MINING EFFECTIVE NEGATIVE TRAINING SAMPLES FOR KEYWORD SPOTTING

Jingyong Hou[1], *Yangyang Shi*[2], *Mari Ostendorf*[3], *Mei-Yuh Hwang*[2], *Lei Xie*[1]

[1]School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China
[2]Mobvoi AI Lab, Redmond, USA
[3]Department of Electrical & Computer Engineering, University of Washington, Seattle, USA
{jyhou,lxie}@nwpu-aslp.org, ostendor@uw.edu, {yyshi,mhwang}@mobvoi.com

ABSTRACT

Max-pooling neural network architectures have been proven to be useful for keyword spotting (KWS), but standard training methods suffer from a class-imbalance problem when using all frames from negative utterances. To address the problem, we propose an innovative algorithm, Regional Hard-Example (RHE) mining, to find effective negative training samples, in order to control the ratio of negative vs. positive data. To maintain the diversity of the negative samples, multiple non-contiguous difficult frames per negative training utterance are dynamically selected during training, based on the model statistics at each training epoch. Further, to improve model learning, we introduce a weakly constrained max-pooling method for positive training utterances, which constrains max-pooling over the keyword ending frames only at early stages of training. Finally, data augmentation is combined to bring further improvement. We assess the algorithms by conducting experiments on wake-up word detection tasks with two different neural network architectures. The experiments consistently show that the proposed methods provide significant improvements compared to a strong baseline. At a false alarm rate of once per hour, our methods achieve 45-58% relative reduction in false rejection rates over a strong baseline.

Index Terms— Keyword spotting, Wake-up word detection, Class imbalance, End-to-end, Hard examples

1. INTRODUCTION

Small-footprint Keyword Spotting (KWS) systems are widely used in IoT devices such as smart speakers and mobile phones for wake-up word detection. On these devices, the KWS system needs to process streaming audio in real-time, locally on the device, to detect some predefined keyword(s). An accurate, fast, and small-footprint KWS system is highly desired to reduce power needs of computation. We classify recent popular KWS architectures into two categories: keyword/filler posterior modeling followed by a search algorithm, and end-to-end (E2E) based architectures.

In the first approach [1, 2, 3, 4, 5], each word (or subword) of the keyword (can be multiple words) is modeled by a Hidden Markov Model (HMM) and usually an additional phone-loop graph is used as a filler model to absorb non-keyword speech segments. Given the posterior probabilities of the (sub)word units, a simple search algorithm is followed to find the occurrence of the keyword phrase, similar to speech recognition. With the success of Deep Neural Networks (DNNs) [6], some recent work has replaced HMMs with pure DNNs [7, 8, 9, 10, 11] or simplified HMM-DNN hybrid models [12, 13, 14, 15, 16]. In these newer approaches, an output node to represent the posterior of filler segments is often used to replace the phone-loop HMM graph.

E2E architectures bring further improvement to small-footprint KWS. They treat a keyword as a single modeling unit and simply detect its presence in the streaming utterance. As each frame arrives, the model decides if a keyword has been discovered. In this case, KWS becomes a keyword/non-keyword binary classification task. The sequence binary classification model is trained to minimize the keyword category cross entropy loss [17, 18, 19, 20].

As Sun *et al.* [21] points out, cross-entropy based training relies on accurate time labeling of the keyword. To alleviate the dependency, they proposed a max-pooling based cross-entropy loss function: for each positive keyword utterance, the keyword category is updated based on the single frame with the highest positive-class posterior within the keyword location. For the non-keyword category, all the frames in non-keyword regions are used, including the non-keyword segments in positive utterances. This causes a severe class imbalance problem; i.e., the ratio of non-keyword vs. keyword training samples is unreasonably large.

Class imbalance is common in small-footprint KWS training, because it is expensive to collect positive keyword training data, while it is easy to find abundant non-keyword data. On the other hand, we do need a large amount of diverse negative training data to prevent false alarms, especially due to phrases similar to the keyword or due to various environment noises. The class imbalance problem in deep KWS systems [7] has been addressed by Liu *et al*. [11] using focal loss.

