Interspeech 2018
2-6 September 2018, Hyderabad

Attention-based End-to-End Models for Small-Footprint Keyword Spotting

Changhao Shan[1,2], *Junbo Zhang*[2], *Yujun Wang*[2], *Lei Xie*[1*]

[1]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Xiaomi Inc., Beijing, China

{chshan, lxie}@nwpu-aslp.org, {zhangjunbo, wangyujun}@xioami.com

Abstract

In this paper, we propose an attention-based end-to-end neural approach for small-footprint keyword spotting (KWS), which aims to simplify the pipelines of building a production-quality KWS system. Our model consists of an encoder and an attention mechanism. Using RNNs, the encoder transforms the input signal into a high level representation. Then the attention mechanism weights the encoder features and generates a fixed-length vector. Finally, by linear transformation and softmax function, the vector becomes a score used for keyword detection. We also evaluate the performance of different encoder architectures, including LSTM, GRU and CRNN. Experiments on wake-up data show that our approach outperforms the recent Deep KWS approach [9] by a large margin and the best performance is achieved by CRNN. To be more specific, with ∼84K parameters, our attention-based model achieves 1.02% false rejection rate (FRR) at 1.0 false alarm (FA) per hour.

Index Terms: attention-based model, end-to-end keyword spotting, convolutional neural networks, recurrent neural networks

1. Introduction

Keyword spotting (KWS), or spoken term detection (STD), is a task to detect pre-defined keywords in a stream of audio. Specifically, as a typical application of KWS, wake-up word detection has become an indispensable function on various devices, in order to enable users to have a fully hands-free experience. A practical on-device KWS module must minimize the false rejection rate at a low false alarm rate to make it easy to use, while limiting the memory footprint, latency and computational cost as small as possible.

As a classic solution, large vocabulary continuous speech recognition (LVCSR) based systems [1, 2] are widely used in the KWS task. Although it is flexible to change keywords according to user's requirement, the LVCSR based systems need to generate rich lattices and high computational resources are required for keyword search. These systems are often designed to search large databases of audio content. Several recent attempts have been proposed to reduce the computational cost, e.g., using end-to-end based acoustic models [3, 4]. But these models are still quite large, making them not suitable for small-footprint, low-latency applications. Another classic technique for KWS is the keyword/filler hidden Markov model (HMM) approach [5], which remains strongly competitive today. HMMs are trained for both keyword and non-keyword audio segments, respectively. At runtime, Viterbi decoding is used to search the best path in the decoding graph, which can be computationally expensive depending on the HMM topology. In these approaches, Gaussian mixture models (GMMs) were originally used to model the observed acoustic features, but with the advances in deep learning, deep neural networks (DNNs) have been recently adopted to substitute GMMs [6] with improved performances. Some studies replaced HMM by an RNN model trained with connectionist temporal classification (CTC) criterion [7] or by an attention-based model [8]. However, these studies are still under the keyword/filler framework.

As a small footprint approach used by Google, Deep KWS [9] has drawn much attention recently. In this approach, a simple DNN is trained to predict the frame-level posteriors of sub-keyword targets and fillers. When a confidence score, produced by a posterior handing method, exceeds a threshold, a keyword is detected. This approach has shown to outperform a keyword/filler HMM approach. In addition, this approach is highly attractive to run on the device with small footprint and low latency, as the size of the DNN can be easily controlled and no graph-search is involved. Later, feed-forward DNNs were substituted by more powerful networks like convolutional neural networks (CNNs) [10] and recurrent neural networks (RNNs) [11], with expected improvements. It should be noted that, although the framework of Deep KWS is quite simple, it still needs a well-trained acoustic model to obtain frame-level alignments.

In this paper, we aim to further simplify the pipelines of building a production-quality KWS. Specifically, we propose an attention-based end-to-end neural model for small-footprint keyword spotting. By saying *end-to-end*, we mean that: (1) a simple model that directly outputs keyword detection; (2) no complicated searching involved; (3) no alignments needed beforehand to train the model. Our work is inspired by the recent success of attention models used in speech recognition [12, 13, 14], machine translation [15], text summarization [16] and speaker verification [17]. It is intuitive to use attention mechanism in KWS: humans are able to focus on a certain region of an audio stream with "high resolution" (e.g., the listener's name) while perceiving the surrounding audio in "low resolution", and then adjusting the focal point over time.

