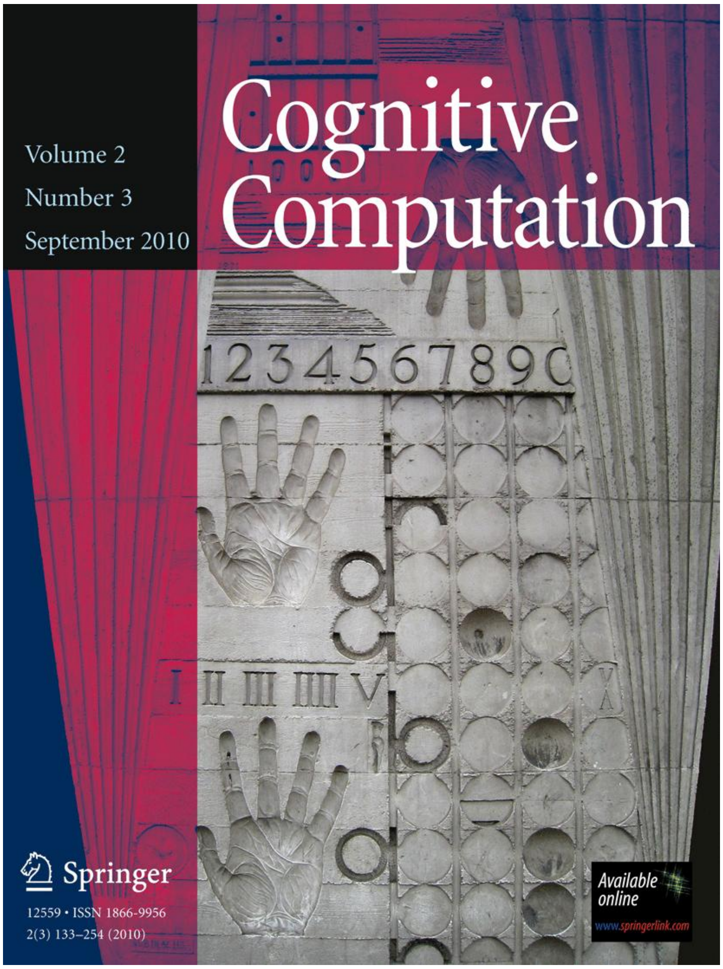
ISSN 1866-9956, Volume 2, Number 3



Bidirectional LSTM Networks for Context-Sensitive Keyword Detection in a Cognitive Virtual Agent Framework

Martin Wöllmer · Florian Eyben · Alex Graves · Björn Schuller · Gerhard Rigoll

Published online: 27 April 2010

Abstract Robustly detecting keywords in human speech is an important precondition for cognitive systems, which aim at intelligently interacting with users. Conventional techniques for keyword spotting usually show good performance when evaluated on well articulated read speech. However, modeling natural, spontaneous, and emotionally colored speech is challenging for today's speech recognition systems and thus requires novel approaches with enhanced robustness. In this article, we propose a new architecture for vocabulary independent keyword detection as needed for cognitive virtual agents such as the SEMAINE system. Our word spotting model is composed of a Dynamic Bayesian Network (DBN) and a bidirectional Long Short-Term Memory (BLSTM) recurrent neural net. The BLSTM network uses a self-learned amount of contextual information to provide a discrete phoneme prediction feature for the DBN, which is able to distinguish between keywords and arbitrary speech. We evaluate our Tandem BLSTM-DBN technique on both read speech and spontaneous emotional speech and show that our method significantly outperforms conventional Hidden Markov Model-based approaches for both application scenarios.

Keywords Keyword spotting · Long short-term memory · Dynamic bayesian networks · Cognitive systems · Virtual agents

M. Wöllmer (✉) · F. Eyben · B. Schuller · G. Rigoll
Institute for Human-Machine Communication, Technische Universität München, Arcisstrasse 21, 80290 München, Germany
e-mail: woellmer@tum.de

A. Graves
Institute for Computer Science VI, Technische Universität München, Boltzmannstrasse 3, 85748 München, Germany

Introduction

In recent years, the design of cognitive systems with the ability to perceive, learn, memorize, decide, act, and communicate has attracted a lot of attention [1]. For natural interaction with intelligent systems, human speech has become one of the most important input modalities [2, 3]. Thus, in this article, we focus on extracting information from speech as an essential perception capability of cognitive virtual agents. Since full spoken language understanding without any restriction of the expected vocabulary is hardly feasible and not necessarily needed in today's human-machine interaction scenarios (e. g. [4]), most systems apply keyword spotting as an alternative to large vocabulary continuous speech recognition. The aim of keyword spotting is to detect a set of predefined keywords from continuous speech signals [5]. When applied in human-like cognitive systems, keyword detectors have to process natural and spontaneous speech, which in contrast to well articulated read speech (as used in [6], for example) leads to comparatively low recognition rates [7]. Since modeling emotion and including linguistic models for affect recognition [8, 9] plays a major role in the design of cognitive systems [10, 1], keyword spotters also need to be robust with respect to emotional coloring of speech. A typical scenario for a cognitive emotionally sensitive virtual agent system that requires keyword detection in emotionally colored speech is the SEMAINE system [4]. Such application areas demand for highly robust speech modeling and highlight the importance of the exploration of non-conventional speech processing approaches.

At present, the predominant methodology for keyword spotting is using Hidden Markov Models (HMM) [11, 12, 13]. However, a major problem with HMM based systems is that they are forced to model the *garbage* (i.e. non-keyword) parts of the signal as well as the keywords themselves. This is difficult because a structure flexible enough to model all possible garbage words is likely to be able to model the keywords as well. For example, if phoneme level models are used, then garbage parts can be accurately captured by a model that connects all possible phonemes [11]; however, such a model will also fit the keywords. One solution is to use whole word models for both, garbage and keywords, but this requires that all the keywords occur many times in the training corpus, and also means that new keywords cannot be added without training new models. Consequently, such a system would be less flexible than a vocabulary independent system [14].

The keyword detection technique introduced in this article overcomes these drawbacks by using a phoneme based recognition system with no explicit garbage model. The architecture is robust to phoneme recognition errors, and unlike methods based on large vocabulary speech recognizers (such as [15], for example), it does not require a language model: only the keyword phonemizations are needed.

In our system, the distinction between keywords and other speech is made by a Dynamic Bayesian Network (DBN), using a hidden garbage variable and the concept of *switching parents* [16]. DBNs (and other graphical models) offer a flexible statistical framework that is increasingly applied to speech recognition tasks [16, 17]. Graphical models (GM) make use of the graph theory in order to describe the time evolution of speech as a statistical process and thereby define conditional independence properties of the observed and hidden variables that are involved in the process of speech decoding. Apart from common HMM approaches, there exist only a small number of works which try to address the task of keyword spotting using the graphical model paradigm. In [18] a graphical model is applied for spoken keyword spotting based on performing a joint alignment between the phone lattices generated from a spoken query and a long stored utterance. This concept, however, is optimized for offline phone lattice generation and bears no similarity to the technique proposed herein. The same holds for approaches towards GM based out-of-vocabulary (OOV) detection [19] where a graphical model indicates possible OOV regions in continuous speech.