In this paper, we focus on improving max-pooling based E2E KWS. To alleviate the class imbalance problem during training, we propose a regional hard-example (RHE) mining algorithm to select representative negative training samples. The idea is inspired by the Online Hard Example Mining algorithm in object detection [22]. Our proposed method includes a few innovations. First, we select effective negative examples dynamically during training, at the same time maintaining a controlled ratio of positive vs. negative training samples within each mini-batch. Second, to address inaccurate time labeling of the keyword associated with automatic force-alignment by existing acoustic models, we use weakly constrained max-pooling, where the restriction of max-pooling over keyword areas is enforced only at early stages of training. In addition, to alleviate over-fitting in training, SpecAugment [23] is applied, which has been proven useful in automatic speech recognition.

To verify our proposals, we conduct experiments using both Gated Recurrent Unit (GRU) [24] and dilated Temporal Convolutional Network (TCN) [25] structures. At a false alarm rate (FAR) of once per hour, our method achieves 45-58% relative reduction in the false rejection rate (FRR) on two different keywords, over a strong baseline system. Our code will be made public after peer review.[1]

[1]github.com/jingyonghou/KWS_Max-pooling_RHE.git

2. METHODS

2.1. KWS with end-to-end solutions

In this section, we define the wake-up word detection task in our E2E detection framework. We use one keyword as an example; it can be easily extended to detect multiple keywords. Suppose we have a predefined keyword or keyphrase α. For each time frame t, we denote its feature vector as x_t. The wake-up word detector Q assigns a score y_t for each x_t. As soon as $y_t > \gamma$, we say keyword α has occurred. $\gamma \in (0, 1)$ is a threshold tuned on a development dataset.

To model the acoustic sequence for keyword spotting, recurrent neural networks (RNNs) and TCNs are two common choices for E2E modeling Q [18, 19, 20, 21, 26]. An RNN models long contextual information by its memory mechanism and recurrent connections, while a TCN models long contextual information through the stacked temporal convolutions with dilate connections. On top of the RNN (here, a GRU) or TCN, a linear layer with a sigmoid activation is applied to do the binary classification.

2.2. Loss function with cross entropy

E2E KWS is a sequence binary classification problem. The cross-entropy (CE) loss for binary classification is formulated as follows, for each mini-batch of size M:

$$\begin{aligned} \text{Loss(CE)} &= \frac{1}{M}\sum_{i=1}^{M} \text{CE}(y_i, y_i^*) \\ &= \frac{1}{M}\sum_{i=1}^{M} [-y_i^* \ln y_i - (1 - y_i^*) \ln(1 - y_i)] \end{aligned} \tag{1}$$

where $y_i^* \in \{0, 1\}$ is the ground-truth class label for frame i, $y_i = Q(x_i; \theta) \in (0, 1)$ is the posterior probability of the keyword category estimated by the model Q with parameter θ.

2.3. Baseline max-pooling

Max-pooling based loss is first proposed by Sun *et al.* [21] for training RNN-based E2E KWS. Assume each positive training utterance contains one single occurrence of the keyword and its beginning and ending timestamps are denoted as (t_b, t_e). For each positive utterance, constrained max-pooling selects the single frame with the highest positive posterior within (t_b, t_e) as a positive training example. The frames outside the keyword segment in the positive utterance and all frames from each negative utterance are treated as negative training examples. Within a mini-batch, let P denote the total number of positive training frames and N the total number of negative training frames, then $M = N + P$. It is easy to see that this data labeling often results in $N \gg P$, i.e., severe data imbalance.

The baseline max-pooling loss we conduct in this paper is slightly different from [21] in the following ways. We use a single output node with a sigmoid activation instead of two output nodes with a softmax to get the posterior probability. We do max-pooling over the ending area of each keyword, as in [18], instead of within the keyword. We also discard the rest of data in the positive training utterance, rather than use it as negative data. To reduce false triggering of similar frames matching the initial segment of the keyword, [21] stacked the current frame with left and right neighboring frames. Patching the input feature with future frames can cause latency at run-time; this is not required by our proposed method.

In this paper, we call the keyword ending segment of $(t_e \pm \delta)$ the trigger region or TR for short, and keep $\delta = 30$ as a constant.

2.4. Proposed max-pooling

Different from [18], we do not use all data from negative utterances for back-propagation. Instead we strategically down-sample negative frames to keep data in check between the two classes. Moreover, constrained max-pooling is used only at early stages of training.

