INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Simultaneous Detection and Localization of a Wake-Up Word using Multi-Task Learning of the Duration and Endpoint

Takashi Maekaku, Yusuke Kida, Akihiko Sugiyama

Yahoo Japan Corporation, Japan
{tmaekaku,ykida,c-aksugiyama}@yahoo-corp.jp

Abstract

This paper proposes a novel method for simultaneous detection and localization of a wake-up word using multi-task learning of the duration and endpoint. An onset of the wake-up word is estimated by going back in time by an estimated duration of the wake-up word from an estimated endpoint. Accurate endpoint estimation is achieved by training the network to fire only at the endpoint in contrast to the entire wake-up word. The accurate endpoint naturally leads to an accurate onset, when it is used as a basis to calculate an onset with an estimated duration that reflects the whole acoustic information over the entire wake-up word. Experimental results with real-environment data show that a relative improvement in accuracy of 41% for onset estimation and 38% for endpoint estimation are achieved compared to a baseline method.

Index Terms: keyword detection, keyword localization, onset, endpoint, LSTM

1. Introduction

Keyword detection (KWD) is a technique to detect a certain keyword (wake-up word) from an acoustic signal. It is widely used to start speech recognition in smart speakers or car navigation systems while keeping a user's hands free. Various KWD methods have been proposed in the last couple of decades. Classical approaches include dynamic time warping (DTW) based system [1, 2] and hidden Markov model (HMM) techniques [3, 4]. However, these approaches are not suitable for inexpensive devices with limited power and computational resources because of the computation required. Small-footprint techniques based on deep neural networks (DNN) [5]–[10] are lighter in computation, and have been reported to achieve better recognition rates than DTW-based and HMM-based methods.

In recent years, some approaches to use acoustic features in a segment of a detected wake-up word for speech recognition have been proposed. King *et al.* used acoustic features in the segment to normalize that of the following speech command segment [11]. Kida *et al.* used the acoustic features in the segment for direction-of-arrival estimation of the speech command [12]. To realize these methods, it is necessary to localize a wake-up word accurately in addition to detecting the wake-up word. However, there is no paper which presents how to detect and localize a wake-up word simultaneously.

It is easy for the conventional KWD algorithms using DTW or HMM to estimate a wake-up word segment because time alignment information at a phoneme level is available through detection. Nevertheless, the conventional KWD algorithms are not good at wake-up word detection itself in contrast to the DNN-based algorithms. On the other hand, localization of a wake-up word is not an easy task for the small foot-print approaches based on DNNs. They directly estimate a posterior probability of the wake-up word with no time alignment information. It is therefore essential to develop a method to simultaneously detect and localize a wake-up word.

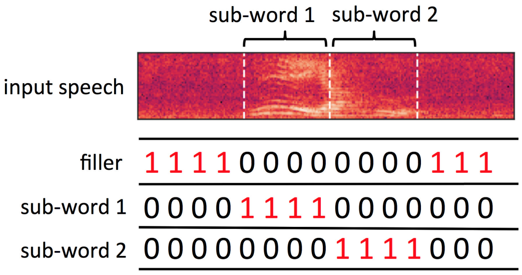

Figure 1: *An example of target labels for Deep KWS.*

This paper proposes simultaneous detection and localization of a wake-up word using multi-task learning of the duration and the endpoint. An onset of the wake-up word is estimated by going back in time by the estimated duration of the wake-up word from an estimated endpoint. Accurate endpoint estimation is achieved by training the network to fire only at the endpoint rather than the entire wake-up word. The accurate endpoint naturally results in an accurate onset, when it is used as the basis to calculate an onset with the duration that reflects the whole acoustic information over the entire wake-up word. The following section reviews a baseline system followed by the proposed method in Section 3. Experimental setup and results are presented in Section 4.

