Improving Keyword Spotting with a Tandem BLSTM-DBN Architecture

Martin Wöllmer[1], Florian Eyben[1], Alex Graves[2], Björn Schuller[1], and Gerhard Rigoll[1]

[1] Institute for Human-Machine Communication, Technische Universität München, Germany
{woellmer,eyben,schuller,rigoll}@tum.de
[2] Institute for Computer Science VI, Technische Universität München, Germany

Abstract. We propose a novel architecture for keyword spotting which is composed of a Dynamic Bayesian Network (DBN) and a bidirectional Long Short-Term Memory (BLSTM) recurrent neural net. The DBN uses a hidden garbage variable as well as the concept of switching parents to discriminate between keywords and arbitrary speech. Contextual information is incorporated by a BLSTM network, providing a discrete phoneme prediction feature for the DBN. Together with continuous acoustic features, the discrete BLSTM output is processed by the DBN which detects keywords. Due to the flexible design of our Tandem BLSTM-DBN recognizer, new keywords can be added to the vocabulary without having to re-train the model. Further, our concept does not require the training of an explicit garbage model. Experiments on the TIMIT corpus show that incorporating a BLSTM network into the DBN architecture can increase true positive rates by up to 10 %.

Keywords: Keyword Spotting, Long Short-Term Memory, Dynamic Bayesian Networks.

1 Introduction

Keyword spotting aims at detecting one or more predefined keywords in a given speech utterance. In recent years keyword spotting has found many applications, e.g. in voice command detectors, information retrieval systems, or embodied conversational agents. Hidden Markov Model (HMM) based keyword spotting systems [9] usually require keyword HMMs and a *garbage* HMM to model both, keywords and non-keyword parts of the speech sequence. However, the design of the garbage HMM is a non-trivial task. Using whole word models for keyword and garbage HMMs presumes that there are enough occurrences of the keywords in the training corpus and suffers from low flexibility since new keywords cannot be added to the system without having to re-train it. Modeling phonemes instead of whole words offers the possibility to design a garbage HMM that connects all phoneme models but implies that the garbage HMM can potentially model any phoneme sequence, including the keyword itself.

J. Solé-Casals and V. Zaiats (Eds.): NOLISP 2009, LNAI 5933, pp. 68–75, 2010.

In this paper we present a new Dynamic Bayesian Network (DBN) design which can be used for robust keyword spotting and overcomes most of the drawbacks of other approaches. Dynamic Bayesian Networks offer a flexible statistical framework that is increasingly applied for speech recognition tasks [2,1] since it allows for conceptual deviations from the conventional HMM architecture. Our keyword spotter does not need a trained garbage model and is robust with respect to phoneme recognition errors. Unlike large vocabulary speech recognition systems, our technique does not require a language model but only the keyword phonemizations. Thereby we use a hidden garbage variable and the concept of *switching parents* [1] to model either a keyword or arbitrary speech.

In order to integrate contextual information into the keyword spotter, we extend our DBN architecture to a Tandem recognizer that uses the phoneme predictions of a bidirectional Long Short-Term Memory (BLSTM) recurrent neural net together with conventional MFCC features. Tandem architectures which combine the output of a discriminatively trained neural net with dynamic classifiers such as HMMs have been successfully used for speech recognition tasks and are getting more and more popular [6,8]. BLSTM networks efficiently exploit past and future context and have been proven to outperform standard methods of modeling contextual information such as triphone HMMs [4]. As shown in [12], the framewise phoneme predictions of a BLSTM network can enhance the performance of a discriminative keyword spotter. In [3] a BLSTM based keyword spotter trained on a fixed set of keywords is introduced. However, this approach requires re-training of the net as soon as new keywords are added to the vocabulary, and gets increasingly complex if the keyword vocabulary grows. The keyword spotting architecture proposed herein can be seen as an extension of the graphical model for spoken term detection we introduced in [13]. Thus, we aim at combining the flexibility of our DBN architecture with the ability of a BLSTM network to capture long-range time dependencies and the advantages of Tandem speech modeling.