2.4.1. Mining regional hard examples (RHE) in negative utterances

To alleviate the class-imbalance issue with max-pooling, we propose a simple algorithm to down-sample negative frames, choosing difficult time samples from negative utterances, as detailed in Algorithm 1. For each negative utterance in a mini-batch, we select the most difficult frame with the top positive posterior probability computed by the current model. This frame is put into a collection $\mathbf{I}$. Then, we mask Δ neighboring frames (both left and right neighbors) of the selected hardest frame. These masked frames are not selected, as they are assumed to be acoustically similar to the selected frame. We continue the RHE mining based on the remaining frames until no more negative frames are left. After processing all the negative utterances in a mini-batch, we rank all negative frames in $\mathbf{I}$ by their posterior probabilities and select the top rP frames for training the negative class, thereby keeping the data ratio between these two classes to be under r.

Algorithm 1 Mining regional hard examples in a negative utterance

Input: $\mathbf{y} = (y_1, y_2, ..., y_T)$: Given a negative utterance of T frames, y_i is the positive posterior probability of frame i computed by the current model. A region parameter Δ is pre-defined to indicate the neighborhood region for frame i: $(i - \Delta, i + \Delta)$.
Output: $\mathbf{I}$: A collection of selected negative frames.

Sort $\mathbf{y}$ descendingly according to the posteriors, yielding $\mathbf{s} = (s_1, s_2, ..., s_T)$, the frame indices after sorting. y_{s_i} corresponds to i-th largest posterior in $\mathbf{y}$.
Denote the availability of the T frames with a binary array: $\mathbf{a} = (a_1, a_2, ..., a_T)$. $a_i = 1$ means frame i in the original input is available for selection. $a_i = 1\ \forall i$ initially.
for $(i = 1; i \leq T; i++)$ **do**
  if sum$(\mathbf{a}) == 0$ **then**
    break
  end if
  if $a_{s_i} == 1$ **then**
    push(I, frame s_i)
    $t_1 = \max(s_i - \Delta, 1)$
    $t_2 = \min(s_i + \Delta, T)$
    $\mathbf{a}[t_1 : t2] = 0$
  end if
end for

2.4.2. Weakly constrained max-pooling for positive utterances

In max-pooling based training, the frame, within TR, that gets the highest positive posterior probability, is used for training. However, TR usually comes from automatic force-alignment by existing acoustic models, which may not be accurate. To alleviate the inaccurate TR/force-alignment problem, we propose a simple strategy which selects the positive frame in TR *only* at early stages of network training (we do so in the first two epochs in our experiments).

Table 1. Corpus statistics (#speakers/#utterances)

Data set	Train (60%)	Dev (10%)	Test (30%)
Hi Xiaowen	474/ 21,825	78/ 3,680	236/10,641
Nihao Wenwen	474/ 21,800	78/ 3,677	236/10,641
Non-keyword	418/113,898	67/17,522	203/51,613
All	474/157,523	78/24,879	236/72,895

This enables the network converge faster and makes training more stable. In later epochs, we relax the TR constraint to select the single frame from any frame in the positive utterance since the model now is better trained. We call the early-epoch TR constraint, the weak constraint.

2.5. Data Augmentation

SpecAugment [23] strategy was first proposed for E2E speech recognition and achieved a great success. We apply time masking and frequency masking in this paper. For each training utterance, we randomly select $0-50$ consecutive frames and set all of their mel-filter banks to zero, for time masking. For frequency masking, we randomly select $0-30$ consecutive dimensions of the 40 mel-filter banks and set their values to zero for all frames of the utterance. For all the utterances in a training mini-batch, one-third of them receive only the time masking, one-third of them only the frequency masking, and the rest of them both maskings. Though this method deforms instead of doubling the original data set, it is considered augmentation as each mini-batch at different epoches is deformed differently. The training observes data with a lot more varieties than the original data amount. Hence it is considered a data-augmentation method.

3. EXPERIMENTS

3.1. Corpus

A wake-up word detection corpus collected from a commercial smart speaker is used to verify our algorithm. The dataset is identical to the corpus in [20], where the corpus consists of two keywords: "Hi Xiaowen" and "Nihao Wenwen". All speakers are recorded saying both keywords, and the keyword lengths range from 30 to 200 frames. Here we train separate models for each keyword, different from [20], which treated this as a multi-class classification problem. When we train a model for one keyword, the other keyword's utterances are used as negative training data. Detailed corpus statistics can be found in Table 1. 40-dimensional mel-filter banks features with 25ms frame length and 10ms frame shift are extracted as input features for model training. To obtain the TR region for the positive utterances, keyword timestamps are automatically generated by force alignment using Kaldi [27] HMM-TDNN (time delay neural network) acoustic model trained on general Mandarin speech data.