2. Deep Keyword Spotting

Deep keyword spotting (Deep KWS) systems [5] are widely used for wake-up word detection. It uses acoustic features extracted from short speech segments as an input to learn a neural network to calculate a frame-wise posterior probability of the wake-up word. Figure 1 illustrates an example of target labels for deep KWS. The network has three output nodes, namely, filler, sub-word1, and sub-word2 and is trained to classify the three. A wake-up word is split into sub-word1 and sub-word2 in this example. Filler corresponds to every frame but those of sub-word1 and sub-word2. The output posterior probability for each node is averaged over the past 30 frames including the current frame to obtain a smoothed posterior probability. The maximum smoothed posterior probabilities for sub-words in the past 100 frames are used to calculate a confidence score in each frame as a geometric mean of the two posterior probabilities. The wake-up word is detected when the confidence score exceeds a predefined threshold. Otherwise, it is declared as filler. It means that wake-up word detection is performed in a frame-wise manner and each frame is assigned 1 as the wake-up word or 0 otherwise.

http://dx.doi.org/10.21437/Interspeech.2019-1180

Deep KWS is superior in terms of detection accuracy compared to conventional approaches. However, it is difficult for Deep KWS to estimate the onset and the endpoint of the wake-up word from a series of posterior probabilities. This is because the same labels are likely to be assigned to consecutive frames to cover the whole wake-up word or sub-word. The labels simply show the presence of the wake-up word and are not designed to highlight the onset or the endpoint.

3. Proposed Method

The proposed method detects the wake-up word by its endpoint and estimates its onset by an estimated duration of the wake-up word thereby simultaneously detecting and localizing the wake-up word. This approach reflects the facts that endpoint detection with rich past information is more accurate than onset detection and that the duration has a small variation for a pre-determined wake-up word. Figure 2 illustrates the architecture of the proposed method. It consists of two LSTM (long short-term memory) [13] layers and the output layer is separated into two sub-layers. The left sub-layer, called "DEE (Detection & Endpoint Estimation) task," is used for wake-up-word detection and its endpoint estimation. The right sub-layer, called "OE (Onset Estimation) task," is used for onset estimation. The entire network is trained with multi-task learning [14]. The loss function $\mathcal{L}$ is calculated as a weighted sum of a loss of the DEE task, $\mathcal{L}_{\mathrm{DEE}}$ and a loss of the OE task, $\mathcal{L}_{\mathrm{OE}}$ as:

$$\mathcal{L} = \rho\mathcal{L}_{\mathrm{DEE}}(\mathbf{W}) + (1-\rho)\mathcal{L}_{\mathrm{OE}}(\mathbf{W}), \tag{1}$$

where $\mathbf{W}$ indicates the LSTM parameters and ρ is a mixing weight.

3.1. DEE task

The purpose of the DEE task is to train the neural network to fire at the end of the wake-up word. This is because a detection point of the wake-up word could be regarded as an endpoint of the wake-up word. For this reason, the proposed method has two separate nodes, namely, wake-up word as the endpoint of the wake-up word and filler as otherwise. Unlike Deep KWS, it is not necessary to use sub-words. An example of the target labels is shown in Fig. 3 (a). Only some frames near the true endpoint of the wake-up word are labeled as wake-up word. LSTMs are used to model longer time dependencies. The network tries to directly detect the endpoint of the wake-up word instead of the wake-up word itself. Practically, in order to compensate for an endpoint estimation error, we modify an estimated endpoint t_{kwd} by a pre-defined offset $\Delta_{\mathrm{endpoint}}$ to obtain a corrected endpoint as $t_{\mathrm{kwd}} + \Delta_{\mathrm{endpoint}}$. The pre-defined offset $\Delta_{\mathrm{endpoint}}$ is optimized by preliminary experiments.

3.2. OE task

If we find an onset in the same way as the endpoint estimation described earlier, it does not work because the LSTM network locates an onset before having gone through the wake-up word. Instead, we estimate time duration of the wake-up word while detecting the wake-up word presence simultaneously. Once the duration estimation is completed, the onset could be estimated by simply subtracting the estimated duration from the endpoint. Besides, the proposed method has a low-latency advantage because an onset is estimated almost at the same time as the wake-up word detection. Preliminary experiments showed that direct learning of wake-up word duration as a regression task sometimes become instable. Therefore, we implement this task as classification. In particular, the duration is divided into multiple classes with the same length for training to achieve successful classification.

