



# Improving Tail Performance of a Deliberation E2E ASR Model Using a Large Text Corpus

Cal Peysner, Sepand Mavandadi, Tara N. Sainath, James Apfel, Ruoming Pang, Shankar Kumar

Google Inc.

{cpeysner, sepand, tsainath, japfel, rpang, shankarkumar}@google.com

## Abstract

End-to-end (E2E) automatic speech recognition (ASR) systems lack the distinct language model (LM) component that characterizes traditional speech systems. While this simplifies the model architecture, it complicates the task of incorporating text-only data into training, which is important to the recognition of tail words that do not occur often in audio-text pairs. While shallow fusion has been proposed as a method for incorporating a pre-trained LM into an E2E model at inference time, it has not yet been explored for very large text corpora, and it has been shown to be very sensitive to hyperparameter settings in the beam search. In this work, we apply shallow fusion to incorporate a very large text corpus into a state-of-the-art E2E ASR model. We explore the impact of model size and show that intelligent pruning of the training set can be more effective than increasing the parameter count. Additionally, we show that incorporating the LM in minimum word error rate (MWER) fine tuning makes shallow fusion far less dependent on optimal hyperparameter settings, reducing the difficulty of that tuning problem.

## 1. Introduction

Rare words pose an ongoing problem to building high-quality speech recognition systems. Since rare words are likely to be named entities such as names and locations, these “tail” words are often critical to the meaning of the decoded transcript. Since they do not occur often in the audio-text pairs that comprise an ASR system’s training set, they are difficult to predict correctly.

Conventional ASR systems contain separate acoustic, pronunciation, and language models which are run one after another. In such systems, the distinct language model provides an opportunity to train part of the model on text-only data, which is often far more plentiful than audio-text pairs, and can contain many occurrences of words that are rare in the acoustic data. The independence of the LM from the ASR system allows its dataset or training procedure to be adapted to specific domains, including tail words [1, 2].

E2E ASR systems consist of a single neural network in which all components are jointly trained. These models offer the advantage of simplifying the alignment of audio to text [3], as well as decreased model size [4]. However, there is no explicit LM in an E2E architecture, complicating the task of integrating text-only data. Many “LM fusion” methods have been proposed, including “shallow fusion” [5], in which LM logits are interpolated with those of an E2E model during inference, as well as more sophisticated methods such as “deep” and “cold” fusion, in which the LM is incorporated into the neural architecture of the E2E system [5, 6]. In [7], shallow fusion was shown to be the most effective fusion method with a state-of-the-art E2E system, although the “density ratio method”, has

been shown to outperform shallow fusion for a domain transfer scenario [8].

Earlier works on shallow fusion such as [9] and [10] use LMs taken from Kaldi [11] recipes which are trained on no more than a few hundred million words. Of course, language models can scale to far larger datasets. [12] trained an RNN-LM on the One Billion Word Benchmark [13], while [14] trained a transformer on 8 million web documents totaling 40GB of text. To our knowledge, the study that uses the most text data in training an RNN-LM for shallow fusion to date is [15], which uses about 4 billion words.

The research has also shown that shallow fusion is difficult to implement correctly. In [9], it is shown that without careful tuning of several hyperparameters, shallow fusion causes transcripts to be cut off after only a few words, massively degrading performance.

We have two goals in this work. First, we seek to reduce the difficulty of tuning fusion hyperparameters. We show that applying shallow fusion during minimum word error rate (MWER) training adapts the model to a particular setting of hyperparameters, and almost eliminates the impact of those parameters in inference. Second, we seek to scale shallow fusion to a text corpus of about 50 billion words, an order of magnitude larger than [15]. We show that tail performance can be improved by careful pruning of the dataset without resorting to extremely large model sizes.

In this study, we focus on the particularly difficult problem of tail words. We use shallow fusion to incorporate an LM into an E2E model trained in the recent deliberation framework [16], which already achieves state-of-the-art transcription quality on rare words. We show that shallow fusion with a large text corpus yields further improvements on the tail.

The rest of this paper is organized as follows. Section 2 outlines the architecture of our deliberation model and summarizes the techniques of MWER fine-tuning and shallow fusion. Section 3 describes the techniques we use to achieve the two goals given above. Section 4 gives details on our dataset and model architecture. Section 5 gives results and analysis, and we conclude in Section 6.

