Can We Train a Language Model Inside an End-to-End ASR Model? - Investigating Effective Implicit Language Modeling

Zhuo Gong[1], **Daisuke Saito**[1], **Sheng Li**[2], **Hisashi Kawai**[2], **and Minematsu Nobuaki**[1]
[1]The University of Tokyo, Tokyo, Japan
[2]National Institute of Information and Communications Technology, Kyoto, Japan
gongzhuo@gavo.t.u-tokyo.ac.jp

Abstract

Language models (LM) have played crucial roles in automatic speech recognition (ASR) to enhance end-to-end (E2E) ASR systems' performance. There are two categories of approaches: finding better ways to integrate LMs into ASR systems and adapting on LMs to the task domain. This article will start with a reflection of interpolationbased integration methods of E2E ASR's scores and LM's scores. Then we will focus on LM augmentation approaches based on the noisy channel model, which is intrigued by insights obtained from the above reflection. The experiments show that we can enhance an ASR E2E model based on encoder-decoder architecture by pre-training the decoder with text data. This implies the decoder of an E2E model can be treated as an LM and reveals the possibility of enhancing the E2E model without an external LM. Based on those ideas, we proposed the implicit language model canceling method and then did more discussion about the decoder part of an E2E ASR model. The experimental results on the TED-LIUM2 dataset show that our approach achieves a 3.4% relative WER reduction compared with the baseline system, and more analytic experiments provide concrete experimental supports for our assumption.

1 Introduction

In the 1980s, a significant step was achieved by introducing the acoustic model (AM) and language model (LM) into ASR framework. From that time, the methodology of ASR shifted from the more intuitive template-based approach (a straightforward pattern recognition paradigm) towards a more rigorous statistical modeling framework (Juang and Rabiner, 2005). Moreover, those two concepts of AM and LM became the foundation of ASR that we are familiar with nowadays. Relying solely on acoustic observations proved to be insufficient to achieve human-like performance.

With the rapid development of deep learning techniques, many powerful neural network-based systems were invented in the new century. Among them, various end-to-end (E2E) systems become prevail (Battenberg et al., 2017; Chan et al., 2016; Kim et al., 2017; Watanabe et al., 2018; Vaswani et al., 2017), which are benefited from sufficient computing power and data sets. From this stage, E2E ASR becomes the mainstream of modern ASR techniques. We have emphasized the importance of LMs in ASR, but how an independent LM can be utilized in an E2E system? The answers are LM integration, and LM adaptation (Zhao et al., 2019; Shan et al., 2019; Sriram et al., 2018). LM integration increases accuracies of E2E ASR systems in practical indeed. However, intuitively, if an E2E ASR model is powerful enough, there is no need for an extra LM. So, the question becomes how an LM can benefit an E2E ASR system. To be more specific, we need to figure out what happens when we try to integrate the E2E ASR model with an LM and how to adapt an LM to an ASR domain. Furthermore, can we reveal the capability of language modeling in an E2E ASR model?

In this paper, we try to answer those questions theoretically and experimentally. Firstly, we analyzed shallow fusion of LM integration mathematically using LM adaptation framework (McDermott et al., 2019). Then, we proposed an implicit LM canceling method to fully control the language modeling functionality of an E2E ASR system. Finally, we discussed the feasibility that a decoder of an E2E ASR model could be treated as an LM by experiments. To the best of our knowledge, we are the first to analyze the language modeling functionality of the decoder part in an E2E ASR model.

The rest of this paper is structured as follows. Section 2 discusses the most common LM integration approach (shallow fusion) to explore its essence from the perspective of probability models.

Proceedings of the 2nd Workshop on When Creative AI Meets Conversational AI (CAI2), COLING 2022, pages 42–47
October 12–17, 2022.

Then we try to figure out a way to compose LM integration and LM adaptation tasks into a single method in Section 3 and 4. In Section 5, we analyze the result and reveal crucial insights about a decoder's characteristics in an E2E ASR system from several experiments. We conclude the paper in Section 6.

