Robust Vocabulary Independent Keyword Spotting with Graphical Models

Martin Wöllmer [1], Florian Eyben [2], Björn Schuller [3], Gerhard Rigoll [4]

Institute for Human-Machine Communication, Technische Universität München, 80333 München, Germany

[1] woellmer@tum.de
[2] eyben@tum.de
[3] schuller@tum.de
[4] rigoll@tum.de

***Abstract*—This paper introduces a novel graphical model architecture for robust and vocabulary independent keyword spotting which does not require the training of an explicit garbage model. We show how a graphical model structure for phoneme recognition can be extended to a keyword spotter that is robust with respect to phoneme recognition errors. We use a hidden garbage variable together with the concept of switching parents to model keywords as well as arbitrary speech. This implies that keywords can be added to the vocabulary without having to re-train the model. Thereby the design of our model architecture is optimised to reliably detect keywords rather than to decode keyword phoneme sequences as arbitrary speech, while offering a parameter to adjust the operating point on the Receiver Operating Characteristics curve. Experiments on the TIMIT corpus reveal that our graphical model outperforms a comparable Hidden Markov Model based keyword spotter that uses conventional garbage modelling.**

I. Introduction

As an important discipline in the field of automatic speech recognition (ASR), keyword spotting has found many applications in recent years. For voice command detection, information retrieval systems, or embodied conversational agents, reliably detecting important keywords is often more important than attempting to capture the whole spoken content of an utterance. Hidden Markov Model (HMM) based keyword spotting systems [1], [2] usually require keyword HMMs and a *filler* or *garbage* HMM to model both, keywords and non-keyword parts of the speech sequence. Using whole word HMMs for the keywords and the garbage model 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. Modelling sub-units of words, such as phonemes, offers the possibility to design a garbage HMM that connects all phoneme models [1]. However, the inherent drawback of this approach is that the garbage HMM can potentially model any phoneme sequence, including the keyword itself. Better garbage models can be trained when modelling non-keyword speech with a large vocabulary ASR system where the lexicon excludes the keyword [3]. Disadvantages of this method are its higher decoding complexity and the large amount of required training data to obtain a reasonable language model. Moreover, large vocabulary ASR systems presume that all keywords are contained in the language model, which makes them less flexible than vocabulary independent systems [4] where no information about the set of keywords is required while training the models.

Apart from the numerous HMM based approaches, more unconventional keyword spotting strategies, such as applying recurrent neural networks [5] or using discriminative learning procedures [6], [7] have been developed. The latter technique non-linearly maps speech features into an abstract vector space which has shown good performance, but requires much more computational power than HMM based methods.

In this paper we present a new graphical model (GM) design which can be used for robust keyword spotting and overcomes most of the drawbacks of other approaches. Graphical models offer a flexible statistical framework that is increasingly applied for speech recognition tasks [8], [9] since it allows for conceptual deviations from the conventional HMM architecture (as in [10] or [11] for example). The GM makes use of the graph theory in order to describe the time evolution of speech as a statistical process and thereby defines 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 [12] 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 optimised 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 [13] where a graphical model indicates possible OOV regions in continuous speech.

In the following sections we introduce the explicit graph representation of a GM based keyword spotter that does not need a trained garbage model and is robust with respect to phoneme recognition errors. Our approach is conceptually more simple than a large vocabulary ASR system since it does not require a language model but only the keyword phonemisations. By introducing a further hierarchy level in a graphical model for phoneme recognition, we present a framework for reliably detecting keywords in continuous speech. Thereby we use a hidden garbage variable and the concept of *switching*

parents [8] to model either a keyword or arbitrary speech.

The structure of this paper is as follows: Section II outlines the graphical model architectures for training and decoding of the keyword spotter. In Section III we evaluate our approach on the TIMIT corpus before drawing conclusions in Section IV.

II. Graphical Model Architectures

Generally speaking, a graphical model $\mathcal{G}(V, E)$ consists of a set of nodes V and edges E, whereas 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 factorisations of the joint probability distribution. Dynamic Bayesian Networks (DBN) are the graphical models of choice for speech recognition tasks, since they consist of repeated template structures over time, modelling the temporal evolution of a speech sequence. Conventional HMM 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, 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 [9], explicit graph representations are advantageous whenever the set of hidden variables has factorisation constraints or consists of multiple hierarchies. In the following sections we illustrate how the inherent benefits of explicit modelling can be exploited for the task of keyword spotting.

A. Training

The graphical model we used to train our keyword spotter is depicted in Figure 1. Compared to the GM that will be applied for decoding (see Section II-B), the GM for the training of the keyword spotter is less complex, since so far, 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).