The graphical model structure presented in this article can be seen as an extension of the GM for keyword spotting that we introduced in [20]. Unlike the model described in [20], our Tandem approach is not only based on Gaussian mixture modeling but additionally applies a recurrent neural network (RNN) to provide improved phoneme predictions, which can then be incorporated into the DBN. The RNN uses the bidirectional Long Short-Term Memory (BLSTM) architecture [21] to access long-range context information along both input directions. BLSTM has been proven to outperform standard methods of modeling context such as triphone HMMs [21] and was found to be well suited for spontaneous, emotional speech [7]. In the area of cognitive systems, BLSTM networks have been successfully applied for emotion recognition [22] from low-level framewise audio features, which also requires the modeling of long-range context.

Tandem or hybrid architectures that combine discriminatively trained neural networks with graphical models such as HMMs are widely used for speech recognition, and their popularity has grown in recent years [23–27]. However, BLSTM is a relatively new architecture that has so far been applied to keyword spotting only in a few works: in [7] the framewise phoneme predictions of BLSTM were shown to enhance the performance of a discriminative keyword spotter [6]; and in [28] a keyword spotter using BLSTM for whole-word modeling was introduced. Yet both approaches significantly differ from the concept introduced in this article and were found to be unsuited for flexible real time keyword detection in a cognitive virtual agent framework. Unlike the model proposed herein, the discriminative approach in [7] does not apply Markov chains to model the temporal evolution of speech, but maps the acoustic representation of an utterance along with the target keyword into an abstract vector space, using a set of feature functions that provide confidence scores based on the output of framewise phoneme classifiers. This strategy, however, is rather suited for off-line keyword search than for on-line applications since it does not operate in real-time. The disadvantage of the method proposed in [28] is that it has a separate output unit for each keyword, which requires excessive amounts of training data for large vocabularies, and also means the network must be retrained when new keywords are added.

The aim of this work is to combine the high-level flexibility of graphical models with the low-level signal processing power of BLSTM in order to create a context-sensitive keyword detector that can cope with spontaneous, emotional speech and thus can be applied in a cognitive virtual agent framework. We evaluate our system on read speech from the TIMIT corpus [29] as well as on natural and emotionally colored speech from the Sensitive Artificial Listener (SAL) corpus—a database that was recorded using a Wizard-of-Oz SAL interface designed to let users work through a range of emotional states [30]. Thereby, we compare the keyword spotting accuracy of our Tandem BLSTM-DBN system to a conventional HMM-based approach and to a hybrid BLSTM-HMM technique. Further, we investigate the benefit of incorporating BLSTM phoneme prediction for context-sensitive speech modeling.

The structure of this article is as follows: Sect. 2 briefly introduces the virtual agents used in the SEMAINE project for which our keyword spotter was developed. In Sect. 3, the principle of context modeling via Long Short-Term Memory is explained, while Sect. 4 outlines the graphical model architecture of our Tandem BLSTM-DBN recognizer. Experimental results are presented in Sect. 5 before concluding in Sect. 6.

The SEMAINE Characters

The aim of the SEMAINE project[1] is to build a Sensitive Artificial Listener—a multimodal dialogue system with the social interaction skills needed for a sustained conversation with a human user. Thereby the user can speak to four different virtual characters, each of whom represents a certain emotional state: 'Prudence' is matter-of-fact, 'Poppy' is cheerful, 'Obadiah' is pessimistic, and 'Spike' is aggressive. During the conversations, all virtual characters aim to induce an emotion that corresponds to *their* typical affective state. All characters encourage the user to speak naturally about different topics while trying to recognize and interpret the user's facial expressions, emotions, as well as relevant keywords. The recognition output is used to react, e. g. to the user's emotion or to certain keywords.

Important preconditions for a keyword spotter applied within the SEMAINE system are besides real-time operation also robustness with respect to emotional coloring of speech and flexibility as far as changes in the keyword vocabulary are concerned. Generally, vocabulary independent systems are preferable since the keyword vocabulary often changes during the design of the virtual agent system. A description of the overall architecture of the SEMAINE system can be found in [4].

Long Short-Term Memory

Since context modeling via Long Short-Term Memory [31] networks was found to enhance keyword spotting performance in natural conversation scenarios [7] as recorded in the SAL database, our keyword spotter uses framewise phoneme predictions computed by a bidirectional LSTM net (see Sect. 4). Thus, this section outlines the basic principle of the Long Short-Term Memory RNNs.

Framewise phoneme prediction presumes a classifier that can access and model long-range context, since due to co-articulation effects in human speech, neighboring phonemes influence the cepstral characteristics of a given phoneme [32, 33]. Consequently, when attempting to predict phonemes frame by frame, a number of preceding (and successive) speech frames have to be taken into account in order to capture relevant speech characteristics. The *number* of speech frames that should be used to obtain enough context for reliably estimating phonemes is hard to determine. Thus, a classifier that is able to *learn* the amount of context is a promising alternative to manually defining fixed time windows. Static techniques such as Support Vector Machines do not explicitly model context but rely on either capturing contextual information via statistical functionals of features [34] or aggregating frames using Multi-Instance Learning techniques [35]. Dynamic classifiers like Hidden Markov Models are often applied for time warping and flexible context modeling, using e. g. triphones or quinphones. Yet, HMMs have drawbacks such as the inherent assumption of conditional independence of successive observations, meaning that an observation is statistically independent of past ones, provided that the values of the hidden variables are known. Hidden Conditional Random Fields (HCRF) [36] are one attempt to overcome this limitation. However, also HCRF offer no possibility to model a self-learned amount of contextual information. Other dynamic classifiers such as neural networks are able to model a certain amount of context by using cyclic connections. These so-called recurrent neural networks 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 led to 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 [37]). This led to various attempts to address the problem of vanishing gradients for RNN, including non-gradient-based training [38], time-delay networks [39, 40, 41], hierarchical sequence compression [42], and echo state networks [43]. One of the most effective techniques is the Long Short-Term Memory architecture [31], 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 layer is composed of recurrently connected memory blocks, each of which contains one or more memory cells, along with three multiplicative 'gate' units: the input, output, and forget gates. The gates perform functions analogous to read, write, and reset operations. More specifically, the cell input is multiplied by the activation of the input gate, the cell output by that of the output gate, and the previous cell values by the forget gate (see Fig. 1). The overall effect is to allow the network to store and retrieve information over long periods of time. For example, as long as the input gate remains closed, the activation of the cell will not be overwritten by new inputs

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

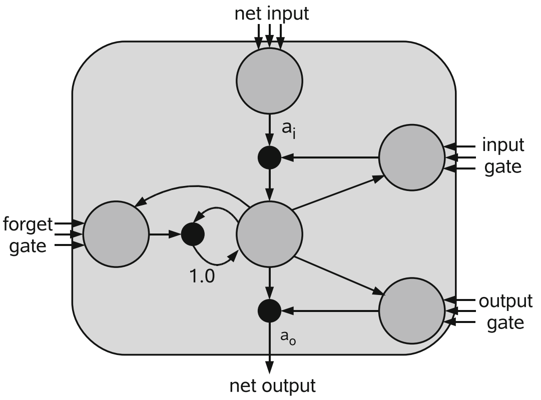


Fig. 1 LSTM memory block consisting of one memory cell: the input, output, and forget gates collect activations from inside and outside the block, which control the cell through multiplicative units (depicted as *small circles*); input, output, and forget gate scale input, output, and internal state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

and can therefore be made available to the net much later in the sequence by opening the output gate.

