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

Large Margin Training for Attention Based End-to-End Speech Recognition

Peidong Wang[1*] *, Jia Cui*[2]*, Chao Weng*[2]*, Dong Yu*[2]

[1]The Ohio State University, Columbus, USA
[2]Tencent AI Lab, Bellevue, USA

wang.7642@osu.edu, {jiaacui, cweng, dyu}@tencent.com

Abstract

End-to-end speech recognition systems are typically evaluated using the maximum a posterior criterion. Since only one hypothesis is involved during evaluation, the ideal number of hypotheses for training should also be one. In this study, we propose a large margin training scheme for attention based end-to-end speech recognition. Using only one training hypothesis, the large margin training strategy achieves the same performance as the minimum word error rate criterion using four hypotheses. The theoretical derivation in this study is widely applicable to other sequence discriminative criteria such as maximum mutual information. In addition, this paper provides a more succinct formulation of the large margin concept, paving the road towards a better combination of support vector machine and deep neural network.

Index Terms: large margin, attention based end-to-end speech recognition, MBR training, MAP evaluation, Switchboard

1. Introduction

End-to-end speech recognition systems have been actively investigated in the past few years [1, 2, 3]. An attention based end-to-end system maps from input audio features to text sequences in three steps, encoding, attention, and decoding [4, 5, 6, 7]. The most commonly used decoder is a recurrent network trained with point-wise cross entropy loss. Recently, sequence discriminative optimization criteria such as the minimum Bayes risk (MBR) based minimum word error rate (MWER) [8, 9, 10] were applied to boost model performances. MWER requires multiple hypotheses during training. The commonly used maximum a posteriori (MAP) evaluation [11], however, reports results only on the 1-best hypothesis. This mismatch indicates that the efficiency of existing training schemes can be improved by reducing the number of hypotheses used for training. In this study, we propose a large margin training scheme to achieve this goal.

Formulating speech recognition as a sequence-to-sequence mapping from audio features directly to output tokens, end-to-end systems enable explicit interactions between acoustic and language models. Popular paradigms of end-to-end speech recognition systems include connectionist temporal classification (CTC) and attention. The attention mechanism was first introduced to speech recognition by Chorowski *et al.* [1]. Chan *et al.* proposed the listen, attend, and spell (LAS) system, which conducts speech recognition by encoding the audio features with a "listener" and generating the decoded tokens with a "speller" [2]. Chiu *et al.* improved the LAS system with various modifications [8]. One of the main improvements is to apply the MBR based MWER training scheme. Experimental results showed that the modified LAS system outperformed conventional DNN-HMM based systems on the Google voice search and dictation tasks. Prabhavalkar *et al.* first proposed MWER training for LAS [9]. Weng *et al.* [10] made further improvements and obtained competitive results on the benchmark Switchboard 300h and 2000h corpora. Recently, an investigation on the lexicon modeling ability of sequence discriminative training criteria was conducted by Cui *et al.* [12]. In addition to sequence discriminative training schemes, a policy gradient based training method and a fine-grained error for attention based end-to-end systems were proposed by Karita *et al.* [13].

The large margin concept is typically fused with support vector machines (SVMs) [14]. By enlarging the margin between the reference sequence and the incorrect sequences, the upper bound of the generalization errors may be minimized [14]. In the past few years, structural SVMs (SSVMs) have been combined with various deep neural networks (DNNs) for speech recognition tasks [15, 16, 17]. In these models, softmax layers are replaced with SSVM layers and the training process consists of two stages. In the first stage, the weights of the SSVM layer are calculated using the cutting-plane algorithm [18] on all of the training samples. The parameters in the DNNs are then updated with the back-propagation algorithm. Sequence-level implementations of the deep neural support vector machine (DNSVM) have shown superior performances to those of the corresponding sequence discriminatively trained DNNs on tasks such as the Windows Phone short message dictation [17].

In this study, we propose a large margin training criterion for attention based end-to-end speech recognition. Using only one hypothesis during training, it yields the same performances as the MWER training using four hypotheses. We present a detailed derivation for the gradient calculation of large margin training. The derivation is widely applicable to other sequence discriminative training criteria such as maximum mutual information (MMI). Another contribution of this study is that it provides a new formulation of the large margin concept, paving the road towards a better combination of SVM and DNN.