In conformance with Figure 1, the following random variables are defined for every time step t: q_t^c is a count variable determining the current position in the phoneme sequence, 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 o_t denotes the observed acoustic features. Figure 1 displays hidden variables as circles and observed variables as squares. Deterministic conditional probability functions (CPFs) are represented by straight lines whereas zig-zagged lines correspond to random CPFs. The grey-shaded arrow in Figure 1, pointing from q_{t-1}^{tr} to q_t^c is only valid during the second training cycle when there are no segmentation constraints, and will be ignored in Equations 1 and 2. Assuming a speech sequence of length T, the DBN structure specifies the factorisation

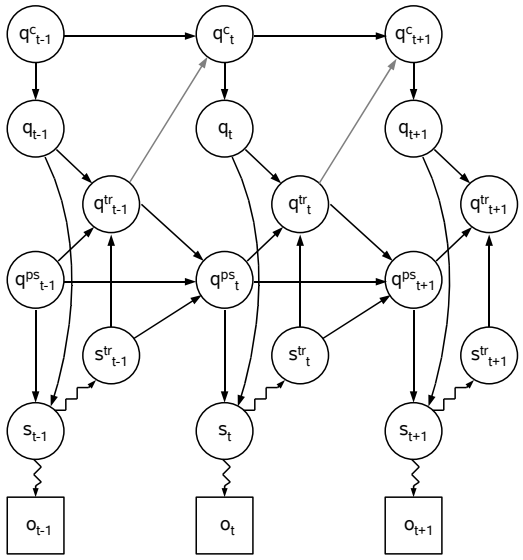

Fig. 1. DBN structure of the graphical model used to train the keyword spotter

$$
\begin{aligned}
&p(q_{1:T}^c, q_{1:T}, q_{1:T}^{tr}, q_{1:T}^{ps}, s_{1:T}^{tr}, s_{1:T}, o_{1:T}) = \\
&\quad \prod_{t=1}^{T} p(o_t|s_t) f(s_t|q_t^{ps}, q_t) p(s_t^{tr}|s_t) f(q_t^{tr}|q_t^{ps}, q_t, s_t^{tr}) f(q_t|q_t^c) \\
&\quad f(q_1^{ps}) f(q_1^c) \prod_{t=2}^{T} f(q_t^{ps}|s_{t-1}^{tr}, q_{t-1}^{ps}, q_{t-1}^{tr}) f(q_t^c|q_{t-1}^c)
\end{aligned}
\tag{1}
$$

with $p(\cdot)$ denoting random conditional probability functions and $f(\cdot)$ describing deterministic CPFs. The probability of the observed sequence can then be computed as

$$
\begin{aligned}
p(o_{1:T}) = \sum_{q_{1:T}^c, q_{1:T}, q_{1:T}^{tr}, q_{1:T}^{ps}, s_{1:T}^{tr}, s_{1:T}} & p(q_{1:T}^c, q_{1:T}, q_{1:T}^{tr}, q_{1:T}^{ps}, \\
& s_{1:T}^{tr}, s_{1:T}, o_{1:T})
\end{aligned}
\tag{2}
$$

whereas the factorisation property given in Equation 1 is exploited in order to optimally distribute the sums over the hidden variables into the products. We therefore used the junction tree algorithm [14] to move the sums as far to the right as possible which reduces computational complexity (see also [8] for a simple example of efficient probabilistic inference). The CPFs $p(o_t|s_t)$ are described by Gaussian mixtures as common in an HMM system. Together with $p(s_t^{tr}|s_t)$, 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 states of a phoneme. This is expressed by the function $f(q_t^{tr}|q_t^{ps}, q_t, s_t^{tr})$. During training, the current phoneme q_t is known, given the position q_t^c in the training utterance, which implies a deterministic mapping $f(q_t|q_t^c)$. In the first training cycle q_t^c is incremented in every time frame, whereas in the second cycle q_t^c is only incremented if $q_{t-1}^{tr} = 1$. 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.

