

# RESCOREBERT: DISCRIMINATIVE SPEECH RECOGNITION RESCORING WITH BERT

Liyan Xu<sup>1,2</sup>

Yile Gu<sup>1</sup>

Jari Kolehmainen<sup>1</sup>

Haidar Khan<sup>1</sup>

Ankur Gandhe<sup>1</sup>

Ariya Rastrow<sup>1</sup>

Andreas Stolcke<sup>1</sup>

Ivan Bulyko<sup>1</sup>

<sup>1</sup>Amazon Alexa AI, USA

<sup>2</sup>Emory University, USA

## ABSTRACT

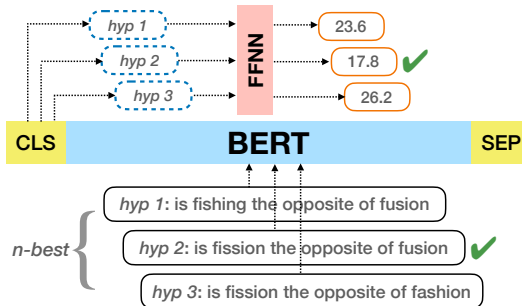
Second-pass rescoring is an important component in automatic speech recognition (ASR) systems that is used to improve the outputs from a first-pass decoder by implementing a lattice rescoring or  $n$ -best re-ranking. While pretraining with a masked language model (MLM) objective has received great success in various natural language understanding (NLU) tasks, it has not gained traction as a rescoring model for ASR. Specifically, training a bidirectional model like BERT on a discriminative objective such as minimum WER (MWER) has not been explored. Here we show how to train a BERT-based rescoring model with MWER loss, to incorporate the improvements of a discriminative loss into fine-tuning of deep bidirectional pretrained models for ASR. Specifically, we propose a fusion strategy that incorporates the MLM into the discriminative training process to effectively distill knowledge from a pretrained model. We further propose an alternative discriminative loss. This approach, which we call RescoreBERT, reduces WER by 6.6%/3.4% relative on the LibriSpeech clean/other test sets over a BERT baseline without discriminative objective. We also evaluate our method on an internal dataset from a conversational agent and find that it reduces both latency and WER (by 3 to 8% relative) over an LSTM rescoring model.

**Index Terms**— masked language model, BERT, second-pass rescoring, pretrained model, minimum WER training

## 1. INTRODUCTION

The two-pass paradigm has been widely adopted in state-of-the-art ASR systems [1, 2, 3, 4, 5], where the first pass generates  $n$ -best hypotheses, and the second pass reranks them. For the second-pass rescoring models, discriminative training with MWER (minimum WER) objective is typically applied [3, 6, 7, 4] to improve performance, such that the model learns to prefer hypotheses with the lowest WER.

Previous work with discriminative training uses causal language models (CLMs), such as LSTMs or Transformer LMs. While pretrained masked language models (MLMs) such as BERT [8] have been highly successful on various natural language understanding (NLU) tasks, they have not been widely applied in second-pass ASR rescoring. Meanwhile, recent studies have shown promising results using BERT in several rescoring studies [9, 10, 11], as BERT is pretrained with large corpora and encodes the full hypothesis context using a deep bidirectional model architecture. In particular, previous work [9] shows that deep bidirectional Transformers, such as BERT, can outperform their unidirectional counterparts (either forward text, backward text, or the two models combined). Another paper [10]



**Fig. 1:** Illustration of discriminative scoring with BERT on  $n$ -best hypotheses. Each hypothesis is individually encoded by BERT and represented by CLS; it is then followed by a feed-forward NN to compute a sentence-level second-pass LM score. The scores are then interpolated with first pass scores for reranking.

shows that a pretrained BERT model that is then fine-tuned on LibriSpeech data can outperform BERT trained from scratch on LibriSpeech clean/other test sets by 4.4%/3.2% WER relative, demonstrating the effectiveness of pretraining in BERT. They also show that BERT can outperform GPT [12] with comparable model size and pretraining data, which the authors argue is due to the bidirectional nature of BERT.