2 Related Work

2.1 LM Integration

In conventional ASR systems, whether or not based on deep learning, an LM is an essential part of the whole system. While in E2E models, an LM is not necessary since they can decode the intermediate representations of input features into a word sequence independently. For an E2E model, it is still beneficial to introduce an LM into the model. an LM is helpful for introducing extra corpora information. The main LM integration approaches in the previous work (Zhao et al., 2019; Shan et al., 2019; Sriram et al., 2018) are referred to as shallow, deep, and cold fusions. In the following section, we focus on investigating the details of shallow fusion.

In Eq. 1 of shallow fusion, $s(y|x)$ is the final score of output tokens based on input features x. The $\beta Penalty(|y|)$ is a penalty item, and it is a function of the output sequence length $|y|$ aiming at suppressing longer candidates. Since a longer sequence tend to produce more meaningless words, such as ah, em, its length should be suppressed. Moreover, α, β are hyper-parameters weighted to determine each item's importance in this equation.

$$\begin{aligned} s(y|x) = &\log(P_{E2E}(y|x)) \\ &+ \alpha \log(P_{LM}(y)) + \beta Penalty(|y|) \end{aligned} \tag{1}$$

where $P_{E2E}(y|x)$ and $P_{LM}(y)$ represent the conditional probabilities of a specific output sequence given input features to an E2E ASR model and an LM.

2.2 LM Adaptation

LM integration is just the first step to introduce LMs into ASR framework. To make an LM fit into a speech domain, we need to introduce LM adaptation. Then, we show how this method can be applied to LM integration analysis.

In previous work (McDermott et al., 2019), the density ratio approach is proposed as a transfer learning method based on Bayes' rule. This previous work studied LM representations in an E2E model. Moreover, this approach makes the following assumptions:

Table 1: List of key variables and their descriptions.

Variable	Description
$P_\phi(W, X)$	The source domain ϕ has some true joint distribution $P_\phi(W, X)$ over text (W) and audio (X)
$P_\tau(W, X)$	The target domain τ has another true joint distribution $P_\tau(W, X)$
$P_\phi(W\|X)$	A source domain E2E model (e.g., RNN-T (Battenberg et al., 2017)) captures $P_\phi(W\|X)$ reasonably well
$P_\phi(W)$ and $P_\tau(W)$	Separately trained LMs (e.g., RNN-LMs) capture $P_\phi(W)$ and $P_\tau(W)$ reasonably well
$p_\phi(X\|W)$ and $p_\tau(X\|W)$	$p_\phi(X\|W)$ as an acoustic model is roughly equal to $p_\tau(X\|W)$, i.e. the two domains are acoustically consistent

According to Bayes's rule, we have:

$$p_\phi(X|W) = p_\phi(X)P_\phi(W|X)/P_\phi(W) \tag{2}$$

Similarly, for the target domain:

$$p_\tau(X|W) = p_\tau(X)P_\tau(W|X)/P_\tau(W) \tag{3}$$

Since these two acoustic models roughly are the same:

$$\hat{P}_\tau(W|X) = k(X)\frac{P_\tau(W)}{P_\phi(W)}P_\phi(W|X) \tag{4}$$

With $k(X) = p_\phi(X)/p_\tau(X)$ shared by all hypotheses W, and density ratio method is named after the ratio $P_\tau(W)/P_\phi(W)$. Based on Eq. 4 we can give the score function of decoding process:

$$\begin{aligned} Score(W|X) = &\log P_\phi(W|X) + \lambda_\tau \log P_\tau(W) \\ &-\lambda_\phi \log P_\phi(W) + \beta \end{aligned} \tag{5}$$

where $Score(W|X)$ is our decoding logits score during beam search.

