice

of applying an LVCSR system including a language model that is built from SEMAINE data as basis for keyword spotting was motivated by preliminary experiments reported in Wöllmer et al. (2011) which showed that a full ASR system with in-domain language model prevails over vocabulary independent techniques. Via openSMILE (Eyben et al. 2010), 13 cepstral mean normalized MFCC features along with first and second order temporal derivatives were extracted from the speech signals every 10 ms using a window size of 25 ms. MFCC features were chosen since previous studies showed that the application of Perceptual Linear Prediction (PLP) features does not result in enhanced noise robustness (Schuller et al. 2009). All cross-word triphone HMMs consisted of 3 emitting states with 16 Gaussian mixtures per state while for non-linguistic vocalizations, we trained HMMs consisting of 9 states. For HMM decoding, we applied the Julius toolkit (Lee and Kawahara 2009).

Figure 3a–d show the Receiver Operating Characteristic (ROC) operating points for clean test material as well as for speech superposed with babble noise at 15 dB, 10 dB, and 5 dB SNR, respectively, when using different acoustic models. Note that since our keyword spotting approach is based on an LVCSR system generating a single word hypothesis (rather than on a technique that outputs confidences for each possible keyword), we obtain single ROC operating points and not ROC *curves*. In other words, our recognition framework does not include a confidence threshold that can be varied in order to adjust the ROC operating point. However, in the light of our target application being conversational agents, the resulting moderate false positive rates are desired since a high number of false alarms tends to be more critical than missing keywords.

As can be seen in Fig. 3a, models exclusively trained on clean speech lead to the best performance for clean test data. We obtain a true positive rate of 56.58% at a false positive rate of 1.89% which is in the range of typical recognition rates for highly disfluent, spontaneous, and emotionally colored speech (Wöllmer et al. 2010). Including noisy training material slightly increases the false positive rate to up to 2.20% at a small decrease of true positive rates. Yet, when evaluating the models on speech superposed by babble noise, multi-condition training significantly increases the true positive rates. A good compromise between high true positive rates and low false positive rates in noisy conditions can be obtained by

