On the Optimal Interpolation Weights for Hybrid Autoregressive Transducer Model

Ehsan Variani, Michael Riley, David Rybach, Cyril Allauzen
Tongzhou Chen, Bhuvana Ramabhadran

Google Inc.

{variani, riley, rybach, allauzen, tongzhou, bhuv} @google.com

Abstract

This paper explores rescoring strategies to improve a two-pass speech recognition system when the first-pass is a hybrid autoregressive transducer model and the second-pass is a neural language model. The main focus is on the scores provided by each of these models, their quantitative analysis, how to improve them and the best way to combine them to achieve better recognition accuracy. Several analyses are presented to emphasize the importance of the choice of the integration weights for combining the first-pass and the second-pass scores. A sequence level combination weight estimation model along with four training criteria are proposed which allows adaptive integration of the scores per acoustic sequence. The effectiveness of this algorithm is demonstrated by constructing and analyzing models on the Librispeech data set. It is shown that the proposed adaptive weight interpolation technique achieves 5 % relative gain over the baseline model with non-adaptive weights.

Index Terms: speech recognition, two-pass recognition, rescoring weights

1. Introduction

State-of-the-art automatic speech recognition (ASR) systems make their final recognition decision by integrating multiple sources of information such as acoustic model (AM), language model (LM), etc. In a single system, all these knowledge sources are combined to construct a large search space which allows the search algorithm access every single source of information anytime during inference. This exhaustive search can potentially lead to an accurate recognition output by the means of significant computation complexity and memory cost. A two-pass recognition system [1, 2, 3] was introduced to mitigate this problem. Here some of the knowledge sources are chosen to serve as the first-pass recognizer generating subset of likely hypotheses either in the form of a n-best list [1] or a lattice [2, 3]. These hypotheses are then reordered in the second-pass using the rest of knowledge sources.

Efficient design of a two-pass recognition system involves choices of knowledge sources, their placement (whether in the first-pass or in the second-pass) and finally effective way of combining scores provided by each of these knowledge sources. In the context of hybrid autoregressive transducer (HAT) [4], there are many ways to design an efficient two-pass recognition system which are detailed in [5]. One such design which is both server and on-device friendly is to use the HAT model as the first-pass recognizer without any external language model and reorder the hypotheses in the second-pass using a powerful external language model. For each output hypothesis, the first-pass HAT model provides an acoustic model (AM) score along with an internal language model (ILM) score. The acoustic score is the likelihood of observing an input acoustic sequence conditioned on the hypothesis. The internal language score provides prior probability that HAT model assigns to the hypothesis. The hypotheses output of the first-pass system are ordered by sum of the AM and ILM scores. In the second-pass these scores are combined by an external language model (ELM) score. The hypotheses are then reordered based on the combined score and the most likely one according to this score is set as the recognition output.

The HAT model formulates AM, ILM and ELM score combination within noisy channel framework using two constant scalar weights. These weights compensate for different dynamic ranges of the score values which is quite significant between acoustic and language model scores. The values of these weights are chosen by sweeping over a range of values on a development set with the objective of minimizing the word error rate (WER) on the set. If the dynamic range of scores significantly differs during inference, such score combination scheme might lead to quality degradation. One way of addressing this issue is to develop an adaptive score combination algorithm which predicts integration weights per input sequence.

This paper presents: (1) a quantitative analysis of AM, ILM and ELM scores for a two-pass recognition system with HAT model as the first-pass and a neural language model as the second-pass, (2) a way to improve acoustic score by leveraging the availability of the input acoustic sequence and the hypothesis at the end of first-pass (beginning of the second-pass), (3) a method to evaluate existence of the optimal combination weights which demonstrates an upper bound for the best achievable WER using adaptive weight methods, and (4) four training criteria for optimizing sequence level weights. All the analysis and experiments are conducted on the Librispeech dataset [6]. While the analysis and methods presented in this paper are used for a very specific two-pass recognition system, it can be applied to any other two-pass configuration.

2. Two-pass Speech Recognition

A HAT model without an external LM is used as the first-pass and a neural LM is used as the second-pass. Standard beam-search algorithm is used for the first-pass inference which outputs n-best hypotheses. The first-pass system provides an acoustic score and an internal language model score for each hypothesis. The acoustic score is the logarithm of the sum of all the alignment paths corresponding to the hypothesis traversed within beam-search. The ILM score is the logarithm of the prior probability assigned to each hypothesis by the first-pass HAT model. In the second-pass, The n-best list of hypotheses is reordered according to linear interpolation of these scores along with the external language model score within noisy channel formulation. Next we briefly describe modeling and inference details of each recognition pass. More details can be found in [4, 5].