3 Implicit LM Canceling Method

Inspired by the density ratio approach, we propose to restructure the shallow fusion of Eq.1 in a more general way:

$$
\begin{aligned}
P_{rescoring}(W|X) &= \beta P_{E2E}(W|X)^{1-\lambda} P_{LM}(W)^{\lambda} \\
&= \beta \left(\frac{P_{E2E}(X|W)P_{E2E}(W)}{P_{E2E}(X)} \right)^{1-\lambda} P_{LM}(W)^{\lambda} \\
&= \beta \left(\frac{P_{E2E}(X|W)P_{E2E}(W)P_{LM}(W)^{\lambda/1-\lambda}}{P_{E2E}(X)} \right)^{1-\lambda}
\end{aligned}
$$

where $P_{rescoring}(W|X)$ is the score for a word sequence W given an observation X. $P_{E2E}(W|X)$ stands for our E2E model which gives the probability score of a word sequence given an observation X, and $P_{LM}(W)$ stands for an independent LM. $P_{E2E}(X|W)$ stands for an implicit pronunciation model inside the E2E model, while $P_{E2E}(W)$ represents the implicit LM inside the E2E model which we focus on. Since $P_{E2E}(X)$ is same for different word sequence candidate, this term should be omitted during scoring.

Then we have a probability score,

$$
\begin{aligned}
&\exp(score(W|X)) = \\
&P_{E2E}(X|W)P_{E2E}(W)P_{LM}(W)^{\hat{\lambda}},
\end{aligned} \tag{6}
$$

where $\hat{\lambda} = \lambda/1 - \lambda$

As we can see from Eq.6, the LM of an ASR system including an E2E model and an actual LM is $P_{E2E}(W)P_{LM}(W)^{\hat{\lambda}}$. That means by shallow fusion we can modify the final LM during rescoring. Moreover, it gives us the ability to change the implicit LM in an E2E model.

$$
\begin{aligned}
P(W|X) &= P_{E2E}(W|X)P_{LM}(W)/P_{E2E}(W) \\
&= \left(\frac{P_{E2E}(X|W)P_{E2E}(W)}{P_{E2E}(X)} \right) \frac{P_{LM}(W)}{P_{E2E}(W)} \\
&= \frac{P_{E2E}(X|W)P_{LM}(W)}{P_{E2E}(X)}
\end{aligned} \tag{7}
$$

where $P(W|X)$ is the probability model of the whole E2E ASR system which includes an E2E ASR model and an LM.

It should be noticed that we have no direct control (modify this model) over this implicit LM (as the probability density function $P_{E2E}(W)$) during decoding. One way to take control of the final LM is to cancel the E2E model's implicit LM and replace it with our external LM. This can be achieved by Eq.7. Just like what has been done in the density ratio approach, we train an E2E ASR model and a LM on audio and transcripts of the source domain speech corpus, and then another LM is pre-trained on extra gigantic corpora and fine-tuned on source domain text to approximate the true distribution of source domain. During decoding, the score function is Eq.8.

$$
\begin{aligned}
score(W|X) &= \log P_{E2E}(W|X) \\
&+ \log P_{LM}(W) - \log P_{E2E}(W)
\end{aligned} \tag{8}
$$

where $score(W|X)$ is the score for beam searching

We propose it as implicit LM canceling method. This kind of approach has no requirements for the E2E model (e.g., RNN-T in density ratio approach) and does not require hyper-parameters to tune the importance of two LMs. Thus, we can build an experimental ASR system based on the state-of-the-art transformer-encoder decoder model plus CTC loss (Kim et al., 2017) function in Fig. 1. The detailed settings can be found in Section 5.2.