In this work, we propose a method to train BERT-style rescoring models with a discriminative objective, to leverage the aforementioned benefits from both approaches. Typically, pseudo log-likelihood (PLL) [9, 10]—the sum of the negative log-likelihoods of each individual token given the bidirectional context—is used to rescore  $n$ -best output to improve WER, a computationally expensive process, particularly for longer sentences. For discriminative training, this issue is exacerbated as the PLL computation needs to be repeated for each hypothesis individually. The previous work [10] solves this issue by distilling the PLL into a single score prediction at the start-of-sentence token (CLS). In this work, illustrated in Figure 1, we extend this approach and use the score from the CLS representation to perform discriminative training, as discussed in Section 2.2.2, with either MWER loss or a novel discriminative training loss dubbed matching word error distribution (MWED), described in Section 2.2.1. Finally, in Section 2.2.3 we propose a fusion strategy that incorporates the MLM into the discriminative training process, giving further improvements.

We name the aforementioned approach RescoreBERT, and evaluate it on four datasets covering multiple domains and locales. Results show that discriminative training significantly improves upon non-discriminative BERT rescoring, on all test sets. The new MWED training loss is found to be a strong alternative to MWER. The results also show that the fusion approach for incorporating MLM into discriminative training can further improve WER. Lastly,

<sup>2</sup>Work done as an applied scientist intern at Amazon Alexa.

to achieve lower latency for streaming applications, we develop a method to further distill the model while maintaining WER improvements.

## 2. APPROACH

### 2.1. BERT Without Discriminative Training

In this section, we review previous work on BERT rescoring models that are not trained with discriminative objective.

#### 2.1.1. Pseudo log-likelihood (PLL)

Let  $E = (e_1, \dots, e_{|E|})$  be a sequence of tokens. Similar to the log-likelihood of a sequence as is commonly used in CLM scoring [13, 14, 15], the pseudo log-likelihood (PLL) of a sequence using an MLM, first introduced by [9], is defined as:

$$\text{PLL}(E) = - \sum_{t=1}^{|E|} \log P(e_t | E_{\setminus t}) \quad (1)$$

where  $E_{\setminus t} = (\dots, e_{t-1}, [\text{MASK}], e_{t+1}, \dots)$  is the sequence whose corresponding position is replaced by the [MASK] token used in MLM pretraining. The PLL is thus the sum of the negative log-likelihoods of each token given the bidirectional context, with lower scores indicating more probable sequences.

#### 2.1.2. MLM distillation (MD)

Although the PLL demonstrates good performance for second-pass rescoring [9], it is not computationally efficient:  $|E|$  additional sequences masking every position need to be generated and encoded by BERT, thus the computation due to PLL is on the order of  $|E|$  times that of an autoregressive Transformer model of similar size. Following [10], one can “distill” the PLL calculation into a single utterance-level score using the CLS representation, such that the model is able to approximate PLL, while eliminating the need for masking  $|E|$  times, as well as the large vocabulary softmax in  $P(e_t | E_{\setminus t})$ , thereby reducing computation significantly. As shown in the equations below, each sentence  $E_i$  is individually encoded by BERT, represented by the hidden state of the CLS token in the last Transformers layer, denoted by  $g_i$ . An additional layer is stacked on top of the CLS hidden states to produce the score  $s_i^l$  for  $E_i$ . The distillation is achieved by training the model to mimic PLL scores using mean squared error (MSE) regression loss:

$$g_i = \text{BERT}^{\text{CLS}}(E_i) \quad (2)$$

$$s_i^l = \text{FFNN}(g_i) \quad (3)$$

$$\mathcal{L}_{\text{MD}} = |s_i^l - \text{PLL}(E_i)|^2 \quad (4)$$

FFNN denotes the learnable feed-forward neural network,  $s_i^l$  is the predicted PLL approximation, and  $\text{PLL}(E_i)$  is precomputed offline using Eq. (1). Note that the PLL can be computed by a larger *teacher* model.

### 2.2. BERT With Discriminative Training

We now propose methods for training BERT with discriminative objective functions. For any utterance, let  $\vec{E} = \{E_1, \dots, E_n\}$  be the  $n$ -best hypotheses obtained from beam search in the first-pass decoder. For any  $E_i \in \vec{E}$ , let  $s_i^a$  be its given score from the first pass, and

$s_i^l$  be the score from the second pass (same as Eq. (3)), with lower scores for more likely hypotheses for both; let  $\epsilon_i$  be its number of word errors (edit distance) from the ground truth transcription. Following the common theme of second pass rescoring approaches, the final score  $s_i$  is the linear combination of the first-pass and second-pass scores:

$$s_i = s_i^a + \beta \cdot s_i^l, \quad (5)$$

where  $\beta$  is the hyperparameter controlling the second-pass contribution.  $s_i$  is then used to compute discriminative loss, as defined next.