For an acoustic feature sequence $x = x_{1:T}$ corresponding to a word sequence w, assume $y = y_{1:U}$ be a tokenization of w where $y_i \in M$ is either a phonetic unit or a character-based unit from a finite-size alphabet M. Since usually $T \neq U$, a notion of alignment is defined between elements of x and y. The alignment sequence $\tilde{y}$ can be defined as a sequence of $T + U$ labels, where label $\tilde{y}_{t+u+1}$ is either equal to blank symbol <b> or is equal to y_{u+1}. The HAT model formulates the local posterior distribution $P(\tilde{y}_{t+u}|x, \tilde{y}_{1:t+u-1})$ by a Bernoulli distribution with parameter $b_{t,u}$ and a label distribution $P_{t,u}$ by:

$$\begin{cases} b_{t,u} & \tilde{y}_{t+u} = \text{<b>} \\ (1 - b_{t,u})P_{t,u}(y_u|x, y_{1:u}) & \tilde{y}_{t+u} = y_u \end{cases}$$

The HAT model does not provide any strict neural parametric form for neither $b_{t,u}$ nor $P_{t,u}$. This means that these distributions can be modeled by any neural architectures with or without sharing parameters. By chaining the local posterior probabilities over an alignment path, the alignment posterior $P(y, \tilde{y}|x)$ is derived. The posterior probability of y given x is then modeled by summing all the alignment posteriors:

$$P(y|x) = \sum_{\tilde{y}:B(\tilde{y})=y} P(y, \tilde{y}|x) \qquad (1)$$

where $B : \tilde{y} \to y$ is the function that maps alignment paths to their corresponding label sequence (it removes blanks). In addition to modeling the posterior probability, the HAT model provides an estimate of the prior, or internal language model (ILM) probability, for any sequence y [4]:

$$P_{\text{ILM}}(y) = \prod_{1:U} P_{t,u}(y_u|\mathbf{0}, y_{1:u-1}) \qquad (2)$$

which is the chain of label distribution $P_{t,u}$ over labels, assuming the encoder activations are set to zero. Using this quantity and Bayes' rule, a pseudo-likelihood sequence-level score [7, 8, 9, 10, 4] is derived which can be used for integration with an external language model either during the first-pass beam search or the second-pass rescoring [4, 5].

The first-pass inference algorithm searches for the most likely alignment path $\tilde{y}^\star$:

$$\tilde{y}^\star = \underset{\tilde{y}}{\text{argmax}}\, P(\tilde{y}|x)$$

which is corresponding to the most likely hypothesis $y^\star = B(\tilde{y}^\star)$. The decoding strategy used here is time-synchronous with breadth-first search. Details of decoding parameters are described in the experiment section. The first-pass system outputs n-best hypotheses $y_1, \cdots, y_n$ with the following scores:

- AM scores: $s(y_1|x), \cdots, s(y_n|x)$ where

$$s_{\text{AM}}(y|x) = \log \sum_{\tilde{y}:B(\tilde{y})=y\ ,\ \tilde{y}\in\mathcal{S}} P(y, \tilde{y}|x) \qquad (3)$$

  which is the sum of all the alignment paths traversed within the search space $\mathcal{S}$.

- ILM scores: $\log P_{\text{ILM}}(y_1), \cdots, \log P_{\text{ILM}}(y_n)$.

2.1. Second-pass: Neural Language Model

The second-pass is a neural language model trained to maximize sequence level likelihood. This model assigns ELM score $\log P_{\text{ELM}}(y_i)$ for every hypothesis y_i in the n-best list. It is assumed that the external language model is trained on the same tokenization unit as the first-pass model. This assumption is not needed and only assumed for simplicity of equations.

Given the AM, ILM and ELM scores, the n-best hypotheses are reordered according to the following combined score:

$$\lambda_1 s_{\text{AM}}(y|x) - \lambda_2 \log P_{\text{ILM}}(y) + \log P_{\text{ELM}}(y) \qquad (4)$$

where λ_1 and λ_2 are two scalar weights and $y = y_1, \cdots, y_n$. The hypothesis with the highest score is set as recognition output.