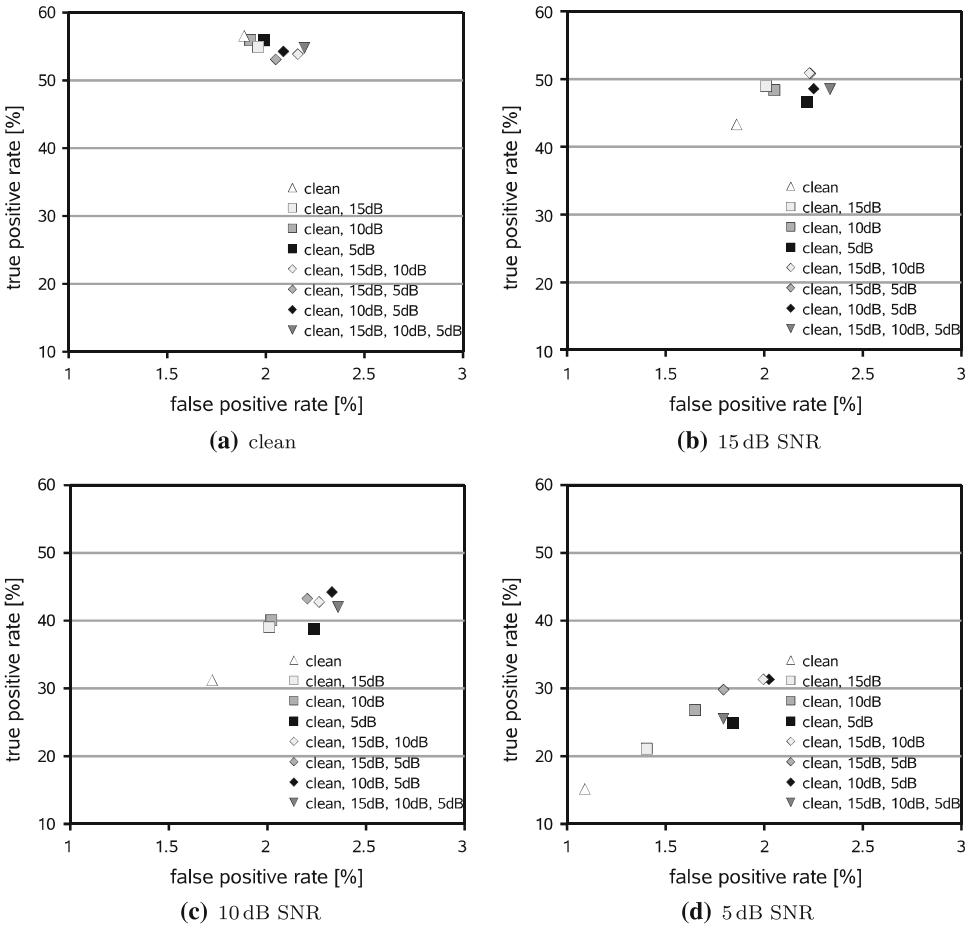

Fig. 3 ROC operating points obtained for different acoustic models when tested on clean speech and speech superposed by babble noise at 15, 10, and 5 dB SNR; acoustic models were trained on unprocessed versions of the SEMAINE, SAL, and COSINE corpora ('clean') and on noisy versions of the SEMAINE and SAL corpora using different SNR level combinations (babble noise)

applying the acoustic models denoted as 'clean, 15, 10 dB' in Fig. 3a–d, i.e., models trained on the clean versions of the SEMAINE, SAL, and COSINE corpora, on the SEMAINE and SAL corpora superposed by babble noise at 15 dB SNR, and on the 10 dB versions of the SEMAINE and SAL databases. For test data superposed by babble noise, this training set combination leads to the highest average true positive rate (41.66%, see Table 1, upper part) at a tolerable average false positive rate. A similar result can be observed for the evaluation on test data corrupted by white noise (see Table 1, lower part). Models that are partly trained on speech superposed by white noise enable higher true positive rates in noisy conditions than 'clean' models. As for the babble noise scenario, a combination of clean, 15 dB SNR, and 10 dB SNR training data results in the best true positive / false positive compromise.

Histogram equalization

Applying the same multi-condition training set combinations as in "Multi-condition training" section, we repeated the keyword spotting experiments using MFCC features normalized via HEQ instead of the cepstral mean normalized features. HEQ was performed as described in "Histogram equalization" section and applied to all 39 feature vector components during training and testing. Histograms were estimated from sequences of feature vectors representing one utterance. The corresponding results for clean test data and for test data superposed by babble noise at 15, 10, and 5 dB SNR can be seen in Fig. 4a–d. When comparing Fig. 3a–d with 4a–d, we can see that HEQ further improves true positive rates in noisy conditions while false positive rates are slightly increased. The average keyword spotting performance when evaluating the different training set combinations together with HEQ on test data corrupted by babble noise and white noise, respectively, can be seen in the right half of Table 1. Again, the best training set combination when testing on speech distorted by babble noise is to use clean speech combined with 15 and 10 dB SNR versions of the data. With HEQ, an average true positive rate of 45.70% can be obtained for this case, which corresponds to a 4% absolute performance improvement compared to non-HEQ models (41.66%). Also for the other training set combinations an absolute true positive rate improvement of between 3.7 and 6.9% can be observed for noisy test data. When evaluating the white noise scenario we also consistently obtain higher true positive rates by HEQ (up to 44.48%, see Table 1).