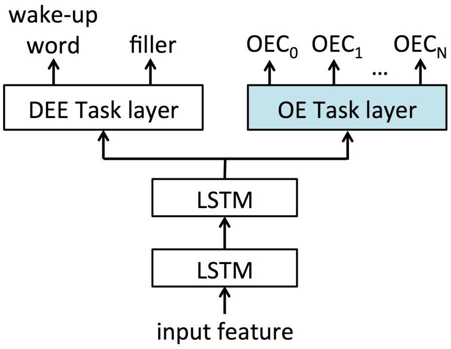

Figure 2: *Illustration of our proposed network.*

First, KWD and the endpoint estimation are performed in the way described in the previous section. Then, a class with the largest posterior probability among all the classes from the OE task is identified. As seen in Fig. 3 (b), the OE task has $N+1$ output nodes each of which corresponds to a single divided duration. For example, a class OEC_1 corresponds to the duration from the 1st to the 3rd frame, and a class OEC_2 corresponds to the duration from the 4th to the 6th frame. The number of the output nodes or equivalently, the divided duration, is determined based on the wake-up word length. OEC_0 represents a node which corresponds to everything but the wake-up word section. Let us denote a posterior probability of class n in frame t as $p_n(t)$. With $\arg\max_{n>0} p_n(t)$ as a class n with the maximum likelihood of frame t, the onset t_{onset} is calculated from the following equation.

$$t_{\mathrm{onset}} = t_{\mathrm{kwd}} - \arg\max_{n>0} p_n(t_{\mathrm{kwd}}) \cdot d + \Delta_{\mathrm{onset}}, \tag{2}$$

where d is the number of frames for each class, and Δ_{onset} is an offset parameter. For example, if the class n which has the largest posterior is 10th node where $t_{\mathrm{kwd}} = 80$, $d = 3$, and $\Delta_{\mathrm{onset}} = 0$, the onset t_{onset} is calculated as $80 - 10 \times 3 + 0 = 50$.

4. Experiments

4.1. Datasets

Training data contained a total of 500K utterances consisting of 100K as the wake-up word and 400K as filler. The wake-up words were real clean speech and the filler was collected from the Internet by Yahoo! Japan Voice Search and Personal Assistant apps. The convolution of each utterance and a room impulse response (RIRs) was calculated and noise was added to the convolution to generate a single training datum. A total of 198,00 RIRs were generated by the image method [15] one of which was randomly selected for convolution. The reverberation time RT60 of the RIRs ranged from 170 to 710 ms. The noise set included TV noise, radio noise, and music recorded in a living room. An SNR upon noise addition was randomly set to a value between [6, 16] dB. The training data were used for network learning. The test data set had 10,895 recorded far-field signal consisting of 895 wake-up word utterances and 10,000 filler utterances with no wake-up word. We used one third of

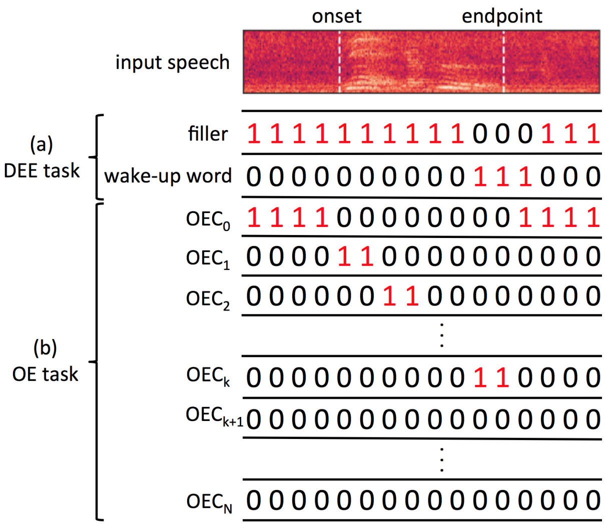
onset
endpoint
input speech
(a)
DEE task
filler 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1
wake-up word 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
(b)
OE task
OEC_0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1
OEC_1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
OEC_2 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
OEC_k 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
OEC_{k+1} 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
OEC_N 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Figure 3: *Target labels for output neurons.*

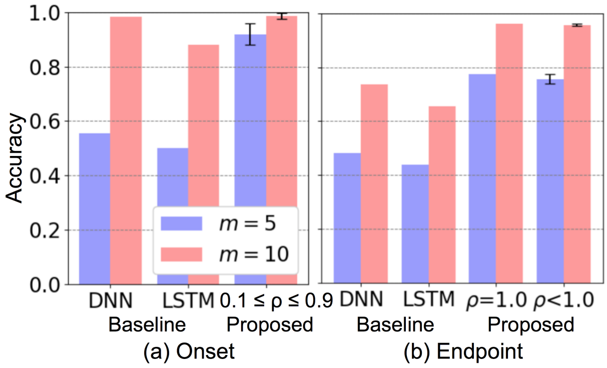

Figure 4: *Accuracy of onset/endpoint estimation for the DNN baseline, the LSTM baseline, and the proposed method. A vertical line on each bar-end represents the distribution of the estimated onsets/endpoint percentage by the proposed method with* $0.1 \leq \rho \leq 1.0$

the test data to optimize the offset values for the onset and the endpoint, and two-thirds as an evaluation set.