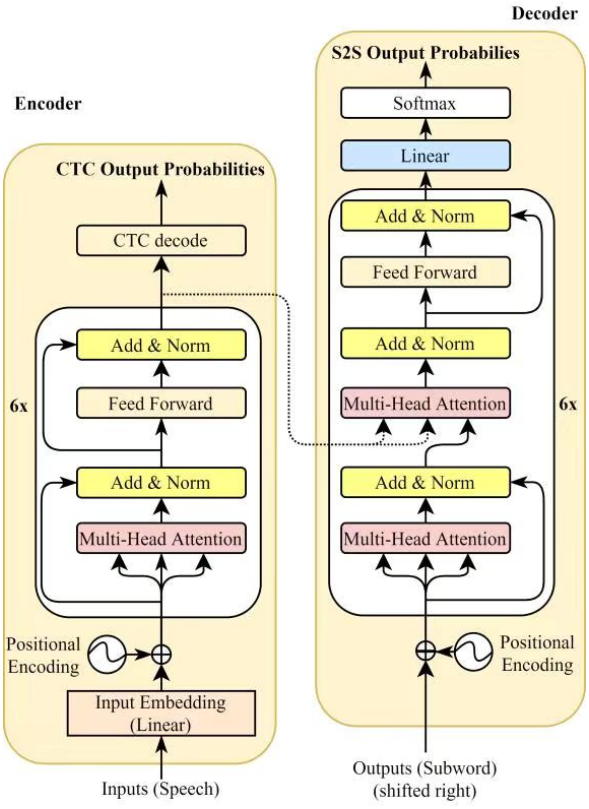

Figure 1: Transformer-based E2E ASR model jointly trained with CTC loss.

4 The Implicit LM of An E2E ASR System

We can check the assumption that the decoder of an E2E model can be treated as an LM from two aspects:

1. Is there a structure supporting the function of an LM in the E2E model?

2. Does it behave like an LM?

The decoder of this E2E model should potentially be an LM because it is an auto-regression model, just like a normal language model, which receives a token sequence and outputs the next token. It has transformer layers for memorizing information from the training set. And from Fig. 1 the multi-head attention layers which receive a word sequence and hidden states from the encoder are built relatively independent. So, we can assume the layers which did not receive hidden states directly from the encoder would be dominated by outputs (subword). Those characteristics above fulfill the first aspect. We can check the second statement by sampling token sequences in an auto-regression manner or calculate its perplexity about an LM's behaviors. Moreover, the above clarification leads to our new proposal: the decoder of an E2E model can be treated as an LM and be pre-trained on text data before E2E training to improve the E2E model. We will validate this in the following experiments.

5 Experiments

5.1 Task Descriptions

The experiments contain two parts: to validate our LM canceling method's performance and test a decoder's potential as an LM with more experiments.

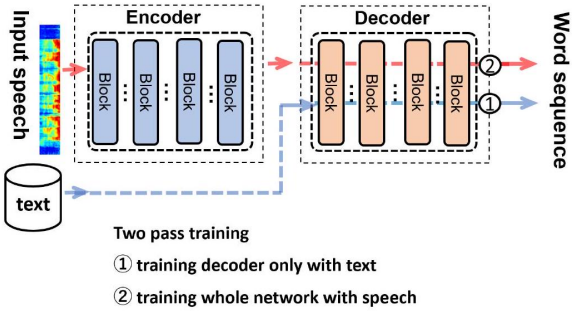

Figure 2: The Workflow of training implicit language model for E2E ASR model.

1. The first part is the same as a standard shallow fusion method. We calculate output logits of three models and apply Eq.7 during beam searching in the testing stage.

2. The second part in Fig. 2 is to train a decoder in an E2E ASR model as an LM. We implement this idea in a straightforward way. We set the intermediate vectors from the encoder to zeros and feed the decoder text corpus to output embedding in Fig. 1 just as if we are training a norm LM while completely omit the encoder. And then, several different experiments are conducted based on a decoder we trained in this manner. The decoders' perplexities and word error rate (WER) of the E2E ASR model with a different decoder are calculated as results.