Our end-to-end KWS model consists of an *encoder* and an *attention* mechanism. Using RNNs, the encoder transforms the input signal into a high level representation. Then the attention mechanism weights the encoder features and generates a fixed-length vector. Finally, by linear transformation and softmax function, the vector becomes a score used for keyword detection. In terms of end-to-end and small-footprint, the closest approach to ours is the one proposed by Kliegl *et al.* [18], where a convolutional recurrent neural network (CRNN) architecture is used. However, the latency introduced by its long decoding window (T=1.5 secs) makes the system difficult to use in real applications.

To improve our approach, we further explore the encoder

[*]Corresponding author.

10.21437/Interspeech.2018-1777

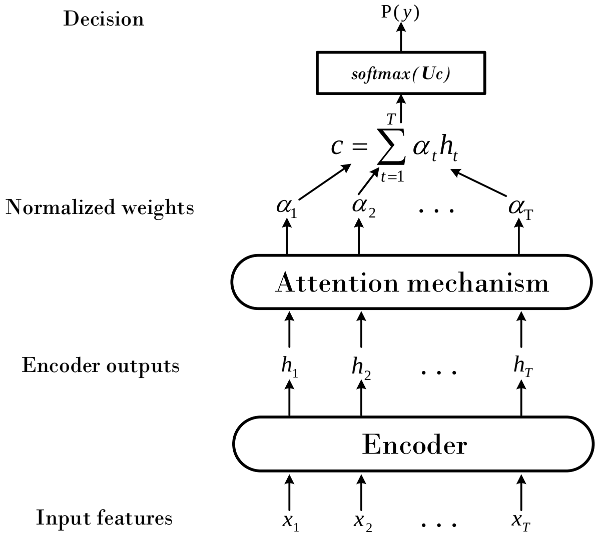

Figure 1: *Attention-based end-to-end model for KWS.*

architectures, including LSTM [19], GRU [20] and CRNN [18]. Experiments on wake-up data show that our approach outperforms Deep KWS by a large margin. GRU is preferred over LSTM and the best performance is achieved by CRNN. To be more specific, with only $\sim$84K parameters, the CRNN-based attention model achieves 1.02% false rejection rate (FRR) at 1.0 false alarm (FA) per hour.

2. Attention-based KWS

2.1. End-to-end architecture

We propose to use attention-based end-to-end model in small-footprint keyword spotting. As depicted in Fig. 1, the end-to-end architecture consists of two major sub-modules: the encoder and the attention mechanism. The encoder results in a higher-level feature representation $\mathbf{h} = (h_1, ..., h_T)$ from the input speech features $\mathbf{x} = (x_1, ..., x_T)$:

$$\mathbf{h} = Encoder(\mathbf{x}). \tag{1}$$

Specifically, the $Encoder$ is usually a RNN that can directly make use of speech contextual information. In our work, we explore different encoder structures, including GRU, LSTM and CRNN. The attention mechanism learns normalized weights $\alpha_t \in [0, 1]$ from the feature representation:

$$\alpha_t = Attend(\boldsymbol{h}_t). \tag{2}$$

Then we form fixed-length vector c as the weighted average of the $Encoder$ outputs $\mathbf{h}$:

$$\boldsymbol{c} = \sum_{t=1}^{T} \alpha_t \boldsymbol{h}_t. \tag{3}$$

Finally, we generate a probability distribution by a linear transformation and the softmax function:

$$p(y) = softmax(\boldsymbol{U}\boldsymbol{c}). \tag{4}$$

where $\mathbf{U}$ is the linear transform, y indicate whether a keyword detected.

2.2. Attention mechanism

Similar to human listening attention, the attention mechanism in our model selects the speech parts which are more likely to contain the keyword while ignoring the unrelated parts. We investigate both average attention and soft attention.

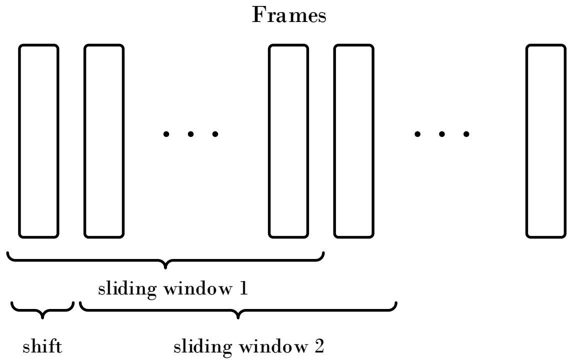

Figure 2: *Sliding windows used in decoding.*

Average attention: The $Attend$ model does not have trainable parameters and the α_t is set as the average of T:

$$\alpha_t = 1/T. \tag{5}$$

Soft attention: This attention method is borrowed from speaker verification [17]. Compared with other attention layers, the shared-parameter non-linear attention is proven to be effective [17]. We first learn a scalar score e_t:

$$e_t = v^T tanh(\boldsymbol{W}h_t + \boldsymbol{b}). \tag{6}$$

Then we compute the normalized weight α_t using these scalar scores:

$$\alpha_t = \frac{exp(e_t)}{\sum_{j=1}^{T} exp(e_j)}. \tag{7}$$

2.3. Decoding

As shown in Fig. 1, unlike some other approaches [9], our end-to-end system outputs a confidence score directly without post-processing. Similar to the Deep KWS system, our system is triggered when $p(y = 1)$ exceeds a preset threshold. During decoding, in Fig. 2, the input is a sliding window of speech features, which has a preset length and contains the entire keyword. Meanwhile, a frame shift is employed. For a sliding window, we only need to feed one frame into the network for computation and the rest frames have been computed already in the previous sliding window. From Table 1, we can clearly see that the small set of parameters of the proposed approach leads to small-footprint memory use while limited multiply operations result in low latency in KWS.

Table 1: *The number of parameters and multiplies.*

Model	Params (K)	Multiplies (K)
DNN KWS [9]	244	244
CNN KWS [10]	47.6	428.5
Attention-based KWS	77.5	83.3

3. Experiments

3.1. Datasets

We evaluated the proposed approach using wake-up data collected from Mi AI Speaker[1]. The wake-up word is a four-syllable Mandarin Chinese term (“xiao-ai-tong-xue”). We collected $\sim$188.9K positive examples ($\sim$99.8h) and $\sim$1007.4K

[1] https://www.mi.com/aispeaker/

Table 2: *Performance comparison between Deep KWS and attention-based models with 2-64 network. FRR is at 1.0 false alarm (FA) per hour.*

Model	FRR (%)	Params (K)
DNN KWS	13.9	62.5
LSTM KWS	7.10	54.1
LSTM average attention	4.43	60.0
LSTM soft attention	3.58	64.3
GRU KWS	6.38	44.8
GRU average attention	3.22	49.2
GRU soft attention	**1.93**	53.4

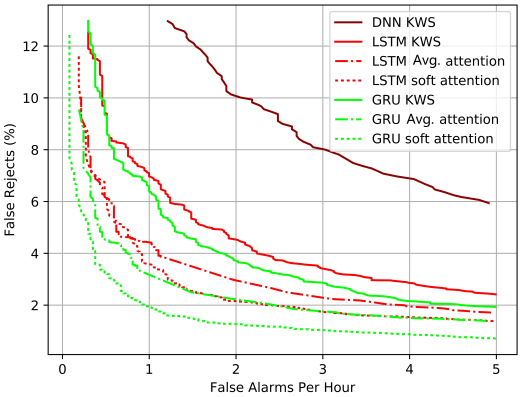

Figure 3: *ROCs for Deep KWS vs. Attention-based system with 2-64 network.*

negative examples (∼1581.8h) as the training set. The held-out validation set has ∼9.9K positive examples and ∼53.0K negative examples. The test data set has ∼28.8K positive examples (∼15.2h) and ∼32.8K negative examples (∼37h). Each audio frame was computed based on a 40-channel Mel-filterbank with 25ms windowing and 10ms frame shift. Then the filterbank feature was converted to per-channel energy normalized (PCEN) [21] Mel-spectrograms.

3.2. Baseline

We reimplemented the Deep KWS system [9] as the baseline, in which the network predicts the posteriors for the four Chinese syllables in the wake-up word and a filler. The “filler” here means any voice that does not contain the keyword. Specifically, we adopted three different networks, including DNN, LSTM and GRU. For a fare comparison, the network configuration was set to have similar size of parameters with the proposed attention models. The feed-forward DNN model had 3 hidden layers and 64 hidden nodes per layer with rectified linear unit (ReLU) non-linearity. An input window with 15 left frames and 5 right frames was used. The LSTM and GRU models were built with 2 hidden layers and 64 hidden nodes per layer. For the GRU KWS model, the final GRU layer was followed by a fully connected layer with ReLU non-linearity. There were no stacked frames in the input for the LSTM and GRU models. The smoothing window for Deep KWS was set to 20 frames. We also trained a TDNN-based acoustic model [22] using ∼3000 hours of speech data to perform frame-level alignment before KWS model training.