Another problem with standard RNNs is that they have access to past but not to future context. This can be overcome by using bidirectional RNNs [44], where two separate recurrent hidden layers scan the input sequences in opposite directions. The two hidden layers are connected to the same output layer, which therefore has access to context information in both directions. The amount of context information that the network actually uses is learned during training, and does not have to be specified beforehand. Thereby, forward and backward context are learned independently from each other. Bidirectional networks can be applied whenever the sequence processing task is not truly online (meaning the output is not required after every input) which makes them popular for speech recognition tasks where the output has to be present e. g. at the end of a sentence [21]. However, often a small buffer is enough in order to profit from bidirectional context, so that bidirectional networks can also be applied for causal systems whenever a short output latency is tolerable. Figure 2 shows the structure of a simple bidirectional network.

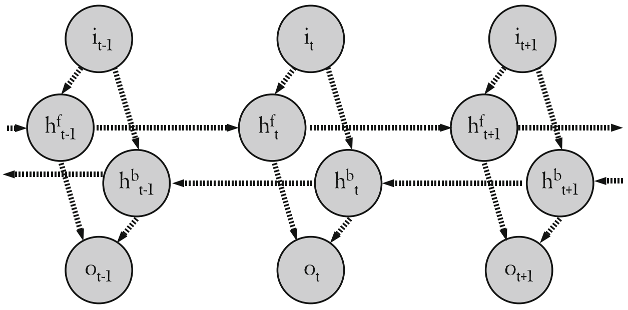


Fig. 2 Structure of a bidirectional network with input i, output o, and two hidden layers (h^f and h^b) for forward and backward processing

Combining bidirectional networks with LSTM gives bidirectional Long Short-Term Memory [21], which has demonstrated excellent performance in phoneme recognition [45], keyword spotting [28], handwriting recognition [46, 47], noise modeling [48], and emotion recognition from speech [49, 50].

Dynamic Bayesian Network Architecture for Keyword Detection

Dynamic Bayesian Networks can be interpreted as graphical models $G(V, E)$ that consist of a set of nodes V and edges E. Nodes represent random variables which can be either hidden or observed. Edges—or rather *missing* edges—encode conditional independence assumptions that are used to determine valid factorizations of the joint probability distribution. Conventional Hidden Markov Model approaches can be interpreted as *implicit* graph representations using a single Markov chain together with an integer state to represent all contextual and control information determining the allowable sequencing. In this work, however, we decided for the *explicit* approach [17], where information such as the current phoneme, the indication of a phoneme transition, or the position within a word is expressed by random variables. As shown in [17], explicit graph representations are advantageous whenever the set of hidden variables has factorization constraints or consist of multiple hierarchies. This section will introduce the explicit graph representation of our Tandem BLSTM-DBN keyword spotting system. Thereby, Sect. 4.1 will focus on the DBN used for decoding speech utterances and detecting keywords, respectively, while Sect. 4.2 outlines the graphical model structure we used during training.

Decoding

The Tandem BLSTM-DBN architecture for keyword spotting is depicted in Fig. 3. The network is composed of five different layers and hierarchy levels respectively: a word layer, a phoneme layer, a state layer, the observed features, and the BLSTM layer (nodes inside the gray shaded box). As can be seen in Fig. 3, the DBN jointly processes speech features and BLSTM phoneme predictions. The BLSTM layer consists of an input layer i_t, two hidden layers h_t^f and h_t^b (one for forward and one for backward processing), and an output layer o_t.

The following random variables are defined for every time step t:q_t denotes the phoneme identity, q_t^{ps} represents the position within the phoneme, q_t^{tr} indicates a phoneme transition, s_t is the current state with s_t^{tr} indicating a state

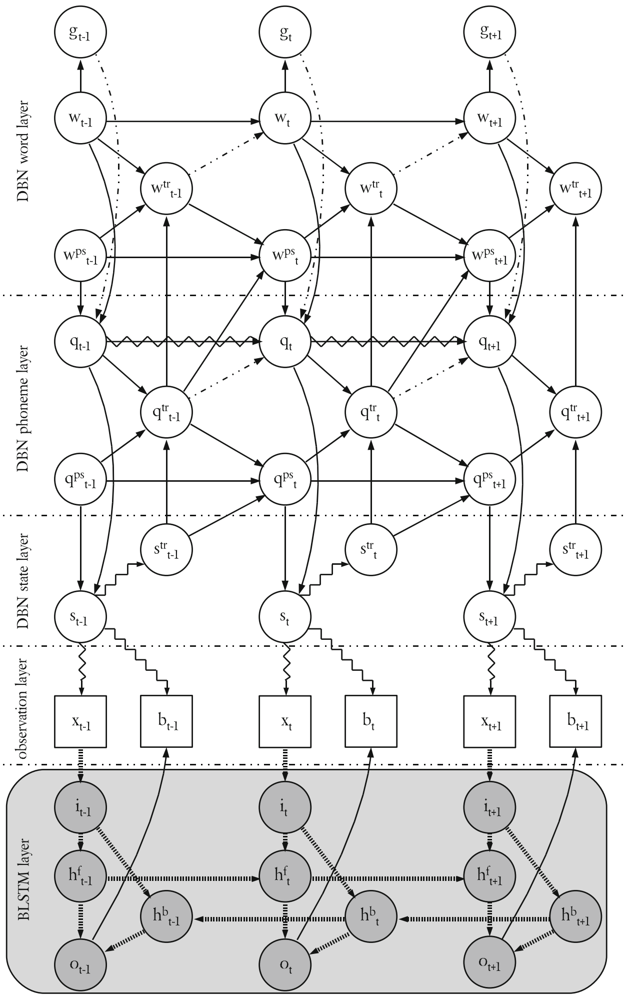


Fig. 3 Structure of the Tandem BLSTM-DBN keyword spotter: the BLSTM network (*grey shaded box*) provides a discrete phoneme prediction feature bt which is observed by the DBN, in addition to the MFCC features xt. The DBN is composed of a state, phoneme, and word layer, consisting of hidden transition (s_t^{tr},q_t^{tr},w_t^{tr},), position (q_t^{ps},w_t^{ps}), and identity (s^t, q_t, w_t) variables. Hidden variables (*circles*) and observed variables (*squares*) are connected via random CPFs (zig-zagged lines) or deterministic relations (*straight lines*). Switching parent dependencies are indicated with *dotted lines*

transition, and x_t denotes the observed acoustic features. The variables w_t, w_t^{ps} , and w_t^{tr} are identity, position, and transition variables for the word layer of the DBN whereas a hidden *garbage variable* g_t indicates whether the current word is a keyword or not. A second observed variable b_t contains the phoneme prediction of the BLSTM. Figure 3 displays hidden variables as circles and observed variables as squares. Deterministic relations are represented by straight lines, and zig-zagged lines correspond to random conditional probability functions (CPFs). Dotted lines refer to so-called *switching parents* [16], which allow a variable's parents to change conditioned on the current value of the switching parent. They can change not only the set of parents but also the implementation (i.e. the CPF) of a parent. The bold dashed lines in the BLSTM layer do not represent statistical relations but simple data streams.