Table 1 Average true positive rates (tpr) and average false positive rates (fpr) obtained with acoustic models trained on clean data and speech superposed by babble/white noise at different SNR conditions; clean and noisy test condition; with and without histogram equalization

Training data	Test condition							
	Without HEQ				With HEQ			
	Noisy		Clean		Noisy		Clean	
SNR (dB)	tpr (%)	fpr (%)	tpr (%)	fpr (%)	tpr (%)	fpr (%)	tpr (%)	fpr (%)
Babble noise								
Clean	29.89	1.56	56.58	1.89	29.89	1.56	56.58	1.89
Clean, 15	36.37	1.81	54.91	1.96	42.60	2.38	55.59	1.96
Clean, 10	38.40	1.91	55.92	1.92	43.90	2.24	53.79	1.91
Clean, 5	36.73	2.10	55.90	1.99	43.67	2.21	54.56	1.92
Clean, 15, 10	**41.66**	2.16	53.87	2.16	**45.70**	2.53	56.04	2.10
Clean, 15, 5	41.29	2.08	53.08	2.05	45.37	2.43	54.94	2.10
Clean, 10, 5	41.38	2.20	54.28	2.09	45.21	2.34	53.69	2.04
Clean, 15, 10, 5	38.67	2.16	54.79	2.20	42.33	2.39	53.89	2.17
White noise								
Clean	19.81	1.26	56.58	1.89	19.81	1.26	56.58	1.89
Clean, 15	39.40	2.28	57.31	2.06	43.45	2.63	55.35	1.99
Clean, 10	39.33	2.44	56.50	2.03	42.94	2.53	54.40	2.00
Clean, 5	23.65	1.20	56.02	2.01	40.72	2.46	52.90	1.98
Clean, 15, 10	**42.47**	2.54	54.55	2.21	44.09	2.73	53.24	2.20
Clean, 15, 5	42.11	2.57	54.28	2.16	43.53	2.61	53.70	2.09
Clean, 10, 5	41.27	2.60	54.09	2.04	43.62	2.60	53.90	2.15
Clean, 15, 10, 5	**42.48**	2.69	50.42	2.27	**44.48**	2.79	53.48	2.29