5.2 Experimental Settings

We adopt a Transformer-based ASR system comprised of 6 encoder blocks and 6 decoder blocks with the feed-forward inner dimension of 2048, the model dimension of 256, and the attention head number 4, which are unchanged in all experiments. The input features were 240-dimensional log Mel-filterbank energy features (80-dim static, $+\Delta$, and $+\Delta\Delta$). The feature is extracted with a 10-ms frameshift of a 25-ms window. Each feature was mean- and variance-normalized per speaker, and every four frames were spliced (three left, one current, and zero right). The low and high cutoff frequencies were set to 20 Hz and 8,000 Hz, respectively. Speed perturbation was not used in the fine-tuning stage. We then subsampled the input features every three frames. The model was jointly trained with CTC (weight $\alpha = 0.2$). The "noam" optimizer was used with 25,000 warmup steps and an initial learning rate of 5. The model was trained with ESPnet toolkit (Watanabe et al., 2018) using batch-size 32 for 30 epochs on an 11-GB GTX1080 TI GPU.

The experiment is conducted on TED-LIUM2 (Rousseau et al., 2012), and the LMs are trained on text data offered by this corpus. Moreover, the LMs are four-layer transformer models.

5.3 Results and Discussions

The performances of the proposed LM canceling method are shown in Table 2 (+Transcripts LM means shallow fusion of the baseline model and

Table 2: Word Error Rate (WER) Results of E2E ASR with different LM settings

E2E baseline	+Transcripts LM (A)	+Text LM (B)	-A+B
11.7	11.6	10.5	11.3

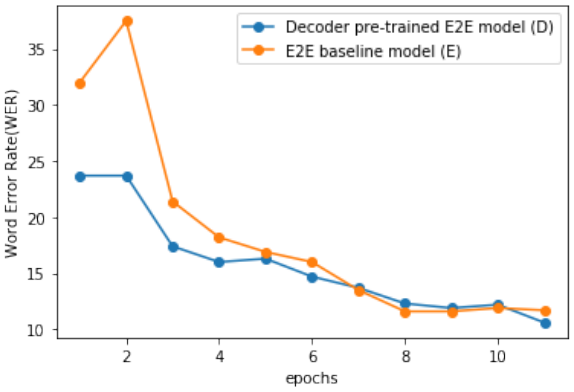

Figure 3: WER of decoder pretrained E2E model and E2E model trained from scratch.

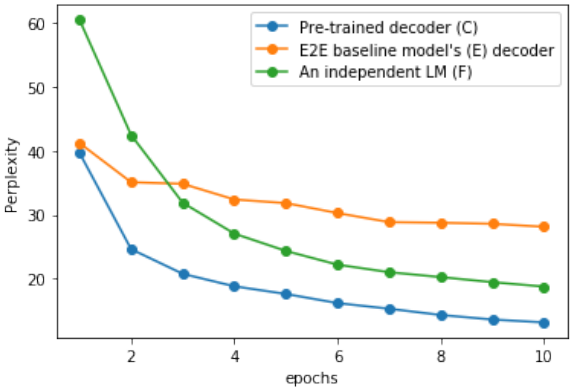

Figure 4: Perplexity of the decoder trained on transcripts, the E2E baseline model trained on paired audio and transcripts and an independent LM trained on transcripts.

an LM trained on transcripts from the baseline corpus; +Text LM means shallow fusion of baseline model and an LM trained on extra text corpus; -A+B means applying the method in Eq.8). We have to admit that results are not good as expected. One main reason is that we have made a strong assumption that the implicit LM $P_{E2E}(W)$ of an E2E model can be represented by an independent explicit LM. In the following section, we investigate why this assumption works not well.

To check the potential that a decoder can be treated as an LM further, we did more analytic experiments. We pre-trained the decoder (C) by feeding it transcripts from the source domain and set the hidden states to be zero vectors. Those hidden states are supposed to be passed from the

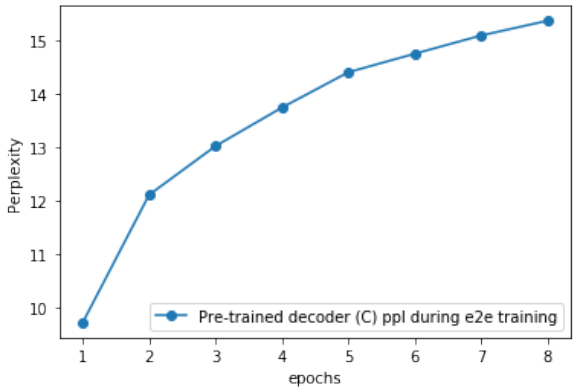

Figure 5: Perplexity of pre-trained E2E model during E2E training.

encoder to the decoder of the same E2E model. This method can ensure that no acoustic related weights will change during training of the decoder. After this decoder pre-training process, the E2E model (D) with the pre-trained decoder will be trained in the speech corpus. A baseline model (E) of the same structure as the previous one will be trained from scratch.