The rest of this paper is organized as follows. In Section 2, we describe the large margin training scheme for attention based end-to-end speech recognition. Section 3 and 4 contain the experimental setup and results on the SWBD 300h corpus. Finally, we conclude this paper in Section 5.

2. System Description

2.1. MBR Training and MAP Evaluation

MBR based training criteria such as MWER aim at minimizing the empirical risk of output hypotheses:

$$L(\theta) = \sum_{(\boldsymbol{\chi},s)\in D} \sum_{s'} l(s',s) p_\theta(s'|\boldsymbol{\chi}) \tag{1}$$

*work performed during an internship at Tencent AI Lab

http://dx.doi.org/10.21437/Interspeech.2019-1680

where $(\boldsymbol{\chi}, s)$ is a sample in the training set D. $\boldsymbol{\chi}$ denotes the input feature, s its corresponding sequence label, s' an output hypothesis generated during training, and θ the parameters. The difference between s and s' is denoted as $l(s', s)$, which is typically chosen to be the word or character level edit distance.

Correspondingly, the output sequence during evaluation should be generated as in equation (2) below:

$$\hat{s}_{MBR} = \operatorname*{argmin}_{\hat{s}} \sum_{s'} l(s', \hat{s}) p_\theta(s'|\boldsymbol{\chi}) \tag{2}$$

where $\hat{s}$ denotes the candidate output sequence and $\hat{s}_{MBR}$ the one chosen by MBR decoding.

The search space of $\hat{s}$ grows exponentially with its length, making MBR decoding mechanisms such as recognition output voting error reduction (ROVER) inefficient [19]. Although n-best list or confusion network based methods can improve the efficiency [20, 21, 22, 23, 24, 25], beam search decoding based on MAP is still one of the most commonly used evaluation methods in practice. In MAP decoding, the output hypothesis with the highest (log) posterior is used directly for evaluation:

$$\hat{s}_{MAP} = \operatorname*{argmax}_{\hat{s}} p_\theta(\hat{s}|\boldsymbol{\chi}) \tag{3}$$

The mismatch between MBR training and MAP decoding suggests that there may be a training scheme which has better efficiency than and comparable performance to MBR training.

2.2. Large Margin Training for Attention Based End-to-End Speech Recognition

Large margin training in this study refers specifically to the sequence level training criterion enlarging the margin between reference sequence and the most probable incorrect sequence. Expression (4) shows the margin of the reference sequence s to the most probable incorrect sequence $\hat{s}$:

$$\min_{\hat{s} \neq s} \log \frac{p_\theta(s|\boldsymbol{\chi})}{p_\theta(\hat{s}|\boldsymbol{\chi})} \tag{4}$$

Enlarging the margin in expression (4) during training is essentially reducing the loss function in equation (5):

$$L(\theta) = \sum_{(\boldsymbol{\chi}, s) \in D} \max_{\hat{s} \neq s} \log \frac{p_\theta(\hat{s}|\boldsymbol{\chi})}{p_\theta(s|\boldsymbol{\chi})} \tag{5}$$

Equation (5) is typically written in the form of equation (6) below:

$$L(\theta) = \sum_{(\boldsymbol{\chi}, s) \in D} \max_{\hat{s} \neq s} \{\log p_\theta(\hat{s}|\boldsymbol{\chi}) - \log p_\theta(s|\boldsymbol{\chi})\} \tag{6}$$

Similar to Zhang *et al.*'s prior work [16, 17], a threshold $l(\hat{s}, s)$ is introduced to control the desired distance between the reference sequence and the most probable incorrect sequence. To filter out the samples that have already satisfied the constraint of the threshold, we apply a rectifier $[\cdot]_+$. The loss function is in the squared form so that it can impact not only the sign but also the value of its gradients. The large margin loss is thus refined as equation (7) below:

$$L(\theta) = \sum_{(\boldsymbol{\chi}, s) \in D} [\max_{\hat{s} \neq s} \{l(\hat{s}, s) - (\log p_\theta(s|\boldsymbol{\chi}) - \log p_\theta(\hat{s}|\boldsymbol{\chi}))\}]_+^2 \tag{7}$$