3.2. Setups

3.2.1. Neural network architecture

Two different neural network architectures are used to verify our proposed method. One is GRU and the other one is dilated TCN.

For GRU, 2 layers of unidirectional GRU and a projection layer with ReLU activation are used. Each GRU layer has 128 cells. The projection layer also has 128 output nodes.

For TCN, 1 preprocessing 1×1 1-d causal convolution layer and 8 dilated causal convolution (with a filter size of 8) layers are used.

Table 2. Systems with different data strategies

Methods	Positive utterances $\delta=30$	Negative utterances $\Delta=200$	Data ratio r
B1	All TR frames	All	35
B2	Max-pooling in TR	All	2114
B3	Max-pooling in TR	Random	200
S1	Max-pooling in TR	RHE	10
S2	Weak constraint	RHE	10
S2+SpecA	Weak constraint	RHE	10

The 8 dilated rates are $\{1,2,4,8,1,2,4,8\}$, resulting in a receptive field of 210 frames. For each layer, ReLU activation is used, the number of filters is 64.

We choose a mini-batch size of M=400. For all systems, we use the warm-up strategy in the first 200 mini-batches by starting from a small learning rate and gradually increasing the learning rate to a predefined maximum value, which is tuned individually for each system, ranging from 0.0005 to 0.01. Adam optimization is used throughout the paper. After each epoch, we evaluate the loss on the validation set. If there is no reduction in loss, the learning rate begins to decay, by a factor of 0.7. Each model is trained for at least 15 epochs. After that, if there is no decrease in the loss on the validation set, we terminate the training.

3.2.2. Baseline systems

As listed in Table 2, three different baseline methods are implemented in this paper. All systems are trained independently with both GRU and TCN architectures.

B1 mainly follows [18]. It uses all TR frames in positive utterances, and all frames in negative utterances, to train a binary classifier. The ratio of negative training data vs. positive training data for "Hi Xiaowen" is 35. For "Nihao Wenwen", it is roughly the same ratio.

B2 is the max-pooling based method proposed by [21], with modifications described in Sec. 2.3. The data ratio is 2114, determined by the roughly 60-frame duration of the TR (due to $\delta=30$).

B3 is also a max-pooling based method. Different from B2, we do not use all the frames in negative utterances. Instead, we randomly down-sample negative training data, setting r to be 200 in each mini-batch.

For all systems we tuned the learning rates to achieve the best for each system. For B1 method, learning rate of 0.005 is chosen to train both GRU and TCN model. For B2, learning rates of 0.003 and 0.0005 are chosen to train GRU and TCN, respectively. For B3, learning rates of 0.005 and 0.0005 are chosen to train GRU and TCN, respectively.

3.2.3. Proposed systems

The bottom three rows in Table 2 apply our proposed algorithms. For S1, instead of using all the frames in negative utterances, it uses the proposed negative RHE mining algorithm to select non-keyword training frames. S2 applies weakly constrained max-pooling described in Sec. 2.4.2 on top of S1. That is, TR is used by positive utterances only at the first two epoches of training. S2+SpecA applies SpecAugment based on S2. $\Delta=200$ and $r=10$ were tuned on the S1 GRU "Hi Xiaowen" system, and then are applied directly to the rest of experiments without further tuning. Optimal learning rates for GRU vs. TCN is 0.01 and 0.006 respectively.

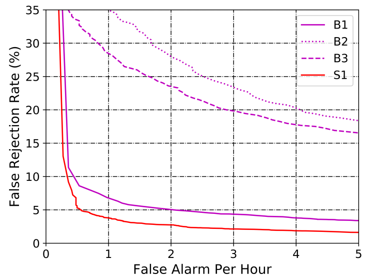

Fig. 1. DET curves on "Hi Xiaowen" with GRU.