Assuming a speech sequence of length T, the DBN structure specifies the factorization

$$\begin{aligned}
&p(g_{1:T}, w_{1:T}, w^{tr}_{1:T}, w^{ps}_{1:T}, q_{1:T}, q^{tr}_{1:T}, q^{ps}_{1:T}, s^{tr}_{1:T}, s_{1:T}, x_{1:T}, b_{1:T})\\
&\quad = \prod_{t=1}^{T} p(x_t|s_t)p(b_t|s_t)f(s_t|q_t^{ps}, q_t)p(s_t^{tr}|s_t)f(q_t^{tr}|q_t^{ps}, q_t, s_t^{tr})\\
&\quad f(w_t^{tr}|q_t^{tr}, w_t^{ps}, w_t)\\
&\quad f(g_t|w_t)f(q_1^{ps})p(q_1|w_1^{ps}, w_1, g_1)f(w_1^{ps})p(w_1)\\
&\quad \prod_{t=2}^{T} f(q_t^{ps}|s^{tr}_{t-1}, q^{ps}_{t-1}, q^{tr}_{t-1})\\
&p(w_t|w^{tr}_{t-1}, w_{t-1})p(q_t|q^{tr}_{t-1}, q_{t-1}, w_t^{ps}, w_t, g_t)\\
&\quad f(w_t^{ps}|q^{tr}_{t-1}, w^{ps}_{t-1}, w^{tr}_{t-1})
\end{aligned} \tag{1}$$

with $p(\cdot)$ denoting random conditional probability functions and $f(\cdot)$ describing deterministic relations. This factorization can be easily derived when inspecting the DBN layers of Fig. 3: in principle we have to build the product of all time steps and all variables while considering that variables might be conditioned on other (parent) variables. This corresponds to arrows in Fig. 3 that point to the corresponding (child) node. In case all parent nodes of a child node are located in the same time frame as the child node, we can build the product from $t = 1$ to $t = T$. Otherwise, if a variable is conditioned on variables from the previous time step, we build the product from $t = 2$ to $t = T$ and define initial CPFs for time step $t = 1$ that are not conditioned on variables from the previous time step (as for example $p(w_1)$). The factorization property in Eq. 1 can be exploited to optimally distribute the sums over the hidden variables into the products, using the junction tree algorithm [51]. Time and space complexity of the DBN is $\mathcal{O}(T \log T)$ and $\mathcal{O}(\log T)$, respectively [52].

The size of the BLSTM input layer i_t corresponds to the dimensionality of the acoustic feature vector x_t, whereas 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:

$$b_t = \max_{o_t}(o_{t,1}, \ldots, o_{t,j}, \ldots, o_{t,P}) \tag{2}$$

The CPFs $p(x_t|s_t)$ are described by Gaussian mixtures as common in an HMM system. Together with $p(b_t|s_t)$ and $p(s_t^{tr}|s_t)$, they are learned via EM training. Thereby s_t^{tr} is a binary variable, indicating whether a state transition takes place or not. The deterministic relations for s_t, q_t^{tr}, q_t^{ps}, w_t^{tr}, and w_t^{ps} are the same as in [20]. The hidden variable w_t can take values in the range $w_t = 0...K$ with K being the number of different keywords in the vocabulary. In case $w_t = 0$, the model is in the *garbage state*, which means that no keyword is uttered at that time. The variable g_t is then equal to one.

In our experiments, we simplified the word bigram $p(w_t|w^{tr}_{t-1}=1, w_{t-1})$ to a zerogram which makes each keyword equally likely. Yet we introduced differing a priori likelihoods for keywords and garbage phonemes:

$$p(w_t = 1:K|w^{tr}_{t-1}=1) = \frac{K \cdot 10^a}{K \cdot 10^a + 1} \tag{3}$$

and

$$p(w_t = 0|w^{tr}_{t-1}=1) = \frac{1}{K \cdot 10^a + 1}. \tag{4}$$

The parameter a can be used to adjust the trade-off between true positives and false positives. Setting $a = 0$ means that the a priori probability of a keyword and the probability that the current phoneme does not belong to a keyword are equal. Adjusting $a > 0$ implies a more aggressive search for keywords, leading to higher true positive and false positive rates.

As in [20], we assume that 'garbage words' always consist of only one phoneme. The variable q_t has two switching parents: q^{tr}_{t-1} and g_t. Similar to the word layer, q_t is equal to $q_{t\text{-}1}$ if $q^{tr}_{t-1}=0$. Otherwise, the switching parent g_t determines the parents of q_t. In case $g_t = 0$—meaning that the current word is a keyword—q_t is a deterministic function of the current keyword w_t and the position within the keyword w^{ps}_t. If the model is in the garbage state, q_t only depends on $q_{t\text{-}1}$ in a way that phoneme transitions between identical phonemes are forbidden.

Note that the design of the CPF $p(q_t|q^{tr}_{t-1}, q_{t-1}, w^{ps}_t, w_t, g_t)$ entails that the DBN will strongly tend to choose $g_t = 0$ (i.e. it will detect a keyword) once a phoneme sequence that corresponds to a keyword is observed. Decoding such an observation while being in the garbage state $g_t = 1$ would lead to 'phoneme transition penalties' since the CPF $p(q_t|q^{tr}_{t-1}=1, q_{t-1}, w^{ps}_t, w_t, g_t = 1)$ contains probabilities less than one. By contrast, $p(q_t|q^{tr}_{t-1} = 1, w^{ps}_t, w_t, g_t = 0)$ is deterministic, introducing no likelihood penalties at phoneme borders.

Training

The graphical model applied for learning the random CPFs $p(x_t|s_t)$, $p(s^{tr}_t|s_t)$, and $p(b_t|s_t)$ is depicted in Fig. 4. Compared to the GM used for keyword decoding (see Sect. 4.1), the GM for the training of the keyword spotter is less complex, since during (vocabulary independent) training, only phonemes are modeled. Thereby, the training procedure is split up into two stages: in the first stage phonemes are trained framewise, whereas during the second stage, the segmentation constraints are relaxed using a forced alignment (embedded training).