B. Decoding

Once the distributions $p(o_t|s_t)$ and $p(s_t^{tr}|s_t)$ are trained, a more complex GM is used for keyword spotting (see Figure 2): in the decoding phase, the hidden variables w_t, w_t^{ps}, and w_t^{tr} are included in order to model whole words. Further, a hidden *garbage variable* g_t indicates whether the current word is a keyword or not. In Figure 2, dotted lines correspond to so-called *switching parents* [8], which allow a variable's parents to change conditioned on the current value of the switching parent. Thereby a switching parent cannot only change the set of parents but also the implementation (i.e. the CPF) of a parent. Considering all statistical independence assumptions, the graphical model can be factorised as follows:

$$\begin{aligned} &p(g_{1:T}, w_{1:T}, w_{1:T}^{tr}, w_{1:T}^{ps}, q_{1:T}, q_{1:T}^{tr}, q_{1:T}^{ps}, s_{1:T}^{tr}, s_{1:T}, o_{1:T}) = \\ &\prod_{t=1}^{T} p(o_t|s_t) f(s_t|q_t^{ps}, q_t) p(s_t^{tr}|s_t) f(q_t^{tr}|q_t^{ps}, q_t, s_t^{tr}) f(g_t|w_t) \\ &f(w_t^{tr}|q_t^{tr}, w_t^{ps}, w_t) f(q_1^{ps}) p(q_1|w_1^{ps}, w_1, g_1) f(w_1^{ps}) p(w_1) \\ &\prod_{t=2}^{T} f(q_t^{ps}|s_{t-1}^{tr}, q_{t-1}^{ps}, q_{t-1}^{tr}) p(q_t|q_{t-1}^{tr}, q_{t-1}, w_t^{ps}, w_t, g_t) \\ &f(w_t^{ps}|q_{t-1}^{tr}, w_{t-1}^{ps}, w_{t-1}^{tr}) p(w_t|w_{t-1}^{tr}, w_{t-1}) \end{aligned} \tag{3}$$

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 simple 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. However, we introduced differing a priori likelihoods for keywords and garbage phonemes:

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

and

$$p(w_t = 0|w_{t-1}^{tr} = 1) = \frac{1}{K \cdot 10^a + 1}. \tag{5}$$

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.

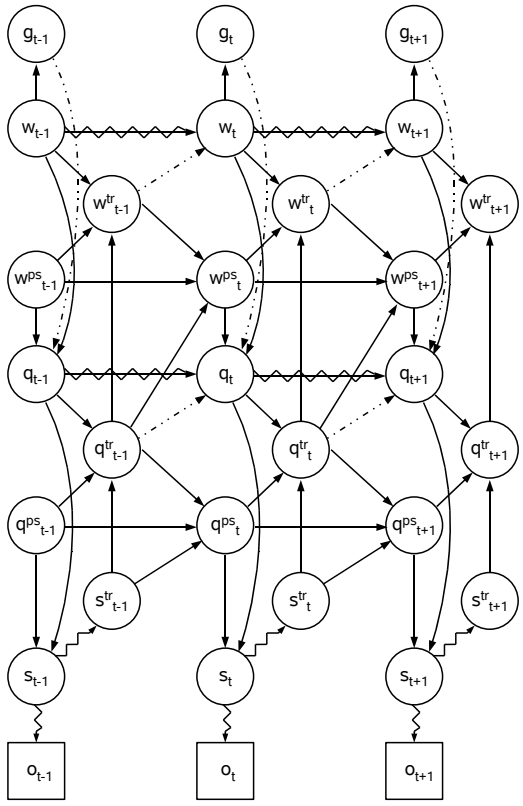

Fig. 2. DBN structure of the graphical model for keyword spotting

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 GM (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} using a trained phoneme bigram $\mathbf{P}$. This phoneme bigram matrix is used to model arbitrary speech and was learnt by simply counting phoneme transitions that occur in a training corpus:

$$\mathbf{P} = \mathbf{N} - f \cdot \mathbf{I} \tag{6}$$

Thereby the bigram matrix $\mathbf{P}$ contains the probabilities

$$P_{ij} = p(q_t = j|q_{t-1}^{tr} = 1, g_t = 1, q_{t-1} = i) \tag{7}$$

that the phoneme j occurs after phoneme i. $\mathbf{N}$ includes the number of phoneme transitions n_{ij}, normalised by the number

N_i of occurrences of the phoneme i in the training corpus, whereas n_{ij} is floored to f:

$$N_{ij} = \max\left(\frac{n_{ij}}{N_i}, \frac{f}{N_i}\right) \quad (8)$$

Since Equation 8 introduces a probability floor value for all possible transitions, the subtraction of the identity matrix $\mathbf{I}$ weighted by f ensures that transitions from phoneme i to phoneme i occur with zero probability.

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 GM 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 $\mathbf{P}$ 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.

III. Experiments

Our GM keyword spotter was trained and evaluated on the TIMIT corpus. The feature vectors consisted of cepstral mean normalised MFCC coefficients 1 to 12, energy, as well as first and second order regression coefficients. The phoneme models were composed of three hidden states each. During the first training cycle of the GM, phonemes were trained framewise using the training portion of the TIMIT corpus. Thereby all Gaussian mixtures were split once 0.02% 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 (embedded training). We randomly chose 60 keywords from the TIMIT corpus to evaluate the keyword spotter GM. The used dictionary allowed for multiple pronunciations. The floor value f (see Equation 8) was set to 10 and the trade-off parameter a (see Equation 4) was varied between 0 and 10.