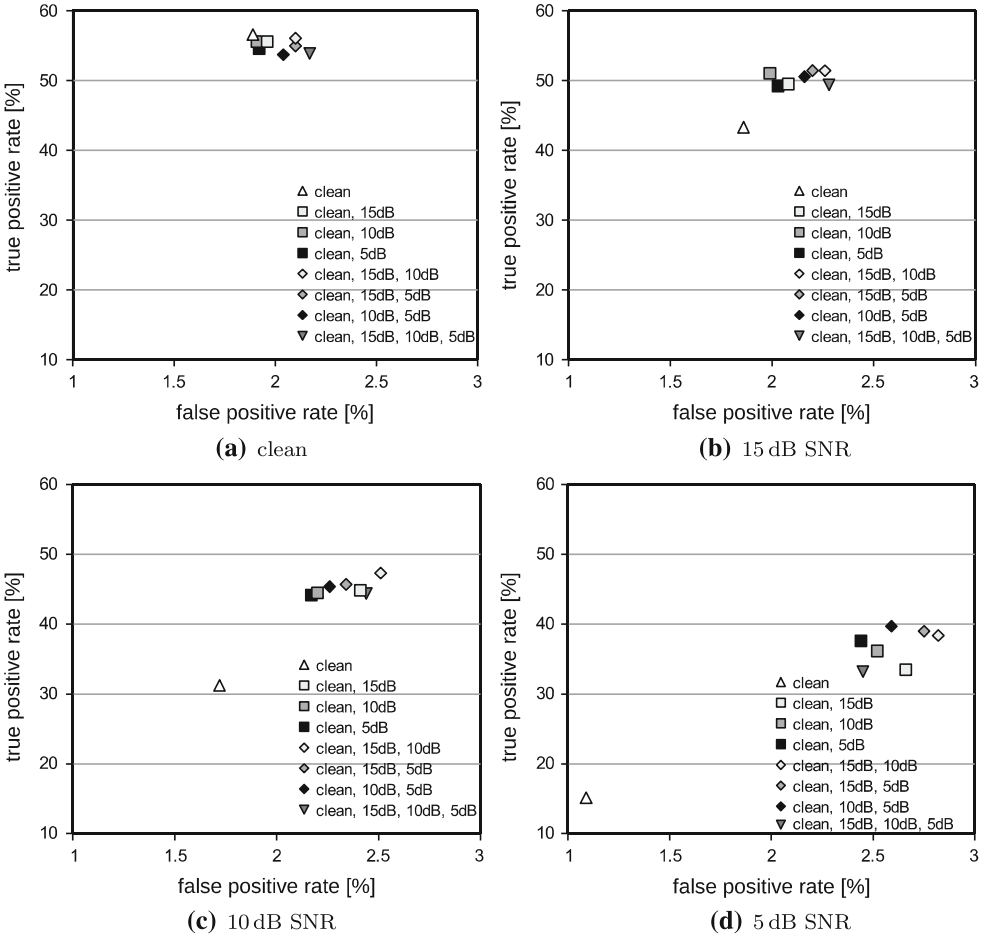

Fig. 4 ROC operating points obtained for different acoustic models when tested on clean speech and speech superposed by babble noise at 15, 10, and 5 dB SNR in combination with histogram equalization; acoustic models were trained on unprocessed versions of the SEMAINE, SAL, and COSINE corpora ('clean') and on noisy versions of the SEMAINE and SAL corpora using different SNR level combinations (babble noise)

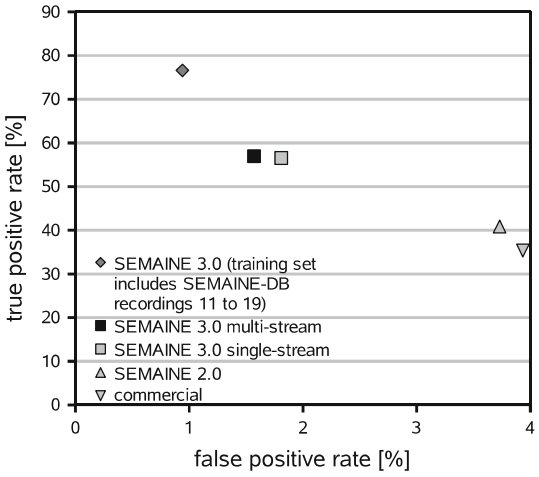

Fig. 5 ROC operating points obtained for different variants of the SEMAINE keyword detector

Multi-stream decoding

Next, we implemented and evaluated the multi-stream LSTM-HMM decoder introduced in "Multi-stream model" section to improve keyword detection. Since the LSTM network was trained on framewise phoneme targets, we used an HMM system to obtain phoneme borders via forced alignment. Techniques that carry out time warping within the neural network (e. g., Connectionist Temporal Classification Graves et al. 2006) were not employed to to their high computational complexity. The multi-stream system was trained on the clean versions of the SEMAINE, SAL, and COSINE databases and applied an LSTM network with a hidden layer consisting of 128 memory blocks. Each memory block contained one memory cell.

For LSTM network training we used a learning rate of 10^{-5} and a momentum of 0.9. Prior to the training process, all weights were randomly initialized in the range from -0.1 to 0.1. Input and output gates used tanh activation functions, while the forget gates had logistic activation functions. Similarly to the HMM recognizer, the the networks were trained on the standard (CMU) set of 39 different English phonemes with additional targets for *silence, breathing, laughing*, and *sighing*. The stream weight variable a was set to one.