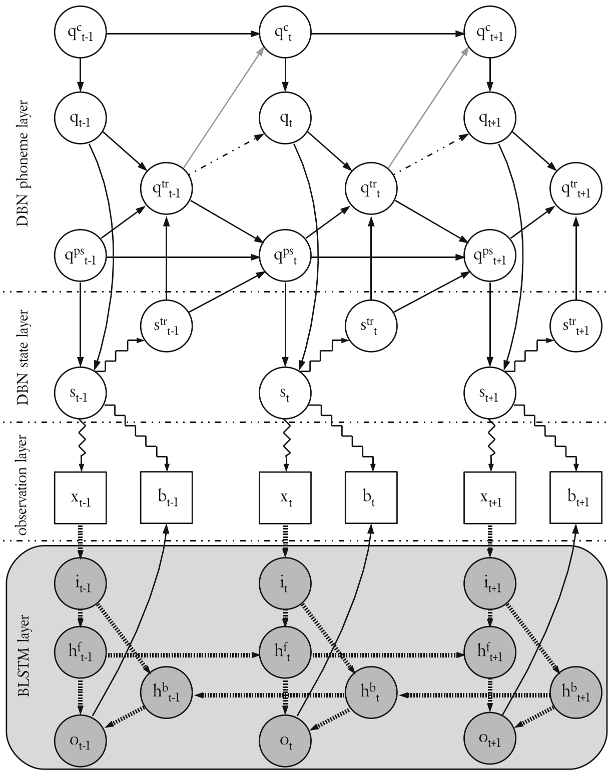


Fig. 4 DBN structure of the graphical model used to train the Tandem keyword spotter: a count variable q^c_t determines the current position in the phoneme sequence

The variable q^c_t shown in Fig. 4 is a count variable determining the current position in the phoneme sequence. Note that the gray-shaded arrow in Fig. 4 pointing from q^{tr}_{t-1} to q^c_t is only valid during the second training cycle when there are no segmentation constraints and will be ignored in Equation 5.

For a training sequence of length T, the DBN structure of Figure 4 specifies the factorization

$$\begin{aligned} & p(q^c_{1:T}, q_{1:T}, q^{tr}_{1:T}, q^{ps}_{1:T}, s^{tr}_{1:T}, s_{1:T}, x_{1:T}, b_{1:T}) \\ & = \prod_{t=1}^{T} p(x_t|s_t)p(b_t|s_t)f(s_t|q^{ps}_t, q_t)p(s^{tr}_t|s_t)f(q^{tr}_t|q^{ps}_t, q_t, s^{tr}_t) \\ & f(q_t|q^c_t)f(q^{ps}_1)f(q^c_1) \\ & \prod_{t=2}^{T} f(q^{ps}_t|s^{tr}_{t-1}, q^{ps}_{t-1}, q^{tr}_{t-1})f(q^c_t|q^c_{t-1}). \end{aligned} \tag{5}$$

During training, the current phoneme q_t is known, given the position q^c_t in the training utterance, which implies a deterministic mapping $f(q_t|q^c_t)$. In the first training cycle, q^c_t is incremented in every time frame, whereas in the second cycle q^c_t is only incremented if $q^{tr}_{t-1} = 1$.

For the training of the DBN we use GMTK [53] which in turn uses Expectation Maximization (EM) [54] and generalized EM (GEM) [55] training, depending on the

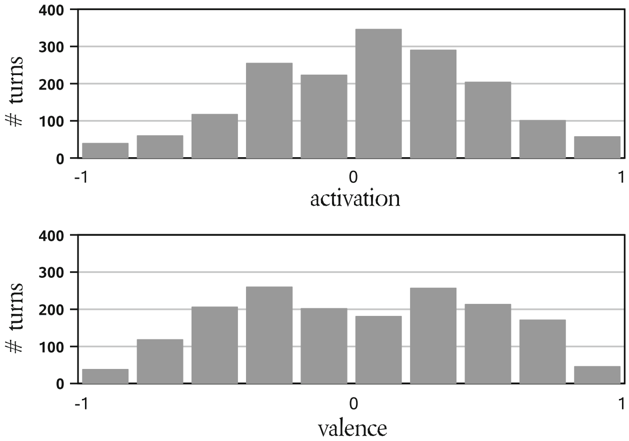


Fig. 5 Histogram of the turn annotations for activation and valence in the SAL corpus

parameter sharing currently in use [53]. A detailed description of both strategies can be found in [56].

The BLSTM network is trained independently with standard backpropagation through time (BPTT) [57] using the exact error gradient as in [21]. All necessary BPTT equations for LSTM training are detailed in [58].

We used a learning rate of 10^{-5} and a momentum of 0.9. To improve generalization, zero mean Gaussian noise with standard deviation 0.6 is added to the inputs during training. Prior to training, all weights of the BLSTM network are randomly initialized in the range from $-$ 0.1 to 0.1.

Experiments

To evaluate our keyword detection system, we used the TIMIT corpus [29] as well as the SAL database [30], containing spontaneous, emotionally colored speech. The SAL corpus is a sub-set of the HUMAINE database[2] and was recorded during natural human-computer conversations. A Wizard-of-Oz SAL interface imitating the functionality of the SEMAINE system (see Sect. 2) was used for emotion induction. The users had to speak to the four different virtual characters introduced in Sect. 2, whereas each character represents one affective state (matter-of-fact, happiness, sadness, or anger) and tried to induce the corresponding emotion in the user. Thus, each utterance spoken by the user can be characterized by its degree of activation and valence, using a scale from -1 (weak and negative, respectively) to 1 (strong and positive, respectively). Histograms showing the distribution of these 'emotional dimensions' over the SAL corpus can be seen in Fig. 5. Thereby, the values for activation and valence correspond to the turn annotations averaged over four different labelers.

Both, the database and the recording procedures are described in more detail in [30]. Training and test sets were split according to [49].

The acoustic feature vectors consisted of cepstral mean-normalized MFCC coefficients 1–12, energy, as well as first and second order delta coefficients. For the training of the BLSTM, 200 utterances of the TIMIT training split were used as validation set while the net was trained on the remaining training sequences. The BLSTM input layer had a size of 39 (one for each MFCC feature) and the size of the output layer was also 39 since we used the reduced set of 39 TIMIT phonemes. Both hidden LSTM layers contained 100 memory blocks of one cell each.

During the first training cycle of the DBN, phonemes were trained framewisely using the training portion of the TIMIT corpus. Thereby, all Gaussian mixtures were split once the change of the overall log likelihood of the training set became less than 0.02 %. The number of mixtures per state was increased to 16. In the second training cycle segmentation constraints were relaxed, whereas no further mixture splitting was conducted. Phoneme models were composed of three hidden states each. Prior to evaluation on the SAL corpus, all means, variances, and weights of the Gaussian mixture probability distributions $p(x_t|s_t)$, as well as the state transition probabilities $p(s_t^{tr}|s_t)$ were re-estimated using the training split of the SAL corpus. Again, re-estimation was stopped once the change of the overall log likelihood of the SAL training set fell below a threshold of 0.02 %.