3. Scores and Integration Weights

The AM score assigned to each hypothesis y incorporates only a subset of all the alignment paths $\tilde{y}$ corresponding to the label sequence y, the ones in the intersection of search space $\mathcal{S}$ and the alignment space $\mathcal{A}_y = \{\tilde{y}|B(\tilde{y}) = y\}$. This means that the AM score in Eq. 3 is always less than $\log P(y|x)$:

$$\begin{aligned} s_{\text{AM}}(y|x) &= \log \sum_{\tilde{y}:B(\tilde{y})=y\ ,\ \tilde{y}\in\mathcal{S}} P(y, \tilde{y}|x) \\ &\leq \log \sum_{\tilde{y}:B(\tilde{y})=y} P(y, \tilde{y}|x) \\ &= \log P(y|x) \end{aligned}$$

where equality holds iff $\mathcal{A}_y \subseteq \mathcal{S}$. This requires search space parameters to be set large enough such that the search space covers all the possible alignment paths. Through the paper, the AM score from first-pass is called partial AM score and $\log P(y|x)$ is called full AM score of hypothesis y.

At the end of the first-pass, both the acoustic sequence x and the hypothesis label sequence y are available, thus the scoring function of Eq. 1 can be modified as:

$$s(x, y_i) = \lambda_1 \log P(y_i|x) - \lambda_2 \log P_{\text{ILM}}(y_i) + \log P_{\text{ELM}}(y_i) \qquad (5)$$

One way of choosing the scalar weights λ_1 and λ_2 in this equation is to search through a range of values on a development set and choose the values which minimize the WER on this set. The chosen weights are then kept constant for every acoustic sequence in the test time. This might not be the optimal way of combining scores for two reasons: first, the dynamic range of scores in the test time might differ from the ones on the development set which can cause WER degradation, second, using different weights per acoustic sequence might lead to WER improvement. The next two sections explore the existence of an optimum weight and how to estimate it with a parameterized model.

3.1. Optimal combination weights

Let y_o be the oracle hypothesis, the one with the lowest WER among the the n-best hypotheses list. If there exists λ_1 and λ_2 such that the combined score of the oracle hypothesis be greater than or equal to any other hypotheses in the list, then the second-pass rescoring can lead to the best achievable WER. Such weight values exist iff for any $i = 1, \cdots, n$:

$$\lambda_1 s_1^o - \lambda_2 s_2^o + s_3^o \geq \lambda_1 s_1^i - \lambda_2 s_2^i + s_3^i \qquad (6)$$

where s_1^i, s_2^i, and s_3^i are the AM, ILM, and ELM scores, respectively. This can be formulated within a system of inequalities as $A\boldsymbol{\lambda} \leq \mathbf{0}$ where A is a $n \times 3$ matrix with i^{th} row being:

$$A[i,:] = [s_1^i - s_1^o, s_2^o - s_2^i, s_3^i - s_3^o]$$

and $\boldsymbol{\lambda}' = [\lambda_1, \lambda_2, 1.0]$. The simplex algorithm [11] can be used to find the feasible solution region of this system of inequalities. If there exists a solution, then there is a set of combination weights that move the oracle hypothesis to the top of the hypotheses list as the most likely hypothesis.

3.2. Adaptive weight estimation

Instead of using constant weights for every acoustic sequence x, the combination weights can be adaptive. For hypothesis i the score function $s(x, y_i; \theta)$ is defined as:

$$\lambda_1(x;\theta) \log P(y_i|x) - \lambda_2(x;\theta) \log P_{\text{ILM}}(y_i) + \log P_{\text{ELM}}(y_i) \quad (7)$$

Note that the weight function can also depend on hypotheses $y_{1:n}$, i.e. $\lambda_i(x, y_{1:n}; \theta)$. For experiments in this paper, only dependency to x is considered. The choice of architecture to model θ is detailed in Section 4. To optimize modeling parameters a proper training criterion $\mathcal{L}$ is required:

- **Regression**:

$$\mathcal{L}(x, y_{1:n}; \theta) = \sum_{i=1,2} \|\lambda_i(x;\theta) - \bar{\lambda}_i(x)\|_p \quad (8)$$

where $\|.\|_p$ is the p-norm and $\bar{\lambda}_i(x)$ is the groundtruth value for i^{th} weight. These values are some solutions of the system of inequalities in Sec. 3.1. Here, simplex algorithm can be used to find one feasible solution which is set as the groundtruth. The regression criterion effectively learns the decision boundary that separates oracle hypothesis score from others.