Table 3: *Performance of different encoder architectures with soft attention. FRR is at 1.0 false alarm (FA) per hour.*

Recurrent Unit	Layer	Node	FRR (%)	Params (K)
LSTM	1	64	4.36	31.2
LSTM	2	64	3.58	64.3
LSTM	3	64	3.05	97.3
LSTM	1	128	2.99	103
GRU	1	64	3.22	28.7
GRU	2	64	1.93	53.4
GRU	3	64	1.99	78.2
GRU	1	128	**1.49**	77.5

Table 4: *Performance of adding convolutional layers in the GRU (CRNN) attention-based model with soft attention. FRR is at 1.0 false alarm (FA) per hour.*

Channel	Layer	Node	FRR (%)	Params (K)
8	1	64	2.48	52.5
8	2	64	1.34	77.3
16	1	64	**1.02**	84.1
16	2	64	1.29	109

3.3. Experimental setup

In the neural network models, all the weight matrices were initialized with the normalized initialization [23] and the bias vectors were initialized to 0. We used ADAM [24] as the optimization method while we decayed the learning rate from 1e-3 to 1e-4 after it converged. Gradient norm clipping to 1 was applied, together with L2 weight decay 1e-5. The positive training sample has a frame length of $T = 1.9$ seconds which ensures the entire wake-up word is included. Accordingly, in the attention models, the input window has set to 189 frames to cover the length of the wake-up word. We randomly selected 189 contiguous frames from the negative example set to train the attention models. At runtime, the sliding window was set to 100 frames and frame shift was set to 1. Performances were measured by observing the FRR at the operating threshold of 1.0 FA per hour, while plotting a receiver operating curve (ROC).

3.4. Impact of attention mechanism

From Table 2 and Fig. 3, we can clearly see the superior performances of the attention models. With similar size of parameters, the proposed attention models outperform the Deep KWS systems by a large margin. We also note that GRU is preferred over LSTM in both Deep KWS and the attention models. Not surprisingly, the soft attention-based model achieves the best performance. At 1.0 FA/hour, the GRU attention model reduces the FRR from 6.38% (GRU Deep KWS) down to 1.93% with a remarkable false rejection reduction.

3.5. Impact of encoder architecture

We further explored the impact of encoder architectures with soft attention. Results are summarized in Table 3, Fig. 4 and Fig. 5. From Table 3, we notice that the bigger models always perform better than the smaller models. Observing the LSTM models, the 1-128 LSTM model achieves the best performance with an FRR of 2.99% at 1.0 FA/hour. In Fig. 4, the ROC curves of the 1-128 LSTM model and the 3-64 LSTM model are overlapped at lower FA per hour. This means making the LSTM network wider or deeper can achieve the same effect. However, observing Fig. 5, the same conclusion does not hold for GRU. The 1-128 GRU model presents a significant advantage over

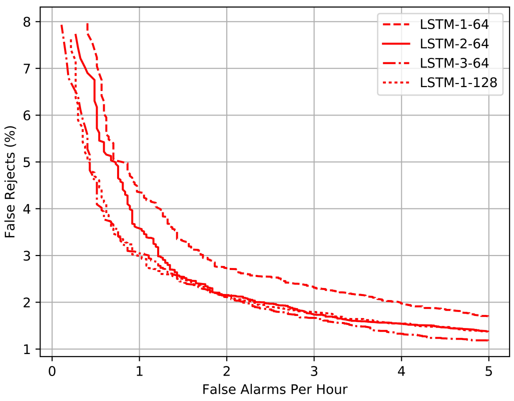

Figure 4: *ROCs for LSTM Attention-based model with soft attention.*

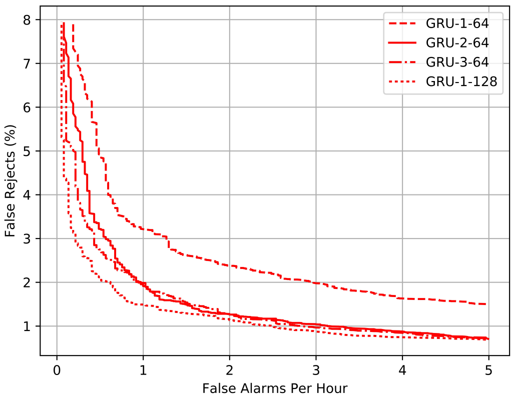

Figure 5: *ROCs for GRU Attention-based model with soft attention.*

the 3-64 GRU model. In other words, increasing the number of nodes may be more effective than increasing the number of layers. Finally, the 1-128 GRU model achieves 1.49% FRR at 1.0 FA/hour.