For comparison, the performance of a phoneme-based keyword spotter using conventional HMM modeling was evaluated. Analogous to the DBN, each phoneme was represented by three states (left-to-right HMMs) with 16 Gaussian mixtures. Thereby, we used cross-word triphone models in order to account for contextual information. The HMMs were trained using HTK [59]. Thereby the initial monophone models consisted of one Gaussian mixture per state. All initial means and variances were set to the global means and variances of all feature vector components (flat start initialization). The monophone models were then trained using four iterations of embedded Baum-Welch re-estimation [60]. After that, the monophones were mapped to tied-state cross-word triphone models with shared state transition probabilities. Two Baum-Welch iterations were performed for re-estimation of the triphone models. Finally, the number of mixture components of the triphone models was increased to 16 in four successive rounds of mixture doubling and re-estimation (four iterations in every round). In each round, the newly created mixture components are copied from the existing ones, mixture weights are divided by two, and the means are shifted by plus and minus 0.2 times the standard deviation.

[2] http://www.emotion-research.net/download/pilot-db/.

For HMM-based keyword detection, we defined a set of keyword models and a garbage model. The keyword models estimate the likelihood of a feature vector sequence, given that it corresponds to the keyword phoneme sequence. The garbage model is composed of phoneme HMMs that are fully connected to each others, meaning that it can model any phoneme sequence. Via Viterbi decoding the best path through all models is found, and a keyword is detected as soon as the path passes through the corresponding keyword HMM. In order to be able to adjust the operating point on the Receiver Operating Characteristic (ROC) curve, we introduced different a priori likelihoods for keyword and garbage HMMs, identical to the word zerogram used for the DBN. Apart from the transition probabilities implied by the zerogram, the HMM system uses no additional likelihood penalties at the phoneme borders.

As a second baseline model, we evaluated the keyword spotting performance of a hybrid BLSTM-HMM system, since this approach was shown to prevail over the standard HMM approach [61]. Unlike the proposed Tandem model, the hybrid approach exclusively uses BLSTM phoneme predictions for keyword detection. Thus, it does not use Gaussian mixture modeling since the MFCC features are not observed by the HMM but only by the BLSTM network. The BLSTM network of the hybrid model is furthermore equipped with a Connectionist Temporal Classification (CTC) output layer [62] which allows the network to be trained on unsegmented data. Typical phoneme prediction errors made by the CTC network are modeled by the HMM layer of the hybrid system (similar to the trained CPFs $p(b_t|s_t)$ for the Tandem model). For further details on the hybrid approach, the reader is referred to [61].

In order to evaluate our keyword spotting system on the TIMIT corpus, we randomly chose 60 keywords (as in [20]). The used dictionary allowed for multiple pronunciations. The trade-off parameter a (see Eq. 3) was varied between zero and seven.

Figure 6 c ROC curves displaying the true positive rate (tpr) as a function of the false positive rate (fpr) for the baseline HMM and the hybrid BLSTM-HMM, as well as for the DBN with and without an additional BLSTM layer. Note that due to the design of the recognizer, the full ROC curve—ending at an operating point tpr = 1 and fpr = 1—cannot be determined, since the model does not include a confidence threshold that can be set to an arbitrarily low value. The most significant performance gain of context modeling via BLSTM predictions occurs at an operating point with a false positive rate of 0.1 %: there, the Tandem approach can increase the true positive rate by 13.5 %, when compared to the DBN without BLSTM layer. Conducting the McNemar's test revealed that the performance difference between the BLSTM-DBN and the DBN is statistically significant at a common significance level of 0.01 (for details about the McNemar's test, see [63]). For higher values of the trade-off parameter a, implying a more aggressive search for keywords, the performance gap between the DBN and the Tandem keyword spotter becomes smaller, as more phoneme confusions are tolerated when seeking for keywords. Furthermore, both DBN architectures significantly outperform the baseline HMM approach. At low false positive rates, the hybrid BLSTM-HMM prevails over the DBN approach, however; as soon as more false positives are tolerated, the performance of the hybrid model approaches the baseline HMM performance and is inferior to the DBN.

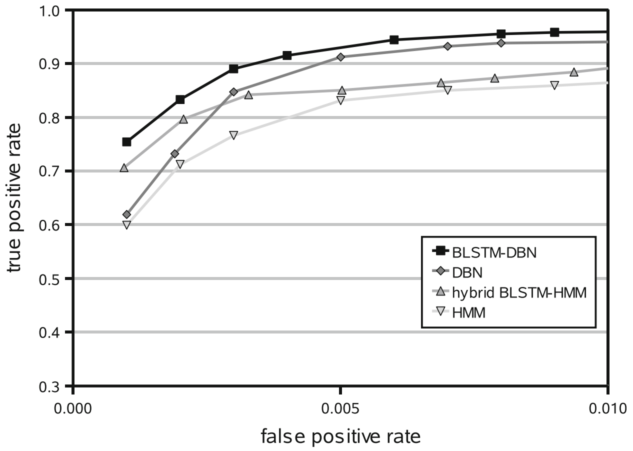


Fig. 6 Evaluation on the TIMIT corpus (60 keywords): Part of the ROC curve for the baseline HMM system, the hybrid BLSTM-HMM, the DBN keyword spotter (without BLSTM phoneme predictions) and the Tandem BLSTM-DBN approach. The operating points correspond to $a = 0, 1, 2, 3$, etc

As mentioned earlier, our keyword spotting techniques are vocabulary independent, meaning that new keywords can be added without having to re-train the system. In order to illustrate that adding new keywords does not downgrade recognition performance, we added 20 randomly selected keywords to the vocabulary (so that we had a total number of 80 keywords) and repeated all experiments. Note that we used the same BLSTM network and the same CPFs $p(x_t|s_t)$, $p(s_t^{tr}|s_t)$, and $p(b_t|s_t)$ as for the original experiments with 60 keywords. As can be seen in Fig. 7, the changes in ROC performance are marginal. The DBN performance at low false positive rates is even slightly better than in the previous experiment. Still, the Tandem BLSTM-DBN significantly outperforms all other investigated approaches.