The ROC operating points representing the keyword detection performance of the standard HMM (SEMAINE 3.0 single-stream) and the LSTM-HMM (SEMAINE 3.0 multi-

Table 2 True positive rates (tpr) and false positive rates (fpr) obtained with HMMs and with the LSTM-HMM architecture trained on clean data and evaluated on speech superposed by babble/white noise at different SNR conditions

Test condition	Babble noise				White noise			
	HMM		LSTM-HMM		HMM		LSTM-HMM	
SNR	tpr (%)	fpr (%)	tpr (%)	fpr (%)	tpr (%)	fpr (%)	tpr (%)	fpr (%)
Clean	56.58	1.89	57.26	1.57	56.58	1.89	57.26	1.57
15 dB	43.30	1.86	42.51	1.54	29.95	1.69	31.39	1.77
10 dB	31.23	1.72	28.62	1.20	19.79	1.32	22.77	1.42
5 dB	15.15	1.09	11.77	0.71	9.71	0.77	13.14	0.82

stream) can be seen in Fig. 5. All systems were evaluated on clean recordings 11–19 from the SEMAINE database. At a slight increase of the true positive rate (57.26% vs. 56.58%), the incorporation of LSTM phoneme predictions can reduce the false positive rate from 1.89 to 1.57%. For comparison, we also included the results for a preliminary version of the SEMAINE keyword detector (referred to as the SEMAINE 2.0 system Schröder et al. 2008) which does not apply an in-domain language model and thus cannot compete with the current version (SEMAINE 3.0). Figure 5 also shows the performance obtained with a commercial recognizer as used in (Principi et al. 2009). The comparably low performance of the commercial system indicates that using acoustic models tailored for the recognition of emotionally colored speech is essential for virtual agent applications such as the SEMAINE system. Since the final SEMAINE 3.0 keyword detector is trained on the *whole* SEMAINE database (including recordings 11–19), Fig. 5 also shows the ROC performance obtained with models trained on all SEMAINE data. Note, however, that this configuration does not allow for a realistic performance assessment since training and test sets are not disjoint in this case.

Table 2 shows how the keyword spotting performance of the HMM system and the multi-stream LSTM-HMM technique is affected if test data is superposed by babble and white noise at different SNR levels. Similar to the clean case, the integration of LSTM leads to a remarkable reduction of the false positive rate if speech is corrupted by babble noise. However, also the true positive rate is slightly lower for the multi-stream system in the babble noise scenario. By contrast, for speech distorted by white noise, the LSTM-HMM consistently reaches higher true positive rates (increase of 1.4, 3.0, and 3.4% for 15, 10, and 5 dB, respectively) at a small increase of the false positive rate (around 0.1% for all SNR levels).

Conclusion

We investigated different techniques to improve keyword spotting performance in challenging conditions. Motivated by the SEMAINE scenario, i.e., an emotionally sensitive conversational agent application, we considered the task of reliably detecting keywords in the SEMAINE database which consists of spontaneous, disfluent, and partly noisy speech and reflects the conditions a real-time keyword spotter incorporated into a cognitive agent system has to face. Since the SEMAINE system is frequently used when people talk in the background, our evaluations were focused on conversational keyword detection in speech superposed by babble noise. To reduce the mismatch between training and (noisy) test conditions, we investigated both, feature enhancement and model adaptation. Feature enhancement was realized via HEQ, a technique that is well-suited for compensating non-linear distortions in the feature speace caused by noise, while models were adapted by multi-condition training. We showed that our proposed combination of HEQ feature enhancement and multi-condition training results in improved keyword detection performance for different noise types and SNR levels.

Furthermore, we implemented and evaluated a novel multi-stream LSTM-HMM architecture with respect to its keyword detection accuracy. The model is composed of a LSTM neural network for context-sensitive phoneme estimation and a multi-stream HMM for dynamic decoding. We found that compared to a single-stream HMM system, the LSTM-HMM technique leads to more accurate keyword detection for most noise scenarios, which is in line with preliminary experiments on continuous speech recognition applying LSTM (Wöllmer et al. 2011).

Future studies should consider alternative feature enhancement approaches such as model-based enhancement with Switching Linear Dynamic Models (Droppo and Acero 2004) in combination with multi-conditon training as well as the combination of HEQ and multi-stream decoding. Furthermore, it might be interesting to examine the potential of LSTM networks for RNN-based feature enhancement, e.g., by training networks that map from noisy to clean speech features as in Parveen and Green (2004). As far as the SEMAINE scenario is concerned, further keyword detection improvements could be possible

if audio and video feature are merged, e.g., via hybrid fusion (Wöllmer et al. 2009), or if multiple audio channels are used, exploiting multichannel feature enhancement (Principi et al. 2010; Rotili et al. 2011).