4.2. Experimental Setup

We used 40-dimensional log mel-frequency filterbank (LMFB) coefficients with a frame length of 25 ms and a frame shift of 10 ms as input features to the network. The mean and variance normalization (MVN) [16] was applied to the features. The initial value of MVN was calculated from the training data, and updates of the mean were performed online. To reduce computational complexity, the lower frame-rate technique [17] was applied to feature extraction so that wake-up word features are calculated every three frames. Weight parameters of all methods were trained with the cross-entropy criterion [18].

4.2.1. Baseline Methods

Two baseline methods based on Deep KWS [5] with a DNN and an LSTM were included. Input features as a 40 dimension column vector in the current frame were spliced with those of the past 30 frames and the future 10 frames to make a 41 × 40 matrix. The DNN network had 2 layers of 128 hidden units. The wake-up word "Néé Yahoo! (Hey, Yahoo!)" was divided into 2 sub-words "Néé" and "Yahoo". The network was trained to classify three output nodes which consist of the two sub-words and filler. The LSTM had 2 layers of 96 hidden units for the reason described in Section 4.2.2. For LSTM training, multiple frames to cover the entire wake-up word were labeled as 1 (wake-up word) and all other frames as 0 (filler), resulting in two output nodes. This is because LSTM models longer time dependencies than DNN. An onset and an endpoint of the wake-up word were estimated by adding different offset values to the frame indices representing the start and the end of a section with wake-up word posterior probability greater than a predefined threshold.

4.2.2. Proposed method

Input features of a 40 dimensional column vector in the current frame were spliced with those of the past 4 frames, which resulted in a 5 × 40 matrix. N and d were set to 25 and 2, respectively, in order to cover at most a 1.5 sec long wake-up word and achieve good performance in KWD in preliminary experiments. As a result, the number of output nodes was 28. The network size of the proposed method was set smaller than that of the DNN baseline to reduce complexity but avoid reduction in KWD performance. It was also important for comparison that the network size of the proposed method is equal to that of the LSTM baseline method except for the number of output nodes. The proposed method with $\rho = 1.0$ reduces to the LSTM baseline except for endpoint estimation as is clear from (1). The mixing weight ρ for the new loss function in (1) was varied between 0 and 1.0 with 0.1 increments.

4.3. Results and discussion

4.3.1. Accuracy of Keyword Endpoint/Onset Estimation

Figure 4 compares accuracy of onset/endpoint estimation with the DNN baseline, the LSTM baseline, and the proposed method. Each bar represents a percentage of estimated onsets/endpoints which fall within $\pm m$ frames of the correct onset/endpoint. The left and the right bar in each pair are the result with $m = 5$ and $m = 10$. Figure 4 (a) on the left hand side and (b) on the right hand side represent the accuracy for onset and endpoint estimation.

It is clear from Fig. 4 (b) that the proposed method with any value of ρ outperforms the DNN and the LSTM baseline. Compared to the DNN baseline, the proposed method achieves a relative improvement of as much as 38% in accuracy for $m = 5$. This is a sign that labeling frames as the target only in the vicinity of the endpoint has a significant effect. Accuracy of the proposed method with $\rho < 1.0$ is comparable to that with $\rho = 1.0$. It indicates that multi-task training with the OE task does not have a negative impact on the endpoint estimation.

The accuracy of onset estimation, calculated in the same manner as for the endpoint estimation, is shown in Fig. 4 (a). Please note that the proposed method with $\rho = 1.0$ is not included in this experiment because $\rho = 1.0$ indicates that the OE (onset estimation) task is eliminated from (1). The accuracy of the proposed method clearly outperforms both of the baseline methods for $m = 5$ and 10. Compared to the DNN baseline, the proposed method with $\rho \leq 0.9$ achieves a relative improvement of 41% for $m = 5$. The short line crossing the bar end in Fig. 4 (a) indicate that the proposed method with