For performance evaluation on the SAL corpus, we randomly selected 24 keywords (the same as in [7]). The resulting ROC performance can be seen in Fig. 8. Obviously the task of keyword detection in emotional speech is considerably harder, implying lower true positive rates and higher false positive rates, respectively. As for the TIMIT experiment, our Tandem BLSTM-DBN approach

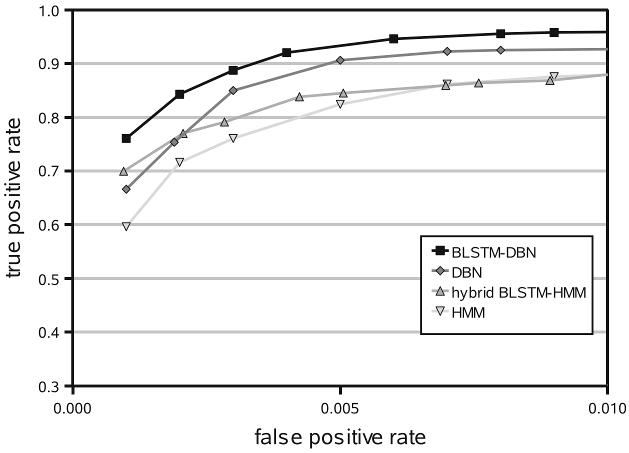


Fig. 7 Evaluation on the TIMIT corpus (80 keywords): Part of the ROC curve for the baseline HMM system, the hybrid BLSTM-HMM, the DBN keyword spotter (without BLSTM phoneme predictions) and the Tandem BLSTM-DBN approach. The operating points correspond to $a = 0, 1, 2, 3$, etc

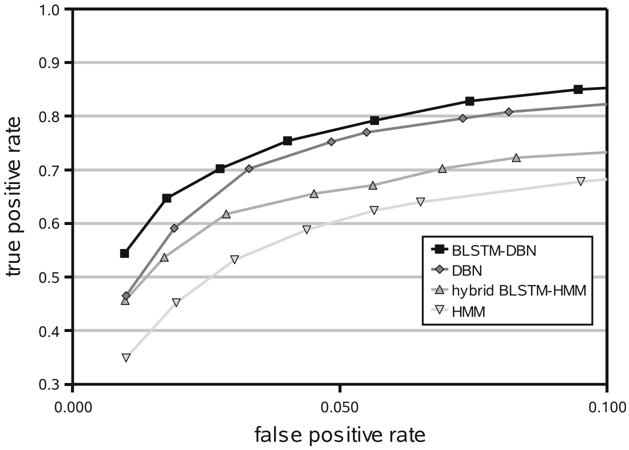


Fig. 8 Evaluation on the SAL corpus (24 keywords): Part of the ROC curve for the baseline HMM system, the hybrid BLSTM-HMM, the DBN keyword spotter (without BLSTM phoneme predictions) and the Tandem BLSTM-DBN approach. The operating points correspond to $a = 0, 1, 2, 3$, etc

prevails over the DBN, the hybrid BLSTM-HMM, and the HMM baseline system with a performance gain of up to 8 % when compared to the DBN.

Conclusion

In this article, we proposed a novel vocabulary independent Dynamic Bayesian Network architecture for robustly detecting keywords in continuous speech. Our keyword spotting system is tailored for usage within cognitive virtual agents such as the SEMAINE system for Sensitive Artificial Listening which demand for robustness, e. g. with respect to emotional coloring of speech.

Apart from conventional MFCC features, our keyword spotting system also takes into account the phoneme predictions of a bidirectional Long Short-Term Memory RNN. Thus, it can model a self-learned amount of contextual information in order to improve the discrimination between keywords and arbitrary speech within the DBN. Since our concept is based on a Tandem phoneme recognizer and does not consider specific keywords during the training phase, new keywords can be added without having to retrain the network. A further advantage of our approach is that it does not require the training of an explicit garbage model.

We showed that incorporating BLSTM phoneme predictions into our DBN architecture can enhance keyword detection performance on the TIMIT corpus, but also on the SAL corpus which contains spontaneous emotional speech as it is to be expected in an emotionally sensitive virtual agent scenario.

By using the same BLSTM recurrent neural network architecture as successfully applied for emotion recognition within the SEMAINE framework [22], our Tandem BLSTM-DBN keyword spotter offers the possibility to create a unified system for jointly modeling phonemes and emotion using a multi-task learning strategy.

Future possibilities also include the investigation of alternative BLSTM network topologies and the combination of triphone and BLSTM modeling. A further interesting approach towards better recognition performance through combined BLSTM and DBN modeling would be to jointly decode speech with LSTM networks and DBNs by using techniques for data fusion of potentially asynchronous sequences such as multidimensional dynamic time warping [64] or asynchronous Hidden Markov Models [65].

Acknowledgements 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).

References

1. Taylor JG (2009) Cognitive computation. Cognit Comput. 1(1): 4–16
2. Vo MT, Waibel A (1993) Multimodal human-computer interaction. In: Proceedings of ISSD. Waseda, pp 95–101
3. Oviatt S (2000) Multimodal interface research: A science without borders. In: Proceedings of ICSLP. pp 1–6
4. 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
5. Rose RC (1995) Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition. Comput Speech Lang 9(4):309–333
6. Keshet J, Grangier D, Bengio S (2007) Discriminative Keyword Spotting. In: Proceedings of NOLISP. Paris. pp 47–50