Different from HMM based systems, attention based end-to-end speech recognition systems cannot generate full posterior graphs because of the explicit dependencies between output tokens. A common approximation of posterior graph is the n-best hypotheses list [8, 10, 12]. In the n-best hypotheses, the 1-best one is chosen to be the most probable incorrect hypothesis. With this approximation, we have the loss function as in equation (8):

$$L(\theta) = \sum_{(\boldsymbol{\chi}, s) \in D} [l(\hat{s}_b, s) - (\log p_\theta(s|\boldsymbol{\chi}) - \log p_\theta(\hat{s}_b|\boldsymbol{\chi}))]_+^2 \tag{8}$$

where $\hat{s}_b$ denotes the 1-best hypothesis.

Note that if the 1-best hypothesis is correct, i.e. its tokens match those of the reference sequence exactly, the gradients of the loss in equation (8) will all be zero, as will be shown in Section 2.3.

In attention based end-to-end systems, the log posterior of a hypothesis can be interpreted as $score$. The above equation can thus be written as equation (9) below:

$$L(\theta) = \sum_{(\boldsymbol{\chi}, s) \in D} [l(\hat{s}_b, s) - (score_\theta(s|\boldsymbol{\chi}) - score_\theta(\hat{s}_b|\boldsymbol{\chi}))]_+^2 \tag{9}$$

Note that the $score$s in equation (9) are not normalized by the lengths of the corresponding sequences.

2.3. Gradient Calculation for Large Margin Training

Formulating the loss function of large margin training for attention based end-to-end systems as equation (9), large margin training can be applied directly with back propagation. It is important to note that each $score$ corresponds to a different output sequence. In other words, the gradients over the two output sequences s and $\hat{s}_b$ should be calculated separately. Equation (10) shows the gradient of the large margin loss w.r.t. the reference posterior probability $\log p_\theta(s_i)$ at token i:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = \frac{\partial \sum_{(\boldsymbol{\chi}, s)} [l(\hat{s}_b, s) - (score_\theta(s) - score_\theta(\hat{s}_b))]_+^2}{\partial \log p_\theta(s_i)} \tag{10}$$

where $score(s|\boldsymbol{\chi})$ and $score(\hat{s}_b|\boldsymbol{\chi})$ are written as $score(s)$ and $score(\hat{s}_b)$ for notation simplicity, respectively. i denotes the token index in s.

Since the gradient is w.r.t. a specific sequence s, $\sum_{(\boldsymbol{\chi}, s) \in D}$ is ignored:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = \frac{\partial [l(\hat{s}_b, s) - (score_\theta(s) - score_\theta(\hat{s}_b))]_+^2}{\partial \log p_\theta(s_i)} \tag{11}$$

If we denote $[l(\hat{s}_b, s) - (score_\theta(s) - score_\theta(\hat{s}_b))]_+$ as γ_+, equation (11) can be written as equation (12) below:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = 2\gamma_+ \frac{\partial [l(\hat{s}_b, s) - (score_\theta(s) - score_\theta(\hat{s}_b))]_+}{\partial \log p_\theta(s_i)} \tag{12}$$

Since γ_+ ensures that the gradient is nonzero only when $l(\hat{s}_b, s) - (score_\theta(s) - score_\theta(\hat{s}_b)) > 0$, equation (12) can be simplified to equation (13) below:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = 2\gamma_+ \frac{\partial (l(\hat{s}_b, s) - (score_\theta(s) - score_\theta(\hat{s}_b)))}{\partial \log p_\theta(s_i)} \tag{13}$$

The threshold $l(\hat{s}_b, s)$ is typically chosen to be the word or character level edit distance. Similar to the gradient derivation for MWER loss [8, 10], we assume $\partial l(\hat{s}_b, s)/\partial \log p_\theta(s_i) = 0$. Since $score_\theta(\hat{s}_b)$ and $\log p_\theta(s_i)$ are independent, we have $\partial score_\theta(\hat{s}_b)/\partial \log p_\theta(s_i) = 0$. With these two simplifications, equation (13) can be written as equation (14):