The structure of this paper is as follows: Section 2 reviews the principle of DBNs and BLSTMs as the two main components of our keyword spotter. Section 3 explains the architecture of our Tandem recognizer while experimental results are presented in Section 4. Concluding remarks are mentioned in Section 5.

2 Keyword Spotter Components

Our Tandem keyword spotter architecture consists of two major components: a Dynamic Bayesian Network processing observed speech feature vectors to discriminate between keywords and non-keyword speech, and a BLSTM network which takes in to account contextual information to provide an additional discrete feature for the DBN. The following sections will shortly review the basic principle of DBNs and BLSTMs.

2.1 Dynamic Bayesian Network

Dynamic Bayesian Networks can be interpreted as graphical models $G(V, E)$ which 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 [2], where information such as the current phoneme, the indication of a phoneme transition, or the position within a word is expressed by random variables.

2.2 Bidirectional LSTM Network

The basic idea of bidirectional recurrent neural networks [10] is to use two recurrent network layers, one that processes the training sequence forwards and one that processes it backwards. Both networks are connected to the same output layer, which therefore has access to complete information about the data points before and after the current point in the sequence. The amount of context information that the network actually uses is learned during training, and does not have to be specified beforehand.

Analysis of the error flow in conventional recurrent neural nets (RNNs) resulted in the finding that long time lags are inaccessible to existing RNNs since the backpropagated error either blows up or decays over time (vanishing gradient problem). This led to the introduction of Long Short Term Memory (LSTM) RNNs [7]. An LSTM layer is composed of recurrently connected memory blocks, each of which contains one or more recurrently connected 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. Their effect is to allow the network to store and retrieve information over long periods of time.

Combining bidirectional networks with LSTM gives bidirectional LSTM, which has demonstrated excellent performance in phoneme recognition [4], keyword spotting [3], and emotion recognition [11]. A detailed explanation of BLSTM networks can be found in [5].

3 Architecture

The Tandem BLSTM-DBN architecture we used for keyword spotting is depicted in Figure 1. 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 grey shaded box). For the sake of simplicity only a simple LSTM layer, consisting of inputs i_t, a hidden layer h_t, and outputs o_t, is shown in Figure 1, instead of the more complex bidirectional LSTM which would contain two RNNs.

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 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 1 displays hidden variables as circles and observed variables as squares. Deterministic conditional probability functions (CPFs) are represented by straight lines and zig-zagged lines correspond to random CPFs. Dotted lines refer to so-called *switching parents* [1], which allow a variable's parents to change conditioned on the current value of the switching parent. Thereby a switching parent can not only change the set of parents but also the implementation (i.e. the CPF) of a parent. The bold dashed lines in the LSTM layer do not represent statistical relations but simple data streams.

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

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

with $p(\cdot)$ denoting random conditional probability functions and $f(\cdot)$ describing deterministic CPFs.

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:

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

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 learnt via EM training. Thereby s_t^{tr} is a binary variable, indicating whether a state transition takes place or not. Since the current state is known with certainty, given the phoneme and the phoneme position, $f(s_t|q_t^{ps}, q_t)$ is purely deterministic. A phoneme transition occurs whenever $s_t^{tr} = 1$ and $q_t^{ps} = S$ provided that S denotes the number of