## 2. Background

In this section, we summarize our baseline model, fine-tuning procedure, and method for language model integration.

### 2.1. Deliberation Architecture

Two-pass ASR models combine a pre-trained recurrent neural network transducer (RNN-T) [17] with a second decoder that rescores top-n hypotheses [18]. In a deliberation model, on the other hand, the second decoder has the option to attend to the RNN-T hypotheses instead of rescore them, allowing all parts

of the model to be trained together.

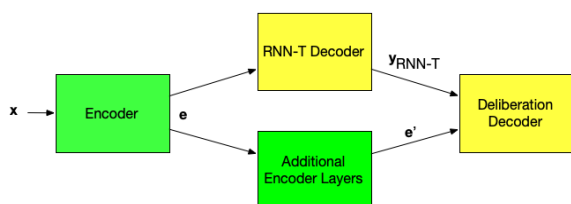


Figure 1: *Deliberation Architecture, adapted from [16].*

More specifically, a deliberation model’s encoder consumes acoustic features  $\mathbf{x}$  and maps them onto encoder features  $\mathbf{e}$ . The RNN-T decoder attends to the encoder features, and an  $n$ -best list of hypothesis  $\mathbf{y}_{\text{RNN-T}}$  is extracted using a beam search. A second encoder adapts  $\mathbf{e}$  into modified encoder features  $\mathbf{e}'$  to be consumed by the deliberation decoder. The deliberation decoder attends to both features derived from  $\mathbf{y}_{\text{RNN-T}}$  and  $\mathbf{e}'$ , and the final transcript is extracted with a second beam search.

## 2.2. MWER Fine-tuning

MWER training [19] is a fine-tuning procedure designed to directly minimize the number of word errors instead of cross-entropy. In MWER training, we seek to optimize the expected number of word errors over all possible hypotheses. Since we cannot practically marginalize over all possible output sequences, we instead compute the expected word error rate from a sample of predictions:

$$L(\mathbf{x}, \mathbf{y}^*) = \sum_{\mathbf{y} \in \mathbf{B}} P(\mathbf{y}|\mathbf{x}) \hat{W}(\mathbf{y}, \mathbf{y}^*) \quad (1)$$

where  $\mathbf{x}$  is the input acoustic features,  $\mathbf{y}^*$  is the ground truth, and  $\hat{W}$  gives a normalized word error count. Here,  $\mathbf{y}$  is a hypothesis from a beam  $\mathbf{B}$  that is sampled from the model using a beam search, and the posterior  $P$  is normalized accordingly so that all probabilities sum to one. It was demonstrated in [20] that this method is effective for beam sizes as small as 4.

## 2.3. Shallow Fusion

In shallow fusion, a language model is incorporated into ASR decoding by interpolating the posteriors directly:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P_{\text{AM}}(\mathbf{y}|\mathbf{x}) + \alpha P_{\text{LM}}(\mathbf{y}) + \beta \mathbf{C} \quad (2)$$

where  $\hat{\mathbf{y}}$  is the selected hypothesis,  $P_{\text{AM}}$  and  $P_{\text{LM}}$  are posteriors in the acoustic and language models respectively, and  $\alpha$  and  $\beta$  are hyperparameters.  $\mathbf{C}$  is a *coverage term* as in [9], which seeks to discourage truncated transcripts by rewarding hypotheses that have been allocated weight above some threshold by the attention mechanism.

# 3. Methods

In this section, we describe our techniques for fusing an LM trained on our large text corpus into our deliberation model.

## 3.1. The Truncation Problem

In [9], the authors identified a failure mode for shallow fusion in which the fused model predicts a shortened transcript consisting of only the first few words that were spoken. As we will see, this “truncation” problem turned out to be quite severe

when incorporating our LMs, which were trained on a large text corpus.

### 3.1.1. Hyperparameters

There are several hyperparameters proposed in the literature for the truncation problem. We experimented with tuning the following:

- The coverage penalty  $\mathbf{C}$ , as in equation 2 above.
- The beam size. In principle, increasing the size of the beam leaves room for longer, less errorful hypotheses on the beam even when truncated hypotheses are present.
- The *maximum EOS logprob delta*. This hyperparameter is proposed in [9]. When a hypothesis is ended with the EOS token during beam search, it must have a log probability no worse than that of the best hypothesis so far minus this value in order to be removed from the beam and marked as complete.