Acknowledgments The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211486 (SEMAINE) and from the Federal Republic of Germany through the German Research Foundation (DFG) under grant no. SCHU 2508/4-1.

References

Ananthakrishnan S, Narayanan S (2007) Improved speech recognition using acoustic and lexical correlates of pitch accent in a n-best rescoring framework. In: Proceedings of ICASSP. Honolulu, pp 873–876

Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166

de la Torre A, Peinado AM, Segura JC, Perez-Cordoba JL, Benitez MC, Rubio AJ (2005) Histogram equalization of speech representation for robust speech recognition. IEEE Trans Speech Audio Process 13(3):355–366

Deng J, Bouchard M, Yeap TH (2007) Noisy speech feature estimation on the Aurora2 database using a switching linear dynamic model. J Multimedia 2(2):47–52

Droppo J, Acero A (2004) Noise robust speech recognition with a switching linear dynamic model. In: Proceedings of ICASSP. Montreal, Canada

Eyben F, Wöllmer M, Schuller B (2010) openSMILE—the Munich versatile and fast open-source audio feature extractor. In: Proceedings of ACM Multimedia. Firenze, pp 1459–1462

Fernandez S, Graves A, Schmidhuber J (2007) An application of recurrent neural networks to discriminative keyword spotting. In: Proceedings of ICANN. Porto, pp 220–229

Gers F, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. Neural Comput 12(10):2451–2471

Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5-6):602–610

Graves A, Fernandez S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: labelling unsegmented data with recurrent neural networks. In: Proceedings of ICML. Pittsburgh, pp 369–376

Graves A, Fernandez S, Liwicki M, Bunke H, Schmidhuber J (2008) Unconstrained online handwriting recognition with recurrent neural networks. Adv Neural Inform Process Syst 20:1–8

Hilger F, Ney H (2006) Quantile based histogram equalization for robust large vocabulary speech recognition. IEEE Trans Audio Speech Language Process 14(3):845–854

Hirsch HG, Pearce D (2000) The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In: ISCA ITRW ASR2000: automatic speech recognition: challenges for the next millennium. Paris

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

Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: Kremer SC, Kolen JF (eds) A field guide to dynamical recurrent neural networks. pp 1–15 IEEE Press, New York, (2001)

Hussain A, Campbell D (1998) Binaural sub-band adaptive speech enhancement using artificial neural networks. Speech Commun 25(1–3):177–186

Jaeger H (2001) The echo state approach to analyzing and training recurrent neural networks. Technical report, Bremen: German National Research Center for Information Technology (Tech. Rep. No. 148)

Ketabdar H, Vepa J, Bengio S, Boulard H (2006) Posterior based keyword spotting with a priori thresholds. In: IDAIP-RR, pp 1–8

Lang KJ, Waibel AH, Hinton GE (1990) A time-delay neural network architecture for isolated word recognition. Neural Netw 3(1): 23–43

Lathoud G, Magimia-Doss M, Mesot B, Boulard H (2005) Unsupervised spectral subtraction for noise-robust ASR. In: Proceedings of ASRU. San Juan, Puerto Rico

Lee A, Kawahara T (2009) Recent development of open-source speech recognition engine julius. In: Proceedings of APSIPA ASC

Lin T, Horne BG, Tino P, Giles CL (1996) Learning long-term dependencies in NARX recurrent neural networks. IEEE Trans Neural Netw 7(6):1329–1338

Mamou J, Ramabhadran B, Siohan O (2007) Vocabulary independent spoken term detection. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. Amsterdam, pp 615–622

McTear MF (2002) Spoken dialogue technology: enabling the conversational user interface. ACM Comput Surv 34(1):90–169