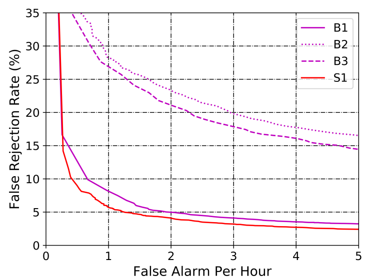

Fig. 2. DET curves on "Hi Xiaowen" with TCN.

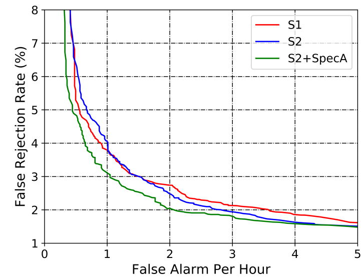

Fig. 3. DET curves on "Nihao Xiaowen" with GRU.

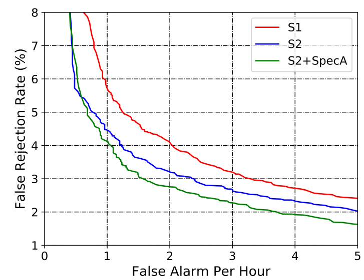

Fig. 4. DET curves on "Nihao Xiaowen" with TCN.

3.3. Results

3.3.1. Effect of negative RHE mining

In Fig. 1 and Fig. 2, we analyze the effect of negative data mining on "Hi Xiaowen" KWS. Comparing the Detection Error Trade-off (DET) curves of all baseline systems (B1, B2, B3), it shows that max-pooling for the positive utterances degrades the performance significantly. Only when max-pooling is combined with our proposed negative data down-sampling that we see significant improvement over B1 and B2.

In order to analyze whether the improvement from B2 to S1 is completely due to data imbalance, we tried a few variations of B3, which randomly sample the negative examples to control the data ratio. We tried data ratio of 200, 100, 40, and 10. Among those, 200 gave us the best performance, shown in Fig. 1 and Fig. 2. Although adjusting the data ratio yields some improvement, B3 is still much worse than B1, not to mention S1. This means that it is crucial to sub-sample negative frames smartly. Specifically, when FAR is fixed at once per hour, S1 achieves 46% and 23% relative FRR decreases with GRU and TCN respectively, compared with B1.

The hyper-parameters $\Delta = 200$ and $r = 10$ are tuned based on S1 GRU "Hi Xiaowen" system, and then frozen without further tuning for the rest of experiments. It shows that these hyper-parameters are robust, at least in our data sets.

3.3.2. Weakly constrained max-pooling and SpecAugment

Based on S1, we further validate the effect of weakly constrained max-pooling (for positive utterances) and SpecAugment. The results are shown in Fig. 3 and 4. As illustrated in the DET curves, we find that weakly constrained max-pooling (S2) has more impact on TCN than GRU. We conjecture that TCN training is more sensitive to accurate alignment. When weakly constrained max-pooling and SpecAugment are combined, both GRU and TCN models are better than S1. When FAR is fixed at once per hour, S2+SpecA obtains 18% and 28% relative FRR decreases compared with S1 on GRU and TCN respectively.

3.3.3. More comparisons on the second keyword

Finally, we verify our algorithms on the second keyword ("Nihao Wenwen") with the same optimal hyper parameters, by comparing the best baseline, B1, and our best system, S2+SpecA. The results in Table 3 confirm the consistent significant improvements in different configurations, with FRR reductions of 45-58%.

4. SUMMARY

We propose a smart negative data mining algorithm, RHE, to dynamically select non-keyword training frames in negative utterances. The proposed algorithm is able to deal with the class-imbalance issue in keyword spotting tasks. We also propose a weakly-constrained max-pooling strategy that restricts the max-pooling region only at early stages of training. We verified the effectiveness of our proposals on two commercial wake-up keywords, using two different neural network architectures. Combining with SpecAugment, our proposed method is 45-58% better than our strongest baseline system.

Table 3. Comparison of B1 and S2+SpecA with different neural network architectures and keywords with FAR fixed at one per hour. Results are given with the percent reduction in FRR%.

Networks / Keywords	GRU	TCN
Hi Xiaowen	7.1/3.1 (56%)	7.4/4.1 (45%)
Nihao Wenwen	6.4/2.7 (58%)	7.3/3.5 (52%)

5. REFERENCES

[1] J Robin Rohlicek, William Russell, Salim Roukos, and Herbert Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. ICASSP*, 1989, pp. 627–630.