### 3.1.2. MWER Fusion

It was shown in [21] that using a particular hyperparameter setting (blank scale, in that work) during training of a conventional ASR system can sometimes adapt the model to that setting in inference. We attempt to reduce the difficulty of the hyperparameter tuning problem given above by showing that this can be done with beam search parameters in a deliberation model.

MWER fine tuning provides an opportunity to do this by running a beam search during training. Since this beam search will include an LM in inference, we would like to perform shallow fusion during MWER fine tuning. [22] develops a technique for fusion with RNN-T in which the LM’s logit values are added to RNN-T’s non-blank outputs, while leaving the logit for the blank output unchanged, and demonstrates small performance improvements. Unlike [22], this work seeks to use MWER training as a hyperparameter adaptation mechanism. So, we instead fine tune the beam search of our second decoder, which will be the site of shallow fusion during inference. Since this decoder does not emit a blank output, we can define a loss with direct logit interpolation:

$$L(\mathbf{x}, \mathbf{y}^*) = \sum_{\mathbf{y} \in \mathbf{B}_{\text{LM}}} (P_{\text{AM}}(\mathbf{y}|\mathbf{x}) + \alpha P_{\text{LM}}(\mathbf{y}) + \beta \mathbf{C}) \hat{W}(\mathbf{y}, \mathbf{y}^*) \quad (3)$$

where  $\mathbf{B}_{\text{LM}}$  are hypotheses drawn using a beam search with shallow fusion.

## 3.2. Taking Advantage of a Large Text Corpus

We implement the following pruning scheme to eliminate noisy data and reduce overfitting to extremely common sentences (e.g. “facebook”):

1. For every sentence, each unigram is compared against a 1 million word vocabulary. Any unigram not in this list is considered to be misspelled, and the sentence is discarded.
2. If a sentence is duplicated  $n$  times in the remaining examples, all but  $\log(n)$  examples are discarded.
3. The desired number of sentences are selected by random sampling.

Large text corpora have been exploited successfully for ASR in the past by sampling a training set that is relevant

to a domain of interest [23]. These results, however, used a maximum-entropy LM, which presents a convex optimization problem that scales naturally to large amounts of data, while we seek to optimize a non-convex RNN-LM. Also, these results targeted geographical queries, which are plentiful in ASR training corpora, while we seek to improve performance on rare words. Nevertheless, we seek to adapt this method to our problem by experimenting with an additional step between steps 2 and 3 above:

- 2\*. For every example, each unigram is compared to a list of word counts from the deliberation model’s training data. Any sentence not containing at least one “rare” word is discarded, where rareness is defined as occurring a number of times smaller than some threshold.

All together, this scheme is designed to take advantage of the large size of our text corpus while still maintaining a manageable number of sentences for LM training. We weigh the impact of this data reduction against that of increasing the LM’s size.

## 4. Experiments

In this section, we describe the parameters of our experiments. We also describe our methods for measuring the success of our LM integration and evaluating performance on the tail.

### 4.1. Deliberation Model Training

Our deliberation model is similar to that presented in [16]. We use 128-dimensional log-Mel audio features with a 32ms window and 10ms shift. The RNN-T component of the deliberation model contains eight LSTM layers in its encoder, each with 2,048 units and a 640-dimensional projection. The joint network contains 640 units, followed by a final softmax layer. Hypotheses from RNN-T are passed to a two-layer bidirectional LSTM which projects them into a 320-dimensional space. Our second decoder attends to both these features and RNN-T encoder output and emits context vectors which are passed to a final 2-layer LSTM.

Our training set is described in [24]. Transcripts are lower-cased and processed with a 4k word piece model.

### 4.2. Language Model Training

Our language models are similar to those in [7]. The models consist of LSTM layers with 512 nodes each, with a projection layer of 256 nodes. Our baseline model has two hidden layers.