- **Binary Classification**:

$$\mathcal{L}(x, y_{1:n}; \theta) = \sum_{i=1}^{n} \sum_{j=1}^{n} H(p_{i,j}(x;\theta),\ \bar{p}_{i,j}(x)) \quad (9)$$

where $H(.)$ is the cross entropy function and

$$p_{i,j}(x;\theta) = \frac{\exp(s(x, y_i; \theta))}{\exp(s(x, y_i; \theta)) + \exp(s(x, y_j; \theta))}$$

$$\bar{p}_{i,j} = \begin{cases} 1.0 & \text{WER}[i] \leq \text{WER}[j] \\ 0.0 & \text{Otherwise} \end{cases}$$

where $s(x, y_i; \theta)$ is the parameterised combined score of Eq. 5. This criterion pushes the oracle hypothesis to have the highest score and preserves the order of hypotheses; lower score, higher WER.

- **Oracle Prediction**:

$$\mathcal{L}(x, y_{1:n}; \theta) = H(p(x, y_{1:n}; \theta),\ \bar{p}(x, y_{1:n})) \quad (10)$$

where both $p(.)$ and $\bar{p}(.)$ are discrete distribution defined for each hypothesis:

$$p(x, y_{1:n}; \theta)[i] = \frac{\exp(s(x, y_i; \theta))}{\sum_{j=1}^{n} \exp(s(x, y_j; \theta))} \quad (11)$$

$$\bar{p}(x, y_{1:n})[i] = \begin{cases} 1.0 & i = o \\ 0.0 & \text{Otherwise} \end{cases}$$

where o is the index of the oracle hypothesis. Note that this is not the only way to define the groundtruth distribution; alternatively it can be defined as a function of edit-distance:

$$\bar{p}(x, y_{1:n})[i] = \frac{\exp(\text{ed}(y_i, \text{ref}))}{\sum_{j=1}^{n} \exp(\text{ed}(y_j, \text{ref})}$$

where ed(.) is the edit-distance function and ref is the reference sequence. The oracle prediction criterion has been also used in [12]. Unlike the binary classification criterion, this criterion does not preserve the order of hypotheses. It only boosts the oracle score independent of how the other hypotheses are scored with respect to each other and their corresponding word error rate.

- **Minimum Bayes Risk**[13]:

$$\mathcal{L}(x, y_{1:n}; \theta) = \sum_{i=1}^{n} \text{ed}(y_i, \text{ref}) p(x, y_{1:n}; \theta)[i] \quad (12)$$

where $p(x, y_{1:n}; \theta)[i]$ is defined in Eq. 11.

4. Experiments

Data: the training data used here is the full 960 hours of the publicly available Librispeech ASR corpus [6]. The input features are 256-dim log Mel extracted from a 64 ms window of the speech signal with a 30 ms shift [14]. The LSTM baselines are trained on clean data, while the conformer baselines are trained with the SpecAugment library [15] using the recipe parameters described in [16]. The full 810M word token Librispeech text corpus was used to train the second-pass neural LM. The transcripts are used without any processing and tokenized by the 28 graphemes that appear in the Librispeech data.

Architecture: For the first-pass model, two HAT models are considered which are only different on the choice of encoder architecture. Both models use streaming encoder network, first model uses 5 layers of long short-term memory (LSTM) [17] with 1024 cells per layer. The encoder output is projected to 768-dim vector with a linear layer, matching the decoder network dimension. The second model uses 12 layer conformer encoder [16] with model dimension 512 followed by a linear layer with output dimension 640. Both models use a two layers decoder network with 256 LSTM cells per layer. The decoder network output is linearly projected to the same dimension as the encoder output. The LSTM model has about 38M parameters and the conformer model has 87M parameters. The neural LM is a 4 layers LSTM with 2048 cells per layer.

The adaptive weight combination model uses 2 layers of bidirectional LSTM with 256 cells per layer. The second layer output is linearly projected and summed over sequence to create a single 128 dimensional sequence embedding. This embedding vector is then linearly projected into a 2-dim vector corresponding to the model weights in Eq. 7. The adaptive weight combination model has 4M parameters in total.