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = -2\gamma_+ \frac{\partial score_\theta(s)}{\partial \log p_\theta(s_i)} \tag{14}$$

Note that $score_\theta(s)$ is the summation of the log posteriors over the tokens in s:

$$score_\theta(s) = \sum_j \log p_\theta(s_j) \tag{15}$$

Taking equation (15) into (14), we can get equation (16) below:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = -2\gamma_+ \frac{\partial \sum_j \log p_\theta(s_j)}{\partial \log p_\theta(s_i)} \tag{16}$$

Since log posteriors are mainly influenced by the input acoustic features and preceding output tokens, we assume that the dependency between different log posteriors can be ignored. Under this assumption, equation (16) can be simplified to equation (17) below:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = -2\gamma_+ \tag{17}$$

Note that only the tokens in s have nonzero gradients. The other token dimensions are kept untouched.

Similar to that of s, the derivative w.r.t. $\hat{s}_b$ is as follows:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(\hat{s}_{b,i})} = 2\gamma_+ \tag{18}$$

Repetitive training on the beginning correct segment of $\hat{s}_b$ may make the training process unstable and prone to overfitting. If equation (17) and (18) are applied to the same output, the gradients on the beginning correct segment will be canceled out automatically. In large margin training, however, the two gradients are applied to separate outputs. In order to avoid training instability and overfitting, we propose to apply the gradients in equation (17) and (18) starting from the first wrong token in $\hat{s}_b$. If we denote the first wrong token as $\hat{s}_{b,w}$, the gradients can be expressed as follows:

$$\frac{\partial L(\theta)}{\partial \log p_\theta(s_i)} = -2\gamma_+ \delta(i \geq w) \tag{19}$$

$$\frac{\partial L(\theta)}{\partial \log p_\theta(\hat{s}_{b,i})} = 2\gamma_+ \delta(i \geq w) \tag{20}$$

where $\delta(\cdot)$ is the Kronecker delta (indicator) function.

After the gradients w.r.t. the log posteriors been manually assigned, the gradients for the remaining parameters in the network can be calculated automatically using deep learning platforms such as PyTorch and Chainer. We will present the implementation details in the next section.

3. Experimental Setup

3.1. Data and Model

Our experiments are conducted on the Switchboard-1 Release 2 dataset. It contains 2,400 two-sided English conversations among 543 speakers. The total duration of the recordings sums up to 260 hours. We use the 2000 HUB5 evaluation set in our experiments. The number of utterances in this evaluation set is 4,458.

The inputs are 40 dimensional log-Mel features extracted using Kaldi [26]. The output layer has 49 nodes corresponding to English letters, numbers, punctuations, special transcribed notations, and indicators including 'space', 'SOS', and 'EOS'. Note that the training samples are selected based on their lengths. In our experiments, utterances longer than 1800 frames are removed from the training set.

The model in our experiments is an input-feeding LAS system. The encoder ("listener" in LAS) consists of six bidirectional LSTM (BLSTM) layers. The number of units in each layer is 512. Different from the encoder, the decoder ("speller" in LAS) uses two unidirectional LSTM layers. Each layer also has 512 units.

3.2. Implementation Details

The end-to-end model is first trained using point-wise cross entropy loss without scheduled sampling. The learning rate is initialized to 10^{-3} and is halved when the validation loss reduction is smaller than 0.01. The model is then further trained using scheduled sampling until convergence. The resulting model is used as the baseline system. Its WER is 13.3%.

Large margin training is applied to the baseline model. We use word level edit distance as $l(\hat{s}_b, s)$. The optimizer is Adam and the learning rate is $7.5 * 10^{-7}$. The dropout rate is selected to be 0.2 and the size of the mini batch 8. After every 131,072 frames been processed, a temporary model is saved. The final model is selected from the saved models.

3.2.1. Cross Entropy Regularization

We apply a cross entropy regularization term to facilitate the convergence of the end-to-end model. The cross entropy loss is added to the large margin loss. The scale for the large margin loss is 1 and that for the cross entropy regularization term 0.01.