#### 2.2.1. Discriminative loss function

We explore two discriminative loss functions: MWER and MWED.

**MWER (Minimum word error rate):** A standard discriminative loss function for ASR rescoring is MWER [7]. The training minimizes the expected number of word errors for the  $n$ -best hypotheses:

$$P_i = \frac{e^{-s_i}}{\sum_{j=1}^n e^{-s_j}} \quad (6)$$

$$\bar{\epsilon}_H = \frac{1}{n} \sum_{i=1}^n \epsilon_i \quad (7)$$

$$\mathcal{L}_{\text{MWER}} = \sum_{i=1}^n P_i \cdot (\epsilon_i - \bar{\epsilon}_H). \quad (8)$$

$P_i$  is the posterior probability of each hypothesis, normalized over the hypotheses list from the first pass, such that higher probabilities indicate preferred hypotheses.  $s_i$  is the final score of the hypothesis as in Eq. (5). The MWER loss  $\mathcal{L}_{\text{MWER}}$  represents the expected number of relative word errors, with  $\bar{\epsilon}_H$  being the averaged word errors across the  $n$ -best list, which does not change the optima but helps to reduce the variance.

**MWED (Matching word error distribution):** MWED is a new loss function proposed here. Its goal is to mimic the distribution of  $n$ -best word errors through the predicted scores. As a result, the ranking of final scores should ideally be exactly the same as ranking by the word errors, which could potentially lead to better score interpolation at evaluation. By contrast, the model trained with the existing MWER loss as in Eq (8) picks the best hypothesis discriminatively, such that the full probability mass should be assigned to the one with minimum word errors in the ideal case.

The MWED loss is proposed as the following:

$$d_i^e = \frac{e^{\epsilon_i}}{\sum_{j=1}^n e^{\epsilon_j}} \quad (9)$$

$$d_i^s = \frac{e^{s_i/T}}{\sum_{j=1}^n e^{s_j/T}} \quad (10)$$

$$\mathcal{L}_{\text{MWED}} = - \sum_{i=1}^n d_i^e \log d_i^s \quad (11)$$

$d_i^e$  and  $d_i^s$  represent the relative distribution of word errors and predicted scores over the  $n$ -best list.  $\mathcal{L}_{\text{MWED}}$  is the cross-entropy from scores to word errors, equivalent to optimizing the Kullback–Leibler divergence between the two distributions. Due to that  $s_i$  contains  $s_i^a$  which is fixed, to stabilize the match of the two distributions, we add a hyperparameter  $T$  to rescale the distribution mass of  $s_i$ . In practice, we found that  $T = \sum_{i=1}^n s_i / \sum_{i=1}^n \epsilon_i$  can yield good performance.

### 2.2.2. Training with discriminative loss only

Training BERT naively with discriminative loss using word-level scores, as done in [4, 6], requires computation of Eq. (1) for every hypothesis and is prohibitively expensive during both training and inference. Instead, it can be fine-tuned such that the sentence-level score from the CLS representation (as in Eq. (3)) minimizes the discriminative loss  $\mathcal{L}_{\text{MWER}}$  or  $\mathcal{L}_{\text{MWED}}$  defined earlier.

In Section 4, we show results using this approach with  $\mathcal{L}_{\text{MWER}}$ , labeled “*MWER Only*”, where we perform MWER training on a pre-trained BERT with domain adaptation.

### 2.2.3. Training with combined MLM and discriminative loss

We propose a fusion strategy to incorporate MLM distillation into discriminative training. It is accomplished by making two modifications to the approach in Section 2.2.2, where only discriminative loss is applied.

First, we apply a pretraining step using MD alone on a large-scale text-only corpus, so that the discriminative training can be warm-started from a better initialization point. Unlike MWER training, MD only needs text-only data and their PLL scores computed by a *teacher* model. Therefore, the distillation itself can be trained on much more data than the *n*-best hypotheses used in MWER training.