Several experiments are also conducted on the corpus TED-LIUM2, and all the LM training (including an independent LM (F) and the decoder (D)) are done on transcripts data of the speech corpus. All the E2E models are built on the same structure of transformer-based sequence-to-sequence model.

The results in Fig. 3 show that pre-training the decoder (C) as a language model does improve the performance of this E2E model (D). Fig. 4 shows the perplexity results for the decoder (C) and the LM (F) trained on transcripts and the E2E model (E) trained on the same transcripts with paired audio data.

As we can see in Fig. 4, the decoder's (C) perplexity effectively decreased during training and even decreased more rapidly than the LM (F), which may be related to more layers in the decoder. This can prove that the decoder (C) can be trained like an LM effectively. Moreover, the perplexity of the E2E model (E) trained from scratch decreased slowly. This phenomenon can explain why the im-

plicit LM in the E2E model (in Table 2) should not be canceled by an external LM (A) trained even on the same transcripts. Because the LM performances of them are not even close. Fig. 5 gives the perplexity tendency of the decoder (C) in E2E training shown in Fig. 3.

The most interesting observation from it is that even the whole E2E model (D) becomes more accurate during training, but the decoder (C) part of it becomes worse as an LM, which implies the E2E training may harm the implicitly language modeling in the decoder (C). This phenomenon alerts all of the developers working on E2E models, and we will make an in-depth investigation to cope with it.

6 Conclusions

This article reflected why we introduced LMs into E2E ASR systems and discussed how LM integration benefits an E2E ASR system by generalizing shallow fusion by probability density function inspired by LM adaptation in ASR. In the general version of shallow fusion, insights about whether there is an implicit LM and how to modify it are obtained. This work reveals the decoder's potential to be trained and improve E2E models by training the decoder independently without external LMs. Moreover, we proposed the implicit LM canceling method. In the ordinary design of this transformer-based system, the decoder needs hidden states from an encoder, but we set these hidden states to zeros vectors to avoid acoustic feature-related weights changing in the decoder during pre-training. In the future, we will find a more sophisticated way to pre-train the decoder, alter the structure, modify the loss function, or change the training schedule. Moreover, we will try to figure out a way to suppress the degeneration phenomenon of the decoder's LM function (C) during E2E training.

In the next step, we plan to find a more sophisticated way to pre-train the decoder, alter the structure, modify the loss function, as well as change the training schedule. Moreover, we will try to figure out a way to suppress the degeneration phenomenon of the decoder's LM function (C) during E2E training.

References

Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu. 2017. Exploring neural transducers for end-to-end speech recognition. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 206–213.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

Biing-Hwang Juang and Lawrence R Rabiner. 2005. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, 1:67.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839.

Erik McDermott, Hasim Sak, and Ehsan Variani. 2019. A density ratio approach to language model fusion in end-to-end automatic speech recognition. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 434–441.

Anthony Rousseau, Paul Deléglise, and Y. Estève. 2012. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*.

Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie. 2019. Component fusion: Learning replaceable language model component for end-to-end speech recognition system. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5361–5635.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2018. Cold fusion: Training seq2seq models together with language models. *ArXiv*, abs/1708.06426.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Yalta, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. Espnet: End-to-end speech processing toolkit. *ArXiv*, abs/1804.00015.

Ding Zhao, Tara N. Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. 2019. Shallow-fusion end-to-end contextual biasing. In *INTERSPEECH*.