The models are trained on a sample of anonymized production traffic to Google applications. We divide this data into domains that describe the origin of the queries. All examples are stripped of metadata, so that only the query text is visible to the model. Our training set is selected from this data using the pruning procedure outlined in section 3.2. The total size of our data before pruning is about 230 billion examples. Vocabulary pruning (Step 1) reduces that size to about 218 billion, and log  $n$  pruning (Step 2) further reduces the size to about 25 billion examples.

For our baseline models, we omit rare word filtering (Step 2\*) from our pruning procedure and sample down to a final size of 4.5 billion examples (about 50 billion words). When we include rare word filtering, we obtain a dataset of about 1 billion total examples (about 11 billion words), and omit Step 3.

### 4.3. Evaluation Sets

We would like to create evaluation sets that measure the degree to which our LM has been integrated into our model, and to determine performance on the tail. We create separate test sets for these two purposes. We split our test sets into those focused on geographical queries (Maps) and general queries (Search). The test sets are created by looking for utterances in the text data that have a very different perplexity distribution compared to the audio-text pair training data.

To measure LM integration, we build test sets consisting of words that are common in the LM training data but rare in the AM training data. To this end, we compute unigram statistics for both corpora and construct a list of unigrams that occur at most five times in the AM data (about three quarters of all words) and at least 150 times in the LM data (about 99% of all

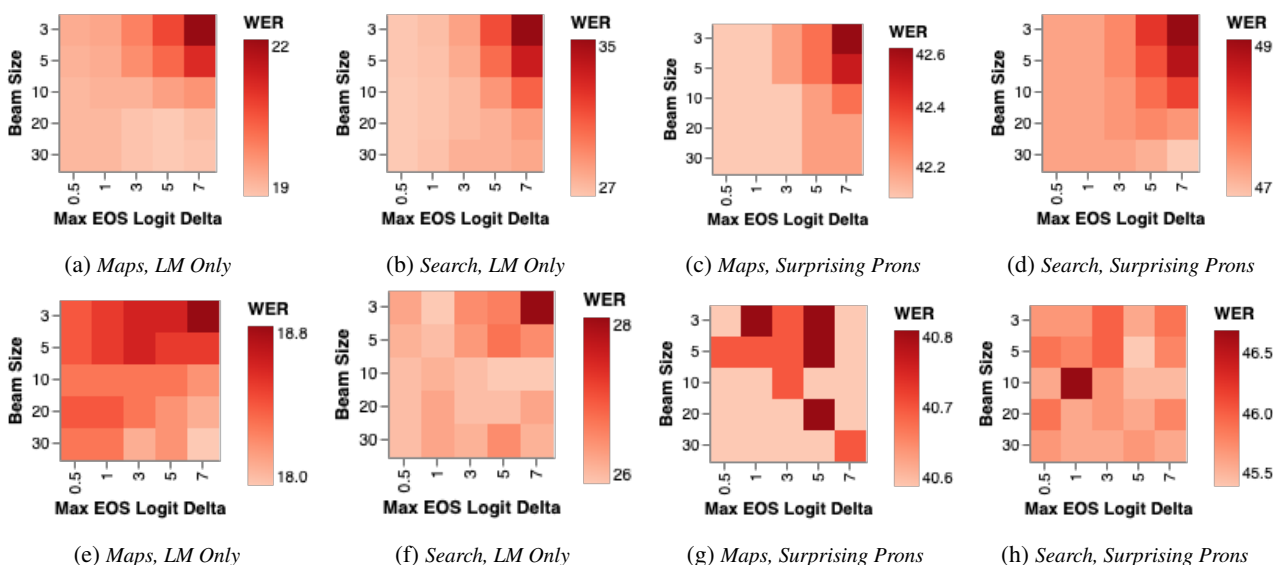


Figure 2: A sweep of maximum EOS log-probability delta and beam size on our test sets, before (a-d) and after (e-h) MWER fine tuning.

Table 1: WER Results of Expanded Models

Experiment	Maps, LM Only	Search, LM Only	Maps, Surprising Prons	Search, Surprising Prons
<b>E1</b>	18.8	27.3	42.1	47.6
<b>E2</b>	18.8	27.1	42.1	47.4
<b>E2-4</b>	18.7	26.9	41.9	47.1
<b>E2-6</b>	18.6	27	41.8	47.2
<b>E2-8</b>	18.6	26.9	41.9	47.1
<b>E3</b>	18.7	26.5	41.2	46.6
<b>E4</b>	18.7	26.6	41.4	46.7

words).