3.2.2. Reference Sequence Score Calculation

As mentioned in Section 2.3, the score of the reference sequence s needs to be calculated separately. In order to do it, a separate forward pass of the end-to-end model is performed. During the forward pass, the tokens in s, rather than those corresponding to the largest log posteriors, are fed to the decoder to generate next tokens. The log posteriors corresponding to the reference sequence s are collected in this process to calculate $score_\theta(s)$.

3.2.3. Backpropagation Implementation Using PyTorch

In large margin training, the gradients at output nodes are assigned manually. The simple $backward$ function in PyTorch thus cannot perform the error backpropagation process. To deal with this, we first detach the decoder projection layer (i.e. the last hidden layer) of the end-to-end model as a separate model. This detached model maps from the 512 dimensional decoder output to the 49 dimensional softmaxed end-to-end model output. After assigning the gradients for output nodes,

the $backward$ function of the detached model is used to calculate the gradients w.r.t. the model parameters. After generating the gradients for the detached layer, the *torch.autograd* class in PyTorch is used to calculate the gradients for the rest of the end-to-end model automatically [27].

3.2.4. Joint Language Modeling

It is observed from prior work that decoding end-to-end speech recognition systems with shallow-fused external language models can improve the recognition performance [28, 29, 30]. During evaluation, the predictions of the language model are fused with the posterior probabilities generated by the end-to-end model at every output token. This way, the end-to-end speech recognition systems incorporates the dependencies between consecutive output tokens explicitly. Note that the language model in our experiments is trained only with the text in the Switchboard-1 Release 2 corpus, i.e. the same corpus as that of the end-to-end system [12].

4. Evaluation Results

4.1. Results and Comparisons of Large Margin Training

The results and comparisons of the large margin training scheme on the Switchboard subset of the 2000 HUB5 evaluation set are shown in Table 1.

Table 1: *Results and comparisons of large margin training.* Cross Entropy *refers to the baseline model and* MWER (4 hypotheses) *denotes the MWER scheme using 4-best hypotheses during training. The results without using the external language model are denoted as* w/o LM *and those with language model* w/ LM.

criterion	w/o LM	w/ LM
Cross Entropy [10]	13.3	-
MWER (4 hypotheses) [10, 12]	**12.2**	**12.0**
Large Margin Training	12.4	**12.0**

The results of MWER using four hypotheses and large margin training using only one hypothesis are both 12.0% with the external language model. This shows that for MAP evaluation, MBR training is an overkill and that one hypothesis may be sufficient for training.

Without using the external language model, the large margin training scheme achieves a WER of 12.4%, outperforming the baseline model by 6.8% relatively. MWER performs better than large margin training in this case. The reason may be that by minimizing the expected loss of four hypotheses, MWER is able to build an implicit language model that has a better generalization ability to the evaluation set. Note that if large margin also uses four hypotheses by applying equation (20) for each of the 4-best hypotheses separately, the WER is 12.2%. This indicates that the performance difference between large margin training and MWER without using the external language model is mainly related to the difference in the number of training hypotheses.

The large margin training scheme benefits from the external language model. The reason may be that the standard edit distance used for $l(\hat{s}_b, s)$ cannot measure the dependencies between consecutive output tokens.

4.2. Comparisons with Previously Proposed Systems

Table 2 shows the comparisons between large margin training and previously proposed end-to-end speech recognition systems.

Table 2: *Comparisons with previously proposed end-to-end systems. The Switchboard subset of the 2000 HUB5 evaluation set is denoted as* SWBD *and the CallHome subset* CH. BRNN *is short for bidirectional recurrent neural network and* BLSTM *refers to bidirectional long short-term memory (network).* Graph *and* Phone *denote two kinds of output sequences for end-to-end systems, grapheme and phoneme, respectively.* Word RNN LM *in [31] refers to a word level RNN language model.*

system	SWBD	CH
Attention + Trigram LM [3]	25.8	46.0
BRNN Graph CTC + Ngram LM [32]	20.0	31.8
BLSTM Phone CTC + Fisher LM [33]	14.8	N/A
Acoustic-to-Word [29]	14.5	25.1
Iterated CTC + Word RNN LM [31]	14.0	25.3
Large Margin Training	12.4	24.3
MWER (4 hypotheses) [10]	12.2	23.3
MWER (4 hypotheses) + LSTM LM [12]	**12.0**	**23.1**
Large Margin Training + LSTM LM	**12.0**	24.6