Memon ZA, Treur J (2010) On the reciprocal interaction between believing and feeling: an adaptive agent modelling perspective. Cogn Neurodyn 4(4):377–394

Mesot B, Barber D (2007) Switching linear dynamic systems for noise robust speech recognition. IEEE Trans Audio Speech Language Process 15(6):1850–1858

Parveen S, Green P (2004) Speech enhancement with missing data techniques using recurrent neural networks. In: Proceedings of ICASSP. Montreal

Principi E, Cifani S, Rocchi C, Squartini S, Piazza F (2009) Keyword spotting based system for conversation fostering in tabletop scenarios: preliminary evaluation. In: Proceedings of HSI. Catania, pp 216–219

Principi E, Cifani S, Rotili R, Squartini S (2010) Comparative evaluation of single-channel MMSE-based noise reduction schemes for speech recognition. J Elec Comput Eng 2010:21:1–21:7

Rotili R, Principi E, Cifani S, Squartini S, Piazza F (2011) Multichannel feature enhancement for robust speech recognition. In: Ipsic I (eds) Speech technologies. InTech, ISBN: 978-953-307-996-7. Available from: http://www.intechopen.com/articles/show/title/multichannel-feature-enhancement-for-robust-speech-recognition

Schaefer AM, Udluft S, Zimmermann HG (2008) Learning long-term dependencies with recurrent neural networks. Neurocomputing 71(13-15):2481–2488

Schmidhuber J (1992) Learning complex extended sequences using the principle of history compression. Neural Comput 4(2):234–242

Schröder M, Cowie R, Heylen D, Pantic M, Pelachaud C, Schuller B (2008) Towards responsive sensitive artificial listeners. In: Proceedings. of 4th international workshop on human-computer conversation. Bellagio, pp 1–6

Schuller B, Wöllmer M, Moosmayr T, Rigoll G (2008) Speech recognition in noisy environments using a switching linear dynamic model for feature enhancement. In: Proceedings of interspeech. Brisbane, pp 1789–1792

Schuller B, Wöllmer M, Moosmayr T, Rigoll G (2009) Recognition of noisy speech: a comparative survey of robust model architecture and feature enhancement. J Audio Speech Music Process. ID 942617

Squartini S, Fagiani M, Principi E, Piazza F (2011) Multichannel cepstral domain feature warping for robust speech recognition. Front Artif Intell Appl 226:284–292

Stupakov A, Hanusa E, Bilmes J, Fox D (2009) COSINE—a corpus of multi-party conversational speech in noisy environments. In: Proceedings of ICASSP. Taipei

Windmann S, Haeb-Umbach R (2008) Modeling the dynamics of speech and noise for speech feature enhancement in ASR. In: Proceedings of ICASSP. Las Vegas

Wöllmer M, Al-Hames M, Eyben F, Schuller B, Rigoll G (2009) A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. Neurocomputing 73(1-3):366–380

Wöllmer M, Eyben F, Keshet J, Graves A, Schuller B, Rigoll G (2009) Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks. In: Proceedings of ICASSP. Taipei, pp 3949–3952

Wöllmer M, Eyben F, Graves A, Schuller B, Rigoll G (2010) Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework. Cogn Comput 2(3):180–190

Wöllmer M, Schuller B, Eyben F, Rigoll G (2010) Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening. IEEE J Select Topics Sig Process 4(5):867–881

Wöllmer M, Blaschke C, Schindl T, Schuller B, Färber B, Mayer S, Trefflich B (2011a) On-line driver distraction detection using long short-term memory. IEEE Trans Intell Transport Syst 12(2):574–582

Wöllmer M, Eyben F, Schuller B, Rigoll G (2011b) A multi-stream ASR framework for BLSTM modeling of conversational speech. In: Proceedings of ICASSP. Prague, pp 4860–4863

Wöllmer M, Marchi E, Squartini S, Schuller B (2011c) Robust multi-stream keyword and non-linguistic vocalization detection for computationally intelligent virtual agents. In: Proceedings of ISNN. Guilin, pp 496–505