To measure tail performance, we target words that have pronunciations that are surprising given the spelling. Unusual pronunciations have been shown to be difficult for ASR systems [25, 26, 27]. To select examples with surprising utterances, we manually assemble a map from grapheme sequences to corresponding phoneme sequences. Our mapping consists of 487 correspondences. For a given example, we process each unigram grapheme-by-grapheme, using the map to assemble a list of possible corresponding phoneme sequences. If none of the predicted pronunciations match the true pronunciation of the unigram, we consider the unigram to have surprising pronunciation.

For each test set, we select 10000 examples and synthesize audio for each transcript with a TTS system as in [28].

## 5. Results

This section presents our experimental results and discussion.

### 5.1. Truncation

We find that optimizing only the LM interpolation weight  $\alpha$  and coverage penalty weight  $\beta$  does not yield improvement over our baseline model. Improvement was only shown after tuning beam size and maximum EOS logprob delta, which relate directly to the beam search. To understand the problem, we compare our models’ word error rate to “truncation word error rate”, which is the word error rate on examples for which the prediction has at most half as many unigrams as the reference. Table 2 compares WER and Truncation WER for our baseline deliberation model to a fusion model with  $\alpha = 0.1$  and  $\beta = 0.06$  and to a second fusion model in which the beam size is set to 20 and maximum EOS logprob delta is set to 0.05. This data suggests that truncation errors are largely responsible

Table 2: Impact of the Truncation Problem

	WER	Truncation WER
Baseline	19.9	2.0
Fusion	21.0	3.6
Fusion w/ BS Params	18.7	2.2

(a) Maps, LM Only

	WER	Truncation WER
Baseline	28.4	6.9
Fusion	31.5	7.6
Fusion w/ BS Params	27.1	7.1

(b) Search, LM Only

for the degradation in WER in the initial fusion model, and that tuning the beam search parameters recovers those losses.

Figure 2 (a-d) shows the results of a sweep of the two beam search parameters: beam size and maximum EOS logprob delta. Interestingly, while we find that increasing beam size yields improvements, for a sufficiently small value of maximum EOS logprob delta the beam size does not make a difference. Nevertheless, it is clear that WER results are highly dependent on correct setting of these hyperparameters.

We find that MWER fine-tuning dramatically diminishes the importance of beam search parameters in evaluation. Figure 2 (e-h) shows the results of training 25 MWER models, using the same combinations of maximum EOS logprob delta and beam size from above during the MWER beam search and then evaluating using shallow fusion with those same parameters. We find a significantly smaller range of WER than before MWER fine tuning. This suggests that MWER fine tuning serves to adapt a model to some choice of beam search parameters by using those parameters during training. This could make MWER useful as a tool to alleviate the difficulty of hyperparameter tuning in shallow fusion.

### 5.2. Language Model Size

We compare the importance of model size to data selection criterion in LM training. Table 1 gives results for shallow fusion with four progressively larger LMs including our baseline (**E1**), an expanded model in which the LM’s projection layer is removed effectively doubling the parameter count (**E2**), and that same model with 4, 6, and 8 hidden layers (**E2-4**, **E2-6**, **E2-8**). Table 1 also gives results for the 4-layer variant in which Step 2\* of the pruning procedure described in Section 3.2 is applied to the training data (**E3**), and finally where the training set is further sampled down to about 50 million examples (**E4**).

We see that filtering rare words shows significantly larger gains than increasing model capacity. This suggests that it is easier to take advantage of a large text corpus by selecting a subset of relevant examples than it is to model the entire distribution. Interestingly, this benefit is strongest when we only prune to 1 billion examples, and weakens when we further prune down to 50 million. This further suggests that an RNN-LM used in fusion is capable of benefiting from a very large text corpus.

## 6. Conclusions

In this paper, we’ve explored shallow fusion using a very large text-only corpus. We’ve quantified and explored solutions to the truncated utterances problem and demonstrated that MWER fine tuning almost eliminates the need for hyperparameter tuning. Finally we showed how a pruning strategy can beat out large models in taking advantage of large amounts of text data.