On the *SWBD* subset of the 2000 HUB5 evaluation set, large margin training using one hypothesis achieves the same result as MWER training using four hypotheses. On the *CH* subset, MWER performs better than large margin training. As mentioned in the previous section, the reason why MWER has a better generalization ability than large margin training may be that by using multiple hypotheses, MWER essentially enlarges the training set. Therefore, as the size of the training set increases, the performance of MWER and large margin training may get closer.

5. Concluding Remarks

In this study, we have proposed a large margin training scheme for attention based end-to-end speech recognition. Using only one hypothesis during training, the large margin scheme yields the same performance as the MWER scheme using four hypotheses. A detailed derivation for the gradient calculation of large margin training is presented in this paper. The derivation is widely applicable to other sequence discriminative training criteria such as MMI. This study also provides a new formulation of the large margin concept, paving the road towards a better combination of SVM and DNN. Future work includes using position-aware threshold functions for large margin training, deploying large margin training to multilingual tasks, improving the generalization ability of large margin training using larger corpora, and combining large margin training with other training schemes.

6. Acknowledgements

We would like to thank S. Zhang for helpful discussions on large margin training.

7. References

[1] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 577–585.

[2] 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.

[3] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5060–5064.

[4] A. Graves, "Generating sequences with recurrent neural networks," in *arXiv:1308.0850*, 2013.

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

[6] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[7] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. of EMNLP*, 2014.

[8] C. Chiu, T. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. Weiss, K. Rao, K. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[9] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C. C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 4839–4843.

[10] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," *Proc. of INTERSPEECH*, pp. 761–765, 2018.

[11] L. Bahl, F. Jelinek, and R. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 179–190, 1983.

[12] J. Cui, C. Weng, G. Wang, J. Wang, P. Wang, C. Yu, D. Su, and D. Yu, "Improving attention-based end-to-end asr systems with sequence-based loss functions," in *Proc. of SLT 2018*, 2018, pp. 353–360.

[13] S. Karita, A. Ogawa, M. Delcroix, and T. Nakatani, "Sequence training of encoder-decoder model using policy gradient for end-to-end speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 5839–5843.

[14] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[15] S. Zhang and M. Gales, "Structured svms for automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 544–555, 2013.

[16] S. Zhang, C. Liu, K. Yao, and Y. Gong, "Deep neural support vector machines for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4275–4279.

[17] S. Zhang, R. Zhao, C. Liu, J. Li, and Y. Gong, "Recurrent support vector machines for speech recognition." in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 5885–5889.

[18] T. Joachims, T. Finley, and C. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[19] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proc. of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 1997, pp. 347–354.

[20] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in n-best list rescoring," in *Proc. of Fifth European Conference on Speech Communication and Technology*, 1997.

[21] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, pp. 373–400, 2000.

[22] G. Evermann and P. C. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. of Speech Transcription Workshop*, vol. 27. Baltimore, 2000, pp. 78–81.

[23] L. R. Bahl, P. R. Brown, P. V. De Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 1986 IEEE International Conference on*, vol. 86, 1986, pp. 49–52.

[24] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Acoustics, Speech and Signal Processing (ICASSP), 2009 IEEE International Conference on*. IEEE, 2009, pp. 3761–3764.

[25] R. Schlüter, W. Macherey, B. Müller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Communication*, vol. 34, pp. 287–310, 2001.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. of IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU)*, no. EPFL-CONF-192584, 2011.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[28] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," *arXiv preprint arXiv:1708.06426*, 2017.

[29] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," *arXiv preprint arXiv:1703.07754*, 2017.

[30] Z. Chen, J. Droppo, J. Li, W. Xiong, Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 184–196, 2018.

[31] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4805–4809.

[32] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[33] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, "An empirical exploration of ctc acoustic models," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2623–2627.