3.6. Adding convolutional layer

Inspired by [18], finally we studied the impact of adding convolutional layers in the GRU attention-model as convolutional network is often used as a way to extract invariant features. For the CRNN attention-based model, we used one layer CNN that has a $\frac{C(20\times5)}{1\times2}$ filter. We explored different numbers of output channel and results are summarized in Table 4 and Fig. 6. From Table 4, we can see that adding convolutional layer can further improve the performance. We achieve the lowest FRR of 1.02% at 1.0 FA/hour with 84.1K parameters. Another observation is that 16-channel models work better than 8-channel models. By increasing layers, the 8-2-64 model achieves a great gain over the 8-1-64 model. But we cannot observe extra benefit when increasing the layers with 16-channel models.

As a summary, Fig. 7 plots the ROC curves for the best three systems. We can see that GRU and CRNN outperform LSTM by a large margin and the best performance is achieved by CRNN.

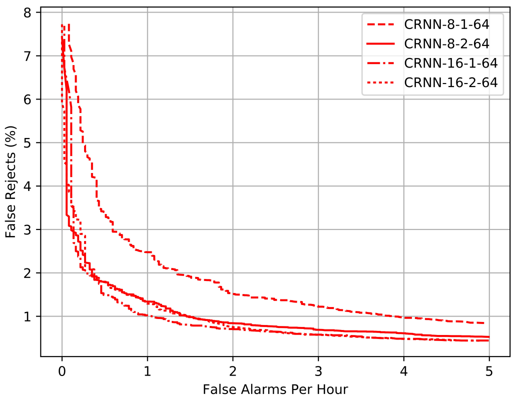

Figure 6: *ROCs for CRNN Attention-based model with soft attention.*

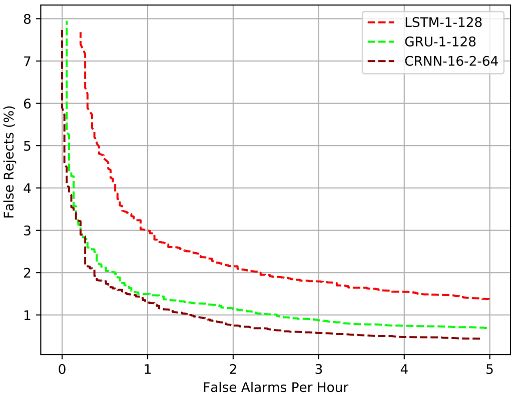

Figure 7: *ROCs for different architectures with soft Attention-based model.*

4. Conclusions

In this paper, we propose an attention-based end-to-end model for small-footprint keyword spotting. Compared with the Deep KWS system, the attention-based system achieves superior performance. Our system consists of two main sub-modules: the encoder and the attention mechanism. We explore the encoder architectures, including LSTM, GRU and CRNN. Experiments show that GRU is preferred over LSTM and the best performance is achieved by CRNN. We also explore two attention mechanisms: average attention and soft attention. Our results show that the soft attention has a better performance than the average attention. With ∼84K parameters, our end-to-end system finally achieves 1.02% FRR at 1.0 FA/hour.

5. Acknowledgements

The authors would like to thank Jingyong Hou for helpful comments and suggestions. The research work is supported by the National Natural Science Foundation of China (Grant No. 61571363).

6. References

[1] P. Motlicek, F. Valente, and I. Szoke, "Improving acoustic based keyword spotting using lvcsr lattices," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4413–4416.

[2] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2338–2347, 2011.

[3] Y. Bai, J. Yi, H. Ni, Z. Wen, B. Liu, Y. Li, and J. Tao, "End-to-end keywords spotting based on connectionist temporal classification for mandarin," in *Chinese Spoken Language Processing (ISCSLP), 2016 10th International Symposium on*. IEEE, 2016, pp. 1–5.

[4] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 5280–5284.

[5] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 627–630.

[6] I. Szke, P. Schwarz, P. Matejka, L. Burget, M. Karafit, M. Fapso, and J. Cernock, "Comparison of keyword spotting approaches for informal continuous speech," in *INTERSPEECH 2005 - Eurospeech, European Conference on Speech Communication and Technology, Lisbon, Portugal, September*, 2005, pp. 633–636.

[7] K. Hwang, M. Lee, and W. Sung, "Online keyword spotting with a character-level recurrent neural network," *arXiv preprint arXiv:1512.08903*, 2015.

[8] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," *arXiv preprint arXiv:1710.09617*, 2017.

[9] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Acoustics, speech and signal processing (ICASSP), 2014 ieee international conference on*. IEEE, 2014, pp. 4087–4091.

[10] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 474–480.

[12] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[13] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[14] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end speech recognition on voice search," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[16] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[17] F. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," *arXiv preprint arXiv:1710.10470*, 2017.

[18] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," *arXiv preprint arXiv:1703.05390*, 2017.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[21] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5670–5674.

[22] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.