Second, we introduce the new loss  $\mathcal{L}$  to replace  $\mathcal{L}_{\text{MWER}}$  or  $\mathcal{L}_{\text{MWED}}$  in the discriminative training step:

$$\mathcal{L} = \mathcal{L}_{\text{Discriminative}} + \lambda \cdot \sum_{i=1}^n \mathcal{L}_{\text{MD}}(E_i), \quad (12)$$

where  $\mathcal{L}_{\text{Discriminative}}$  is the discriminative loss that can be either  $\mathcal{L}_{\text{MWER}}$  or  $\mathcal{L}_{\text{MWED}}$ .  $\mathcal{L}_{\text{MD}}$  is similar to cross-entropy regularization added to MWER loss in [6], and controlled by the hyperparameter  $\lambda$ . It is found that MD pretraining is more important than adding additional MD loss. On top of MD pretraining, having an additional step of adding MD loss yields less than 0.5% relative improvement from all experiments.

In Section 4, we show the results using this approach with  $\mathcal{L}_{\text{MWER}}$  and  $\mathcal{L}_{\text{MWED}}$ , which are named *MD-MWER* and *MD-MWED*, respectively.

## 3. EXPERIMENTS

### 3.1. Datasets

We evaluated our approach on four datasets in multiple domains and locales to test its general applicability, including one public dataset LibriSpeech [16] and three internal dataset based on a conversational agent (one for Information (Info) domain in English (en), and two for Info and Navigation (Nav) domains in Japanese (ja)).

For LibriSpeech, an LAS model [17] is adopted as the first-pass decoder, and we use the same decoded 100-best hypotheses of the dev and test set along with their first-pass scores used by [9, 10]. Since there is no dedicated training set provided for MWER, we combine the decoded hypotheses from both dev-clean and dev-other as the MWER training set, and randomly hold out 10% utterances as the MWER dev set. The resulting training/dev set has 5011/556 utterances, with up to 100 hypotheses per utterance. For MLM distillation, we sample 4 million utterances from the in-domain text corpus provided by LibriSpeech as the training set, similar to [10].

All internal datasets consist of de-identified live user interactions with a conversational agent, decoded by a RNN-T model [18] for en or a hybrid HMM model for ja. For Info (en),  $\sim 100/5$  hours of

utterances are used as the MWER training/dev set; the test set has  $\sim 2$  hours of long-tail info utterances. For Info and Nav in ja, we use a single MWER training/dev set that consists of  $\sim 220/10$  hours of utterances in multiple domains including Info and Nav domains. The test set has  $\sim 1/3$  hours of utterances for Info/Nav respectively. For MD, 4 million in-domain utterances sampled from user interactions are used as training set.

### 3.2. Implementation

For LibriSpeech, we use the uncased BERT<sub>Base</sub> for our experiments to enable direct comparison with previous work. For internal datasets, we use an in-house multilingual BERT of  $\sim 170\text{M}$  parameters (excluding embedding size) with 16 layers and 1024 hidden size, supporting both en and ja locales. In addition, our final MWER setting also includes two smaller BERT models of  $\sim 17\text{M}/5\text{M}$  parameters distilled [19] from the 170M BERT model, both with only 4 layers and 768/320 hidden size respectively. All BERT models are implemented in PyTorch and pretrained on public data. We limit the maximum sequence length to 128 for LibriSpeech, and 32 for others; longer utterances will be truncated. The  $\lambda$  parameter in Eq. (12) is set to  $10^{-4}$ . We found that  $\lambda$  between  $10^{-4}$  and  $10^{-3}$  generally yield same performance.

Before we conducted any training, we first performed domain adaptation for BERT, as BERT contains general world knowledge from pretraining but not necessarily task-specific knowledge. Therefore, we take the pretrained BERT and further train with the MLM objective on the in-domain corpus. For LibriSpeech, we train 400K steps on the provided text corpus, similar to [10]. For internal datasets, we train 200K steps on the in-domain transcriptions for each of the en and ja locales. Each step has an effective batch size of 256 utterances for both of these cases.