## 7. References

- [1] H. Sak, F. Beaufays, K. Nakajima, and C. Allauzen, "Language model verbalization for automatic speech recognition," in *IEEE ICASSP*, 2013, pp. 8262–8266.
- [2] S. Huang and S. Renals, "Hierarchical bayesian language models for conversational speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 1941–1954, 2010.
- [3] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, p. 1018, 08 2019.
- [4] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *IEEE ICASSP*, 2019, pp. 6381–6385.
- [5] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *CoRR*, vol. abs/1503.03535, 2015. [Online]. Available: <http://arxiv.org/abs/1503.03535>
- [6] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *INTERSPEECH*, 2018.
- [7] S. Toshiwal, A. Kannan, C.-C. Chiu, Y. Wu, T. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," *IEEE SLT*, 2018.
- [8] 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)*, 2019, pp. 434–441.
- [9] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *INTERSPEECH*, 2016.
- [10] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 206–213, 2017.
- [11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlcek, Y. Qian, P. Schwarz, J. Silovsk, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.
- [12] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *CoRR*, vol. abs/1602.02410, 2016. [Online]. Available: <http://arxiv.org/abs/1602.02410>
- [13] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *INTERSPEECH*, vol. abs/1312.3005, 2014.
- [14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: <https://openai.com/blog/better-language-models/>
- [15] A. Kannan, Y. Wu, P. Nguyen, T. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *IEEE ICASSP*, 04 2018, pp. 1–5828.
- [16] K. Hu, T. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *IEEE ICASSP*, 2020.
- [17] A. Graves, "Sequence transduction with recurrent neural networks," in *International Conference of Machine Learning (ICML) Workshop on Representation Learning*, vol. abs/1211.3711, 2012.
- [18] C.-C. Chiu, D. Rybach, I. McGraw, M. Visontai, Q. Liang, R. Prabhavalkar, R. Pang, T. Sainath, T. Strohmaier, W. Li, Y. R. He, and Y. Wu, "Two-pass end-to-end speech recognition," in *Interspeech*, 2019.
- [19] M. Shannon, "Optimizing expected word error rate via sampling for speech recognition," in *INTERSPEECH*, 2017.
- [20] 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 *IEEE ICASSP*, 2018, pp. 4839–4843.
- [21] E. McDermott, "A deep generative acoustic model for compositional automatic speech recognition," in *Proceedings of Neural Information Processing Systems (NeurIPS) Workshop: Interpretability and Robustness in Audio, Speech, and Language*, 2018. [Online]. Available: <https://openreview.net/pdf?id=S1fbqB0noQ>
- [22] C. Weng, C. Yu, J. Cui, C. Zhang, and D. Yu, "Minimum bayes risk training of rnn-transducer for end-to-end speech recognition," *ArXiv*, vol. abs/1911.12487, 2019.
- [23] F. Biadsy, M. Ghodsi, and D. Caseiro, "Effectively building tera scale maxent language models incorporating non-linguistic signals," in *INTERSPEECH*, 2017.
- [24] A. Narayanan, A. Misra, K. C. Sim, G. Pundak, A. Tripathi, M. Elfeky, P. Haghani, T. Strohmaier, and M. Bacchiani, "Toward domain-invariant speech recognition via large scale training," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018.
- [25] C. Peyser, T. N. Sainath, and G. Pundak, "Improving proper noun recognition in end-to-end asr by customization of the mwer loss criterion," in *IEEE ICASSP*, 2020, pp. 7789–7793.
- [26] F. Beaufays, A. Sankar, S. Williams, and M. Weintraub, "Learning name pronunciations in automatic speech recognition systems," in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, Nov 2003, pp. 233–240.
- [27] A. Laurent, S. Meignier, T. Merlin, and P. Deleglise, "Acoustics-based phonetic transcription method for proper nouns," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, 01 2010, pp. 2286–2289.
- [28] X. Gonzalvo, S. Tazari, C. Chan, M. Becker, A. Gutkin, and H. Silén, "Recent advances in google real-time hmm-driven unit selection synthesizer," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, N. Morgan, Ed. ISCA, 2016, pp. 2238–2242. [Online]. Available: <https://doi.org/10.21437/Interspeech.2016-264>