7. 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
8. Liu H, Lieberman H, Selker T (2003) A model of textual affect sensing using real-world knowledge. In: Proceedings of the 8th international conference on intelligent user interfaces. Miami, Florida. pp 125–132
9. Ma C, Prendinger H, Ishizuka M (2005) A Chat system based on emotion estimation from text and embodied conversational messengers. In: Entertainment Computing. vol. 3711/2005. Springer. pp 535–538
10. Ziemke T, Lowe R (2009) On the role of emotion in embodied cognitive architectures: from organisms to robots. Cognit Comput 1(1):104–117
11. Rose RC, Paul DB (1990) A hidden markov model based keyword recognition system. In: Proceedings of ICASSP. Albuquerque. p. 129–132
12. Ketabdar H, Vepa J, Bengio S, Boulard H (2006) Posterior based keyword spotting with a priori thresholds. In: IDAIP-RR. pp 1–8
13. Benayed Y, Fohr D, Haton JP, Chollet G (2003) Confidence measure for keyword spotting using support vector machines. In: Proceedings of ICASSP. pp 588–591
14. 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
15. Weintraub M (1993) Keyword-spotting using SRI's DECIPHER large vocabulary speech recognition system. In: Proceedings of ICASSP. Minneapolis. pp 463–466
16. Bilmes JA (2003) Graphical models and automatic speech recognition. In: Rosenfeld R, Ostendorf M, Khudanpur S, Johnson M (eds). Mathematical foundations of speech and language processing. New York: Springer. pp 191–246
17. Bilmes JA, Bartels C (2005) Graphical model architectures for speech recognition. IEEE Signal Process Mag 22(5):89–100
18. Lin H, Stupakov A, Bilmes JA (2009) Improving multi-lattice alignment based spoken keyword spotting. In: Proceedings of ICASSP. Taipei. pp 4877–4880
19. Lin H, Bilmes JA, Vergyri D, Kirchhoff K (2007) OOV detection by joint word/phone lattice alignment. In: Proceedings of ASRU. Kyoto. pp 478–483
20. Wöllmer M, Eyben F, Schuller B, Rigoll G (2009) Robust vocabulary independent keyword spotting with graphical models. In: Proceedings of ASRU. Merano. pp 349–353
21. Graves A, Fernandez S, Schmidhuber J (2005) Bidirectional LSTM networks for improved phoneme classification and recognition. In: Proceedings of ICANN. Warsaw. pp 602–610
22. Eyben F, Wöllmer M, Graves A, Schuller B, Douglas-Cowie E, Cowie R (2009) On-line emotion recognition in a 3-D activation-valence-time continuum using acoustic and linguistic cues. J Multimodal User Interfaces (JMUI), Special Issue on Real-time Affect Analysis and Interpretation: Closing the Loop in Virtual Agents 3:7–19
23. Hermansky H, Ellis DPW, Sharma S (2000) Tandem connectionist feature extraction for conventional HMM systems. In: Proceedings of ICASSP. Istanbul. pp 1635–1638
24. Ketabdar H, Bourlard H (2008) Enhanced phone posteriors for improving speech recognition systems. In: IDIAP-RR. 39. pp 1–23
25. Ellis DPW, Singh R, Sivadas S (2001) Tandem acoustic modeling in large-vocabulary recognition. In: Proceedings of ICASSP. Salt Lake City. pp 517–520
26. Boulard H, Morgan N (1994) Connectionist speech recognition: a hybrid approach. Kluwer Academic Publishers, Dordrecht
27. Bengio Y (1999) Markovian models for sequential data. Neural Comput Surv 2:129–162
28. 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
29. Garofolo JS, Lamel LF, Fisher WM, Fiscus JG, Pallett DS, Dahlgren NL (1993) DARPA TIMIT acoustic phonetic continuous speech corpus CDROM. NIST
30. Douglas-Cowie E, Cowie R, Sneddon I, Cox C, Lowry O, McRorie M, et al. (2007) The HUMAINE Database: addressing the collection and annotation of naturalistic and induced emotional data. In: Affective computing and intelligent interaction. vol. 4738/2007. Springer. pp. 488–500
31. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
32. Yang HH, Sharma S, van Vuuren S, Hermansky H (2000) Relevance of time-frequency features for phonetic and speaker/channel classification. Speech Commun. 31:35–50
33. Bilmes JA (1998) Maximum mutual information based reduction strategies for cross-correlation based joint distributional modeling. In: Proceedings of ICASSP. pp 469–472
34. Schuller B, Müller R, Eyben F, Gast J, Hörnler B, Wöllmer M, et al. (2009) Being bored? recognising natural interest by extensive audiovisual integration for real-life application. Image Vis Comput J (IMAVIS), Special Issue on Visual and Multimodal Analysis of Human Spontaneous Behavior 27(12):1760–1774
35. Schuller B, Rigoll G (2009) Recognising interest in conversational speech—comparing bag of frames and supra-segmental features. In: Proceedings of interspeech. Brighton. pp 1999–2002
36. Quattoni A, Wang S, Morency LP, Collins M, Darrell T (2007) hidden conditional random fields. IEEE Trans Pattern Anal Mach Intell 29:1848–1853
37. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J (2001) 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. IEEE Press, . pp 1–15
38. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
39. Schaefer AM, Udluft S, Zimmermann HG (2008) Learning long-term dependencies with recurrent neural networks. Neurocomputing 71(13-15):2481–2488
40. 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
41. Lang KJ, Waibel AH, Hinton GE (1990) A time-delay neural network architecture for isolated word recognition. Neural Netw 3(1):23–43
42. Schmidhuber J (1992) Learning complex extended sequences using the principle of history compression. Neural Comput 4(2):234–242
43. Jaeger H (2001) The echo state approach to analyzing and training recurrent neural networks. Bremen: German national research center for information technology. (Tech. Rep. No. 148)
44. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681
45. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5-6):602–610
46. Graves A, Fernandez S, Liwicki M, Bunke H, Schmidhuber J (2008) Unconstrained online handwriting recognition with recurrent neural networks. Adv Neural Inf Process Syst. 20:1–8
47. Liwicki M, Graves A, Fernandez S, Bunke H, Schmidhuber J (2007) A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In: Proceedings of ICDAR. Curitiba. pp 367–371

48. Wöllmer M, Eyben F, Schuller B, Sun Y, Moosmayr T, Nguyen-Thien N (2009) Robust in-car spelling recognition—a tandem BLSTM-HMM approach. In: Proceedings of interspeech. Brighton. p. 2507–2510
49. Wöllmer M, Eyben F, Reiter S, Schuller B, Cox C, Douglas-Cowie E, et al. (2008) Abandoning emotion classes—towards continuous emotion recognition with modelling of long-range dependencies. In: Proceedings of interspeech. Brisbane. p. 597–600
50. Wöllmer M, Eyben F, Schuller B, Douglas-Cowie E, Cowie R. Data-driven clustering in emotional space for affect recognition using discriminatively trained LSTM networks. In: Proceedings of interspeech. Brighton. pp 1595–1598 (2009)
51. Jensen FV (1996) An introduction to Bayesian networks. Springer, Brelin
52. Zweig G, Padmanabhan M (2000) Exact alpha-beta computation in logarithmic space with application to map word graph construction. In: Proceedings of ICSLP. Beijing. pp 855–858
53. Bilmes J, Zweig G (2002) The graphical models toolkit: an open source software system for speech and time-series processing. In: Proceedings of ICASSP. pp 3916–3919
54. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B. 39:185–197
55. Bilmes J (2008) Gaussian models in automatic speech recognition. In: Signal processing in acoustics. Springer, New York. pp 521–555
56. Bilmes J (1997) A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models. University of Berkeley. Technical Report ICSI-TR-97-02
57. Williams RJ, Zipser D (1995) Gradient-based learning algorithms for recurrent neural networks and their computational complexity. In: Chauvin Y, Rumelhart DE, (eds) Back-propagation: theory, architectures and applications. Lawrence Erlbaum Publishers, Hillsdale, pp 433–486
58. Graves A (2008) Supervised sequence labelling with recurrent neural networks. Technische Universität München, Germany
59. Young S, Evermann G, Gales M, Hain T, Kershaw D, Liu X et al. (2006) The HTK book (v3.4). Cambridge University Press, Cambridge
60. Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41(1):164–171
61. Wöllmer M, Eyben F, Schuller B, Rigoll G (2010) Spoken term detection with connectionist temporal classification—a novel hybrid CTC-DBN approach. In: Proceedings of ICASSP. Dallas. pp. 5274–5277
62. Graves A, Fernandez S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: Labelling unsegmented data with recurrent neural networks. In: Proceedings of ICML. Pittsburgh. p. 369–376
63. Gillick L, Cox SJ (1989) Some statistical issues in the comparison of speech recognition algorithms. In: Proceedings of ICASSP. Glasgow. pp 23–26
64. 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 73:366–380
65. Bengio S (2003) An asynchronous Hidden Markov model for audio-visual speech recognition. Advances in NIPS 15. pp 1–8