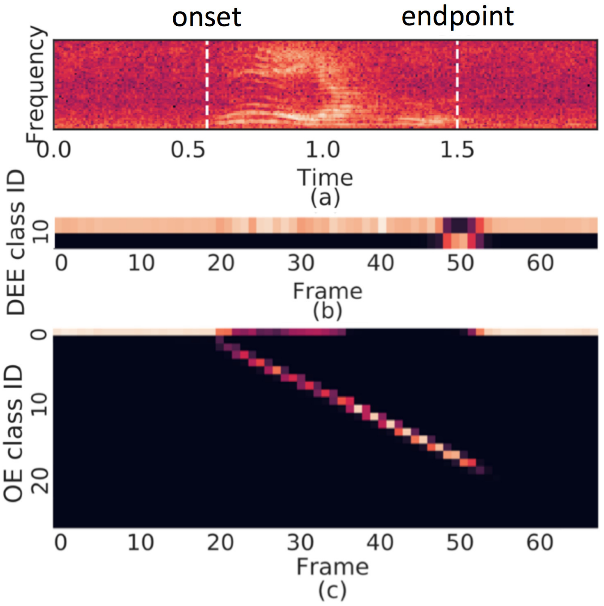

Figure 5: *Posteriorgram by the proposed method. (a) Input spectrogram, (b) DEE task (2 nodes), (c) OE task (28 nodes).*

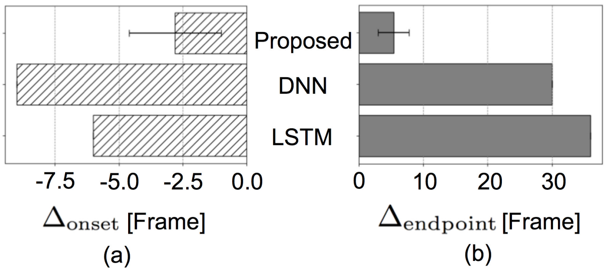

Figure 6: *Optimized offset Δ for onset (a) and endpoint (b) for DNN, LSTM, and the proposed method. The length of each bar is a mean and a line across the bar-end represents range of Δ with different ρ.*

$\rho \leq 0.9$ achieves high accuracy regardless of the value of ρ.

An output posteriorgram of the DEE task and the OE task are shown in Fig. 5 (b) and (c) with an input spectrogram in (a). The brighter color means a larger posterior. Fig. 5 (b) shows that the posterior probability for DEE Class ID 1 (the wake-up word endpoint) is concentrated around frame 50 which is the true endpoint. On the contrary, the posterior probability for DEE Class ID 0 is evenly distributed except around frame 50. As shown in Fig. 5 (c), the class ID with the highest posterior for the OE task, which represents the wake-up word duration, increases with time. This suggests that the network training by the proposed method for estimating a wake-up word duration is successful, which enables accurate onset estimation as was already demonstrated in Fig. 4 (a).

Optimized offset Δ for the onset (a) and the endpoint (b) for the conventional methods and the proposed method are compared in Fig. 6 as a bar chart. The length of each bar is a mean value and a horizontal line across the bar end represents the range of Δ for the proposed method with different values of ρ. Figure 6 (b) clearly shows that offset values for the endpoint with the proposed method are significantly smaller compared to those with the baseline methods. It means that the estimated endpoint is sufficiently correct so that only a small shift to cover the true endpoint is needed. This observation also applies to Fig. 6 (a) with respect to onset estimation. Moreover, a narrow range of the offset values for both onset and endpoint for different values of ρ indicates that the proposed method is insensitive to the selection of ρ. The proposed method is more accurate in onset/endpoint estimation than the two baseline methods and insensitive to the value of the mixing weight ρ.

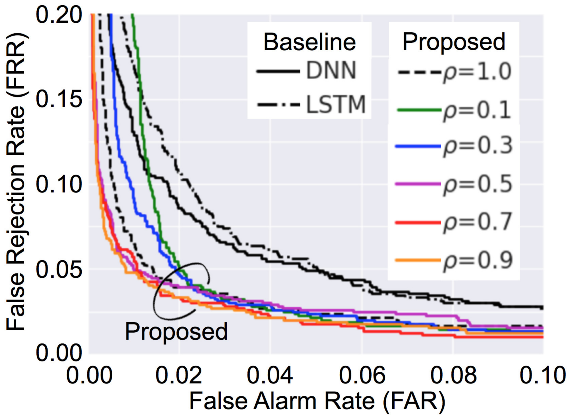

Figure 7: *ROC curves for KWD of different methods on data obtained from real environments.*

4.3.2. Performance of KWD

We also evaluated performance of KWD. The experimental results are shown in Fig. 7. The horizontal axis indicates the false alarm rate (FAR) and the vertical axis does the false rejection rate (FRR). FAR is the percentage of non-wake-up word frames incorrectly classified as wake-up word, and FRR is the percentage of wake-up word frames incorrectly classified as non-wake-up word. The receiver-operating-characteristic (ROC) curves is plotted by varying the threshold value of the estimated posterior.