For comparison, a phoneme based keyword spotter using conventional HMM modelling was trained and evaluated on the same task. Analogous to the GM experiment, each phoneme was represented by three states (left-to-right HMMs) with either 16 or 32 Gaussian mixtures. Thereby we used cross-word triphone models in order to account for contextual information. Like the GM, all phoneme HMMs were re-trained using embedded training. For 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. We thereby allowed for the same keyword pronunciation variants as in the GM experiment. 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 ROC curve we introduced different a priori likelihoods for keyword and garbage HMMs, identical to the word zerogram used for the graphical model. Apart from the transition probabilities implied by the zerogram, the HMM system uses no additional likelihood penalties at the phoneme borders. In this respect our HMM baseline is similar to a system as described in [15].

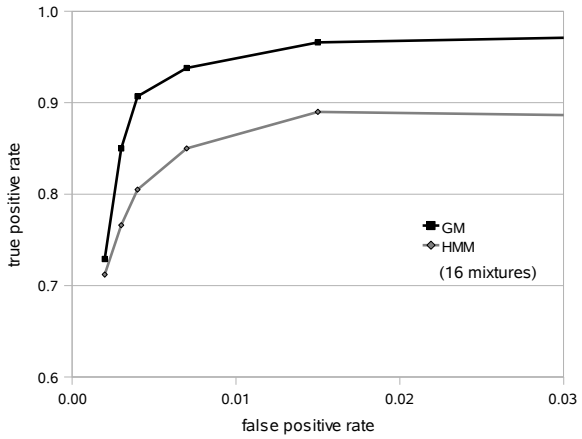

Fig. 3. Part of the ROC curve for the GM- and for the HMM keyword spotter (using 16 Gaussian mixtures per state) – the operating points correspond to the values $a = 0, 1, 2, 3, 5, 10$ (the larger a, the greater the false positive rate)

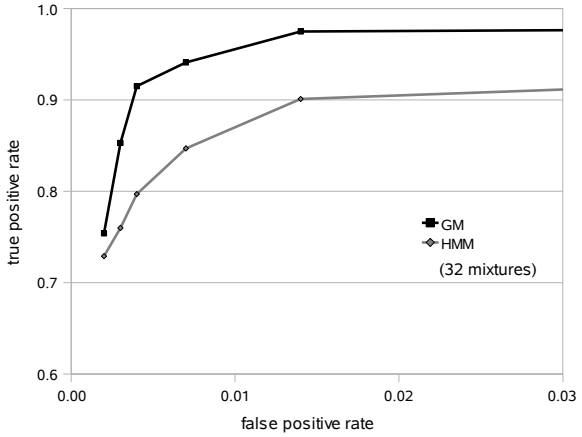

Fig. 4. Part of the ROC curve for the GM- and for the HMM keyword spotter (using 32 Gaussian mixtures per state) – the operating points correspond to the values $a = 0, 1, 2, 3, 5, 10$ (the larger a, the greater the false positive rate)

Figures 3 and 4 show a part of the Receiver Operating Characteristics (ROC) curve for our GM keyword spotter and the HMM based keyword spotter, displaying the true positive rate (tpr) as a function of the false positive rate (fpr) when using 16 mixtures (Figure 3) or 32 mixtures (Figure 4). Note that due to the design of the decoder, 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.

Due to the inherent robustness with respect to phoneme recognition errors, which was outlined at the end of Section II-B, our graphical model architecture achieves significantly higher true positive rates at equal false positive rates, compared

to the trivial HMM approach. We thereby can observe a performance difference of up to 10% (see Figures 3 and 4). Conducting the McNemar's test revealed that the performance difference between the GM keyword spotter and the HMM approach is statistically significant at a common significance level of 0.01. For both decoders 32 mixtures performed slightly better than 16 mixtures.

IV. Conclusion and Discussion

In this work we showed how a graphical model can be used for the task of keyword spotting. We presented the explicit graph representation of a GM that can be used to train phoneme models and extended the graph in a way that a set of defined keywords can be reliably detected in continuous speech. The aim was to encode all model assumptions via hidden variables and conditional probability functions in a unified GM framework and to create a basis for investigating architectural modifications and refinements. A major advantage of using graphical models in general, and *explicit* graph representations in particular, is that they allow for rapid prototyping if the potential of new model architectures shall be investigated (as done in [10] for example). Thus, this work can be seen as an attempt to view the task of spoken term detection from the graphical model perspective – as it is done e.g. in [9] for a simple whole word decoder – and thereby offer all the advantages involved with that point of view.