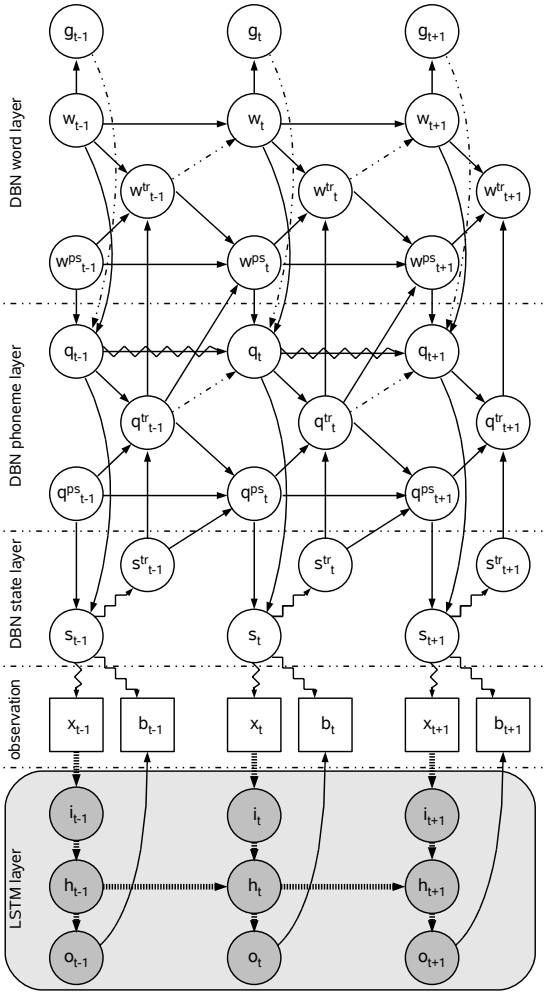

Fig. 1. Structure of the Tandem BLSTM-DBN keyword spotter

states of a phoneme. This is expressed by the function $f(q_t^{tr}|q_t^{ps}, q_t, s_t^{tr})$. The phoneme position q_t^{ps} is known with certainty if s_{t-1}^{tr}, q_{t-1}^{ps}, and q_{t-1}^{tr} are given.

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. w_{t-1}^{tr} is a switching parent of w_t: if no word transition is indicated, w_t is equal to w_{t-1}. Otherwise a word bigram specifies the CPF $p(w_t|w_{t-1}^{tr} = 1, w_{t-1})$. In our experiments we simplified the word bigram to a zerogram which makes each keyword equally likely. Yet, we introduced differing a priori likelihoods for keywords and garbage phonemes:

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

and

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

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. The CPFs $f(w_t^{tr}|q_t^{tr}, w_t^{ps}, w_t)$ and $f(w_t^{ps}|q_{t-1}^{tr}, w_{t-1}^{ps}, w_{t-1}^{tr})$ are similar to the phoneme layer of the DBN (i.e. the CPFs for q_t^{tr} and q_t^{ps}). However, we assume that "garbage words" always consist of only one phoneme, meaning that if $g_t = 1$, a word transition occurs as soon as $q_t^{tr} = 1$. Consequently w_t^{ps} is always zero if the model is in the garbage state. The variable q_t has two switching parents: q_{t-1}^{tr} and g_t. Similar to the word layer, q_t is equal to q_{t-1} if $q_{t-1}^{tr} = 0$. 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_t^{ps}. If the model is in the garbage state, q_t only depends on q_{t-1} in a way that phoneme transitions between identical phonemes are forbidden.

Note that the design of the CPF $p(q_t|q_{t-1}^{tr}, q_{t-1}, w_t^{ps}, w_t, g_t)$ 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 $p(q_t|q_{t-1}^{tr} = 1, q_{t-1}, w_t^{ps}, w_t, g_t = 1)$ contains probabilities less than one. By contrast, $p(q_t|q_{t-1}^{tr} = 1, w_t^{ps}, w_t, g_t = 0)$ is deterministic, introducing no likelihood penalties at phoneme borders.

4 Experiments

Our keyword spotter was trained and evaluated on the TIMIT corpus. The feature vectors consisted of cepstral mean normalized MFCC coefficients 1 to 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. To improve generalization, zero mean Gaussian noise with standard deviation 0.6 was added to the inputs during training. We used a learning rate of 10^{-5} and a momentum of 0.9.

The independently trained BLSTM network was then incorporated into the DBN in order to allow the training of the CPFs $p(b_t|s_t)$. 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 0.02%