Training: The LSTM models are trained on 4×4 TPUs with a batch size 4096. The conformer models are trained on 8×8 TPUs with a batch size of 2048. The training examples with more than 768 feature frames or more than 384 labels are filtered out. The LSTM models are trained with Adafactor optimizer [18] with all the default parameters. The conformer model uses Adam optimizer [19] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ with transformer learning rate schedule [20].

Evaluation: The time-synchronous decoding strategy used here is breadth-first search. At every time frame t, all the paths with

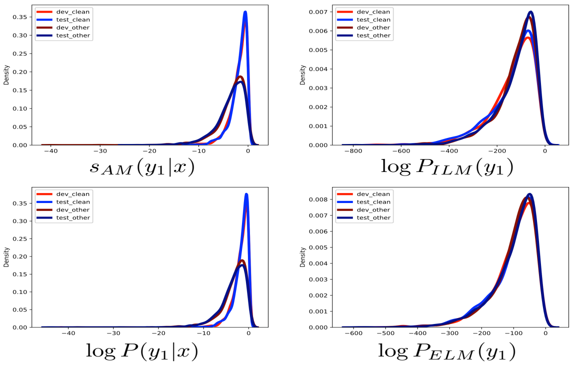

Figure 1: *Distribution of AM score (top-left), ILM score (top-right), full AM score (bottom-left), ELM score (bottom-right).*

model	clean [WER]		other [WER]	
	Hyp1	Oracle	Hyp1	Oracle
LSTM-HAT [38M]	8.6	3.9	20.1	12.3
Confomer-HAT [87M]	6.6	4.0	11.6	7.0

Table 1: *First-pass: baselines WER of top hypothesis (Hyp1) and Oracle hypothesis.*

the same label prefix(without blank) are merged and their probabilities are summed. The beam size of 100 and beam width of 20 were used for the first-pass inference. The top 20 hypotheses of the first-pass recognition output are passed as n-best list to the second-pass. The results are reported on standard Librispeech test sets: test_clean and test_other. Table.1 summarizes the performances of both first-pass models in terms of top hypothesis and oracle WER.

AM, ILM and ELM scores: Figure. 1 presents distribution of the partial AM score, ILM score, full AM score and ELM score for the top hypothesis output of the first-pass LSTM model. The distributions are plotted for both dev and test sets in clean and other. The dynamic range and shape of curves are very consistent between dev and test. The full AM score has less variance compared to the partial AM score. Similarly the ELM scores seem more concentrated than the ILM score. The dynamic range of AM and LM scores are significantly different which explains the need for accurate estimation of the combination weights of Eq. 4.

Rescoring with full AM score: The least expensive rescoring algorithm of the hypotheses in the second-pass is to use the full AM score instead of the partial score. The results are shown in Table. 2 (row 1 and row 5). The full score rescoring does not bring any significant WER gain. This might suggest that that the decoding parameters, beam width and beam size, are set large enough such that the difference between partial and full AM score does not make a considerable impact on reordering of hypotheses.

Rescoring with AM and ILM scores: If the external LM is not present, the hypotheses list can be rescored using ILM score instead, row 3 and row 7 of Table. 2. This effectively does not introduce any computation cost while can bring slight WER improvement particularly for the weaker first-pass model: 8.5% → 8.2% on test_clean and 20% → 19.4% on test_other.

Rescoring with all scores: Combining AM, ILM and ELM scores together leads to the best WER for both LSTM and Conformer model (row 4 and row 8 in Table. 2). The relative WER gains for the LSTM model are 23% and 20% on test_clean and test_other, respectively. The Conformer model WER is improved by 12% and 14% on the same test sets. While the LSTM model performs significantly weaker than the Conformer model after first-pass, the performance gap is considerably reduced after the second-pass rescoring.

model	ILM score	ELM score	constant λ		adaptive λ	
			clean	other	clean	other
LSTM	N	N	8.5	20.0	8.5	20.0
		Y	6.8	17.2	5.9	16.5
	Y	N	8.2	19.4	7.2	17.2
		Y	**6.0**	**16.0**	**5.1**	**15.1**
Conformer	N	N	6.6	11.5	6.6	11.5
		Y	6.1	10.5	5.2	9.3
	Y	N	6.5	11.4	5.9	10.5
		Y	**5.8**	**9.9**	**4.8**	**8.6**

Table 2: *Comparison of different rescoring strategies for LSTM and Conformer models.*

WER[%]	regression	binary class.	oracle pred.	minimum bayes risk
clean	7.5	6.0	**5.6**	6.1
other	18.3	15.8	**15.6**	16.0

Table 3: *Comparison of different training criteria for adaptive weight combination for LSTM model.*

Table. 2 also reports the WER for the rescoring strategy which ignores ILM score and merely uses the interpolation of AM score with ELM score (row 2 and 6). While this rescoring strategy improves over baseline (no language model), its performance lags behind the resocring strategy which uses all three scores. This demonstrates the importance of noisy channel formulation in Eq. 4.