[2] Richard C Rose and Douglas B Paul, "A hidden Markov model based keyword recognition system," in *Proc. ICASSP*, 1990, pp. 129–132.

[3] JG Wilpon, LG Miller, and P Modi, "Improvements and applications for keyword recognition using hidden Markov modeling techniques," in *Proc. ICASSP*, 1991, pp. 309–312.

[4] Marius-Calin Silaghi and Hervé Bourlard, "Iterative posterior-based keyword spotting without filler models," in *Proc. ASRU*, 1999, pp. 213–216.

[5] Marius-Calin Silaghi, "Spotting subsequences matching an HMM using the average observation probability criteria with application to keyword spotting," in *Proc. AAAI*, 2005, pp. 1118–1123.

[6] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP*, 2014, pp. 4087–4091.

[8] Rohit Prabhavalkar, Raziel Alvarez, Carolina Parada, Preetum Nakkiran, and Tara N Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *Proc. ICASSP*, 2015, pp. 4704–4708.

[9] Tara N Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2015, pp. 1106–1610.

[10] Jinyu Li, Rui Zhao, Zhuo Chen, Changliang Liu, Xiong Xiao, Guoli Ye, and Yifan Gong, "Developing far-field speaker system via teacher-student learning," in *Proc. ICASSP*, 2018, pp. 5699–5703.

[11] Bin Liu, Shuai Nie, Yaping Zhang, Shan Liang, Zhanlei Yang, and Wenju Liu, "Loss and double-edge-triggered detector for robust small-footprint keyword spotting," in *Proc. ICASSP*, 2019, pp. 6361–6365.

[12] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Björn Hoffmeister, and Shiv Vitaladevuni, "Multi-task learning and weighted cross-entropy for DNN-based keyword spotting.," in *Proc. INTERSPEECH*, 2016, pp. 760–764.

[13] Ming Sun, David Snyder, Yixin Gao, Varun Nagaraja, Mike Rodehorst, Nikko Strom Panchapagesan, Spyros Matsoukas, and Shiv Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2017, pp. 3607–3611.

[14] Kenichi Kumatani, Sankaran Panchapagesan, Minhua Wu, Minjae Kim, Nikko Strom, Gautam Tiwari, and Arindam Mandai, "Direct modeling of raw audio with DNNs for wake word detection," in *Proc. ASRU*, 2017, pp. 252–257.

[15] Jinxi Guo, Kenichi Kumatani, Ming Sun, Minhua Wu, Anirudh Raju, Nikko Ström, and Arindam Mandal, "Time-delayed bottleneck highway networks using a DFT feature for keyword spotting," in *Proc. ICASSP*, 2018, pp. 5489–5493.

[16] Minhua Wu, Sankaran Panchapagesan, Ming Sun, Jiacheng Gu, Ryan Thomas, Shiv Naga Prasad Vitaladevuni, Bjorn Hoffmeister, and Arindam Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *Proc. ICASSP*, 2018, pp. 5494–5498.

[17] R Alvarez and HJ Park, "End-to-end streaming keyword spotting," in *Proc. ICASSP*, 2019, pp. 6336–6340.

[18] Alice Coucke, Mohammed Chlieh, Thibault Gisselbrecht, David Leroy, Mathieu Poumeyrol, and Thibaut Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *Proc. ICASSP*, 2019, pp. 6351–6355.

[19] Haitong Zhang, Junbo Zhang, and Yujun Wang, "Sequence-to-sequence models for small-footprint keyword spotting," *arXiv preprint arXiv:1811.00348*, 2018.

[20] Jingyong Hou, Yangyang Shi, Mari Ostendorf, Mei-Yuh Hwang, and Lei Xie, "Region proposal network based small-footprint keyword spotting," *IEEE Signal Processing Letters*, vol. 26, no. 10, pp. 1471–1475, 2019.

[21] Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyros Matsoukas, Nikko Strom, and Shiv Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Proc. SLT*, 2016, pp. 474–480.

[22] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick, "Training region-based object detectors with online hard example mining," in *Proc. CVPR*, 2016, pp. 761–769.

[23] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[24] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc EMNLP*, 2014, pp. 1724–1734.

[25] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[26] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2018, pp. 2037–2041.

[27] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.