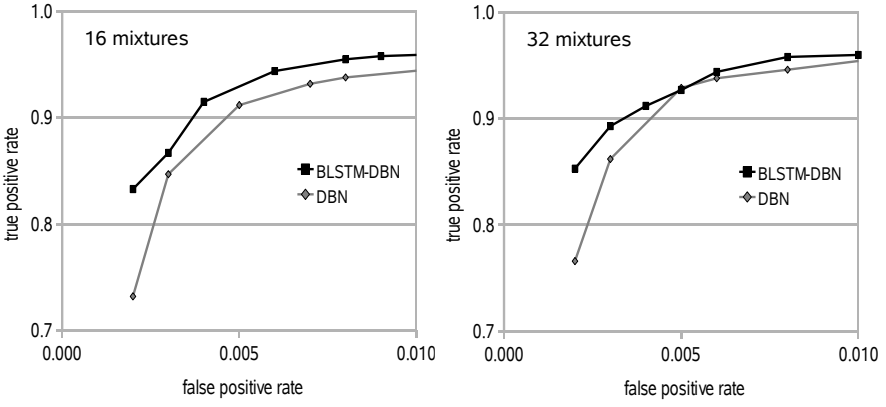

Fig. 2. Part of the ROC curve for the DBN keyword spotter and the Tandem BLSTM-DBN approach using different values for the trade-off parameter a. Left side: 16 Gaussian mixtures; right side 32 Gaussian mixtures.

convergence was reached until the number of mixtures per state increased to 16 and 32 respectively. 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.

We randomly chose 60 keywords from the TIMIT corpus to evaluate the keyword spotter. The used dictionary allowed for multiple pronunciations. The trade-off parameter a (see Equation 3) was varied between 0 and 10.

Figure 2 shows a part of the Receiver Operating Characteristics (ROC) curves for the DBN and the Tandem BLSTM-DBN keyword spotter, displaying the true positive rate (tpr) as a function of the false positive rate (fpr). 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 0.2%: there, the Tandem approach can increase the true positive rate by 10%. 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 higher values of the trade-off parameter a, implying a more aggressive search for keywords, the performance gap becomes smaller as more phoneme confusions are tolerated when seeking for keywords.

5 Conclusion

This paper introduced a Tandem BLSTM-DBN keyword spotter that makes use of the phoneme predictions generated by a bidirectional Long Short-Term Memory recurrent neural net. We showed that the incorporation of contextual

information via BLSTM networks leads to significantly improved keyword spotting results.

Future works might include a combination of triphone and BLSTM modeling as well as processing the entire vector of BLSTM output activations instead of exclusively using the most likely phoneme prediction.

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. Bilmes, J. A.: Graphical models and automatic speech recognition. In: Mathematical Foundations of Speech and Language Processing (2003)
2. Bilmes, J.A., Bartels, C.: Graphical model architectures for speech recognition. IEEE Signal Processing Magazine 22(5), 89–100 (2005)
3. Fernández, S., Graves, A., Schmidhuber, J.: An Application of Recurrent Neural Networks to Discriminative Keyword Spotting. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 220–229. Springer, Heidelberg (2007)
4. Graves, A., Fernandez, S., Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition. In: Proc. of ICANN, Warsaw, Poland, vol. 18(5-6), pp. 602–610 (2005)
5. Graves, A.: Supervised sequence labelling with recurrent neural networks. Phd thesis, Technische Universität München (2008)
6. Hermansky, H., Ellis, D.P.W., Sharma, S.: Tandem connectionist feature extraction for conventional HMM systems. In: Proc. of ICASSP, Istanbul, Turkey, vol. 3, pp. 1635–1638 (2000)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
8. Ketabdar, H., Bourlard, H.: Enhanced phone posteriors for improving speech recognition systems. IDIAP-RR, no. 39 (2008)
9. Rose, R.C., Paul, D.B.: A hidden markov model based keyword recognition system. In: Proc. of ICASSP, Albuquerque, NM, USA, vol. 1, pp. 129–132 (1990)
10. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681 (1997)
11. Wöllmer, M., Eyben, F., Reiter, S., Schuller, B., Cox, C., Douglas-Cowie, E., Cowie, R.: Abandoning Emotion Classes - Towards Continuous Emotion Recognition with Modelling of Long-Range Dependencies. In: Proc. of Interspeech, Brisbane, Australia, pp. 597–600 (2008)
12. Wöllmer, M., Eyben, F., Keshet, J., Graves, A., Schuller, B., Rigoll, G.: Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks. In: Proc. of ICASSP, Taipei, Taiwan (2009)
13. Wöllmer, M., Eyben, F., Schuller, B., Rigoll, G.: Robust vocabulary independent keyword spotting with graphical models. In: Proc. of ASRU, Merano, Italy (2009)