Best feasible WER: Last column of Table. 2 shows the best feasible WER for different rescoring strategies assuming the weights are sequence dependent. These values are calculated as follows: a system of inequalities like Eq. 6 is formed for each example in the test set, the feasibility of existence of a solution is evaluated using Simplex algorithm [11], if this system has a solution then there is a combination weight which boost the oracle hypothesis to the top, otherwise the hypotheses order is remained unchanged. This metric approximates the potential WER gain if the optimal combination weights be used. This quantity is slightly higher than oracle WER and considerably lower than best WER with constant weights in Table. 2.

Adaptive weights: Table 3 compares four training criteria presented in Section 3. The regression criterion performs worse than constant weight combination. This criterion directly learns the decision boundary (combination weights) which is not that straight-forward. The binary classification and minimum bayes risk criteria are performing on par with the constant weight combination scheme. The oracle prediction criterion significantly improves the other criteria and the constant weight combination scheme: 6% → 5.6% on test_clean and 16.0 → 15.6%.

5. Conclusions

Several rescoring strategies for a two-pass speech recognition system was presented. The first-pass is a HAT model and the second-pass is a neural LM model. It was shown that combining the AM, ILM and ELM score within noisy channel formulation can significantly outperform other rescoring strategies. An adaptive score combination scheme is proposed along with different training criteria. The benefit of adaptive score combination scheme was demonstrated on the Librispeech dataset.

6. References

[1] R. Schwartz and Y.-L. Chow, “The n-best algorithms: an efficient and exact procedure for finding the n most likely sentence hypotheses,” in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1990, pp. 81–84.

[2] H. Ney and X. Aubert, “A word graph algorithm for large vocabulary, continuous speech recognition,” in *Third International Conference on Spoken Language Processing*, 1994.

[3] M. Riley, A. Ljolje, D. Hindle, and F. Pereira, “The at&t 60,000 word speech-to-text system.” in *Eurospeech*, 1995.

[4] E. Variani, D. Rybach, C. Allauzen, and M. Riley, “Hybrid autoregressive transducer (HAT),” in *ICASSP*, 2020, pp. 6139–6143.

[5] C. Allauzen, E. Variani, M. Riley, D. Rybach, and H. Zhang, “A hybrid seq-2-seq asr design for on-device and server applications,” 2021.

[6] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[7] N. Morgan and H. Bourlard, “Continuous speech recognition using multilayer perceptrons with hidden markov models,” in *International conference on acoustics, speech, and signal processing*. IEEE, 1990, pp. 413–416.

[8] E. Variani, E. McDermott, and G. Heigold, “A gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4270–4274.

[9] N. Kanda, X. Lu, and H. Kawai, “Minimum bayes risk training of ctc acoustic models in maximum a posteriori based decoding framework,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4855–4859.

[10] E. McDermott, H. Sak, and E. Variani, “A density ratio approach to language model fusion in end-to-end automatic speech recognition,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 434–441.

[11] G. B. Dantzig, “Origins of the simplex method,” in *A history of scientific computing*, 1990, pp. 141–151.

[12] E. Variani, T. Chen, J. Apfel, B. Ramabhadran, S. Lee, and P. Moreno, “Neural oracle search on n-best hypotheses,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7824–7828.

[13] J. Kaiser, B. Horvat, and Z. Kacic, “A novel loss function for the overall risk criterion based discriminative training of hmm models.” in *INTERSPEECH*, 2000, pp. 887–890.

[14] E. Variani, T. Bagby, E. McDermott, and M. Bacchiani, “End-to-end training of acoustic models for large vocabulary continuous speech recognition with tensorflow,” in *INTERSPEECH*, 2017, pp. 1641–1645.

[15] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.

[16] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” *arXiv preprint arXiv:2005.08100*, 2020.

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

[18] N. Shazeer and M. Stern, “Adafactor: Adaptive learning rates with sublinear memory cost,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4596–4604.

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

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.