We can see that the proposed method with $\rho = 1.0$ significantly better performance than that of LSTM baseline. This result indicates that the labelling manner of the proposed method also positively affected to KWD performance. This is not surprising because a word cannot be regarded as the wake-up word until its endpoint is observed. The figure also demonstrates that the proposed method with $\rho \geq 0.5$ shows further improvement compared to that with $\rho = 1.0$. This result suggests that the OE task has a complementary effect on the DEE task. We think that the OE task works as a constraint because the duration of wake-up word has a small variation.

5. Conclusion

In this paper, we have proposed a novel method for simultaneous detection and localization of a wake-up word. An onset of the wake-up word is estimated by going back in time by an estimated duration of the wake-up word from an estimated endpoint. Accurate endpoint estimation is achieved by network training to fire only at the endpoint in contrast to the entire wake-up word. The experimental results with real-environment data show that a relative improvement in accuracy of 41% for onset estimation and 38% for endpoint estimation are achieved compared to a baseline method. It is observed that the proposed method outperforms the baseline methods with respect to the performance in keyword detection.

6. References

[1] K. Levin, K. Henry, A. Jansen, and K. Livescu, “Fixed dimensional acoustic embeddings of variable-length segments in low-resource settings,” in *Proc. ASRU*, 2013, pp. 410–415.

[2] Y. Zhang and J. R. Glass, “Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams,” in *Proc. ASRU*, 2009, pp. 398–403.

[3] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, “Continuous hidden Markov modeling for speaker-independent word spotting,” in *Proc. ICASSP*, 1989, pp. 627–630.

[4] R. C. Rose and D. B. Paul, “A hidden Markov method based keyword recognition system,” in *Proc. ICASSP*, 1990, pp. 129–132.

[5] G. Chen, C. Parada, and G. Heigold, “Small-footprint keyword spotting using deep neural networks,” in *Proc. ICASSP*, 2014, pp. 4087–4091.

[6] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, “Model compression applied to small-footprint keyword spotting,” in *Proc. INTERSPEECH*, 2016, pp. 1878–1882.

[7] K. J. Lang, A. Waibel, and G. E. Hinton, “A time-delay neural network architecture for isolated word recognition,” *Neural Networks*, vol. 3, no. 1, pp. 23–43, 1990.

[8] M. Sun et al., “Compressed time delay neural network for small-footprint keyword spotting,” in *Proc. INTERSPEECH*, 2017, pp. 3607–3611.

[9] M. Chen, S. Zhang, M. Lei, Y. Liu, H. Yao, and J. Gao, “Compact Feedforward Sequential Memory Networks for Small-footprint Keyword Spotting,” in *Proc. INTERSPEECH*, 2018, pp. 2663–2667.

[10] C. Shan, J. Zhang, Y. Wang, and L. Xie, “Attention-based End-to-End Models for Small-Footprint Keyword Spotting,” in *Proc. INTERSPEECH*, 2018, pp. 2037–2041.

[11] B. King, *et al*., “Robust Speech Recognition Via Anchor Word Representations,” in *Proc. INTERSPEECH*, 2017, pp. 2471–2475.

[12] Y. Kida, D. Tran, M. Omachi, T. Taniguchi, and Y. Fujita, “Speaker selective beamformer with keyword mask estimation,” in *Proc. SLT*, 2018.

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[14] R. Caruana, “Multitask learning: A knowledge-based source of inductive bias,” in *Proc. Machine Learning*, 1993.

[15] J.B. Allen and D.A. Berkley, “Image method for efficiently simulating small-room acoustics,” *J. Acoustic Society of America*, vol. 65, no. 4, pp. 943, Apr. 1979.

[16] O. Viikki and K. Laurila, “Cepstral domain segmental feature vector normalization for noise robust speech recognition,” *Speech Communication*, vol. 25, no. 1-3, pp. 133–147, Aug. 1998.

[17] G. Pundak and T. N. Sainath, “Lower frame rate neural network acoustic methods,” in *Proc. INTERSPEECH*, 2016, pp. 22–26.

[18] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” *MIT Press*, 2016, pp. 215.