Our graphical model was designed in a way that it overcomes most of the drawbacks of previous keyword spotting techniques. The model is vocabulary independent meaning that during the training phase no knowledge about the specific set of keywords the system shall be applied for, is necessary. This implies that the GM can be trained on *any* corpus, no matter if and how many times the keywords occur in the training database. It is only in the testing phase that the model needs to know the pronunciations of the keywords. Thereby it is important to notice that even though our concept bases on joint modelling of 'garbage phonemes' and keywords, the same effect of vocabulary independent keyword detection cannot be achieved with a system (and language model respectively) trained on mixed phonetic/word transcriptions, where only non-keywords are represented phonetically. Such a system would again presume that the keywords are contained in the training set and would therefore be less flexible.

Moreover, in contrast to many other approaches, the GM introduced in this paper does not need an explicitly trained garbage model. It rather uses a hidden garbage variable that serves as a switching parent of the phoneme node in the network. Thus, the model can switch between keywords and non-keyword parts of a speech sequence without requiring a model that was trained on 'garbage speech'. Of course the proposed architecture is not the only way to implement a distinction between keywords and garbage speech within the GM framework. If the phoneme node is for example conditioned on further parent nodes, the switching could also be encoded in the conditional probability function of the phoneme node. Yet, in order to reduce the dimensionality of the CPFs, we decided for the proposed technique, using the concept of switching parents.

When evaluating the GM on a keyword detection task using the TIMIT database, we found that our technique can outperform a comparable HMM based system that uses a garbage model which connects all phoneme models. More precisely, the GM approach achieves higher true positive rates since the design of the graphical model implicitly introduces a certain robustness with respect to errors made by the underlying phoneme recognition network.

Future works might investigate alternative GM structures, discriminative learning strategies, or advanced techniques of context modelling (such as Long Short-Term Memory [7]) in order to further improve keyword spotting with graphical models.

Acknowledgment

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] R. C. Rose and D. B. Paul, "A hidden markov model based keyword recognition system," in *Proc. of ICASSP*, Albuquerque, NM, USA, 1990.

[2] H. Ketabdar, J. Vepa, S. Bengio, and H. Boulard, "Posterior based keyword spotting with a priori thresholds," in *Proc. of Interspeech*, Pittsburgh, Pennsylvania, 2006.

[3] M. Weintraub, "Keyword-spotting using sri's decipher large vocabulary speech recognition system," in *Proc. of ICASSP*, Minneapolis, USA, 1993.

[4] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, 2007.

[5] S. Fernandez, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Proc. of ICANN*, Porto, Portugal, 2007, pp. 220–229.

[6] J. Keshet, D. Grangier, and S. Bengio, "Discriminative keyword spotting," in *Workshop on Non-Linear Speech Processing, NOLISP*, Paris, France, 2007.

[7] M. Wöllmer, F. Eyben, J. Keshet, A. Graves, B. Schuller, and G. Rigoll, "Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks," in *Proc. of ICASSP*, Taipei, Taiwan, 2009.

[8] J. A. Bilmes, "Graphical models and automatic speech recognition," *Mathematical Foundations of Speech and Language Processing*, 2003.

[9] J. A. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 89–100, 2005.

[10] M. Wöllmer, F. Eyben, A. Graves, B. Schuller, and G. Rigoll, "A Tandem BLSTM-DBN architecture for keyword spotting with enhanced context modeling," in *Proc. of NOLISP*, Vic, Spain, 2009.

[11] M. Wöllmer, F. Eyben, B. Schuller, Y. Sun, T. Moosmayr, and N. Nguyen-Thien, "Robust in-car spelling recognition - A Tandem BLSTM-HMM approach," in *Proc. of Interspeech*, Brighton, UK, 2009.

[12] H. Lin, A. Stupakov, and J. Bilmes, "Spoken keyword spotting via multi-lattice alignment," in *Proc. of ICASSP*, Taipei, Taiwan, 2009.

[13] H. Lin, J. Bilmes, D. Vergyri, and K. Kirchhoff, "OOV detection by joint word/phone lattice alignment," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Kyoto, Japan, 2007, pp. 478–483.

[14] F. V. Jensen, *An introduction to Bayesian Networks*. Springer, 1996.

[15] F. Jelinek, *Statistical Methods for Speech recognition*. MIT Press, Cambridge, 1997.