### 3.3. Baseline and Evaluation Protocol

For LibriSpeech, we use the results of the MLM distillation (MD) as the baseline, which can be seen as our re-implementation of the “sentence-level fine-tuning” results from [10], which has the same low-latency scoring as our MWER setting. We also provide the results of high-latency PLL scores for comparison. WER is used as the evaluation metric, and the optimal interpolation weight  $\beta$  in Eq. (5) is linearly searched on the dev set.

For internal datasets, we use the LM scoring from a 2-layer LSTM trained with noise contrastive estimation (NCE) [20] as the baseline, which is often employed in industrial settings for streaming applications. The LSTM is trained on the same data used for domain adaptation for BERT. New scores from BERT replace the existing LSTM scores, and the optimal weight is searched on the dev set as well. We report the relative improvements in WER for en, and in CER (character error rate) for ja.

## 4. RESULTS AND ANALYSIS

### 4.1. Comparing Different BERT Rescoring Approaches

Tables 1(a) and (b) show the evaluation results on both LibriSpeech and internal datasets. Here, *PLL* denotes the approach in Section 2.1.1 which is computationally expensive; *MD* denotes the approach in Section 2.1.2 that distills the PLL score; *MWER Only* denotes the approach described in Section 2.2.2 that trains BERT with the MWER objective only; *MD-MWER* and *MD-MWED* denote the approaches described in Section 2.2.3 that incorporate MLM into

**Table 1:** Evaluation results on the test partitions of all datasets. Details of the baseline and evaluation protocol are described in Section 3.3. The case of “MWED only” is not included, and the relative difference between it and “MWER only” is within 1% for all tests.

(a) WER on the two test sets of LibriSpeech using BERT<sub>Base</sub>. Numbers inside parentheses are relative improvements compared to the baseline.

	Test-Clean	Test-Other
First-Pass PLL	7.26	20.37
PLL	4.54	16.08
Baseline (MD)	4.67	16.15
MWER Only	4.82 (-3.2%)	16.35 (-1.2%)
MD-MWER	4.42 (5.3%)	15.87 (1.7%)
MD-MWED	<b>4.36 (6.6%)</b>	<b>15.60 (3.4%)</b>

(b) Relative improvements of WER (for en) and CER (for ja) on three internal datasets, using the in-house 170M BERT model.

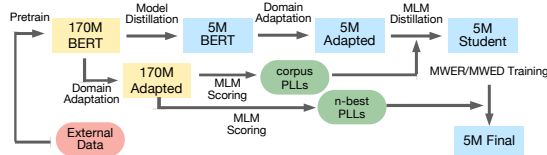
	Info (en)	Info (ja)	Nav (ja)
LSTM	Baseline	Baseline	Baseline
MD	2.6%	3.7%	5.6%
MWER Only	5.3%	11.8%	11.2%
MD-MWER	4.0%	<b>12.0%</b>	<b>12.7%</b>
MD-MWED	<b>6.6%</b>	10.4%	12.2%

discriminative training with MWER and MWED loss functions (as in Section 2.2.1), respectively.

Based on these results, we can make three observations. First, discriminative training significantly improves upon non-discriminative BERT rescoring (MD) across all test sets: 6.6%/3.4% WER relative improvement on LibriSpeech and 4%/8.3%/7.1% relative WER reduction on internal datasets. What is particularly striking is that on LibriSpeech, because of discriminative training, both MD-MWER and MD-MWED now outperform the much more computationally expensive PLL approach. Second, the fusion approach of incorporating MLM in discriminative training improves on all test sets. The effect is particularly strong on the LibriSpeech test sets, where MWER Only would actually perform worse than MD with 3.2%/1.2% relative WER degradations, but where the fusion approach now gives 5.3%/1.7% relative improvement over MD. Third, to compare the new loss function MWED with the existing MWER, MD-MWED achieves better performance on both LibriSpeech (1.3% and 1.7% relative) and Info (en) (2.6% relative) over MD-MWER, but worse than MD-MWER on both ja test sets. This result shows that MWED can be a strong alternative loss in the MWER training, and the final performance can be dataset-specific. One potential explanation for MWED being less effective for ja is that the CER distribution is spikier than the WER distribution, resulting in less stable gradients from the relative entropy in Eq. (11).

#### 4.2. Low Latency for Streaming Applications

A main challenge when applying second-pass rescoring for streaming applications is to keep user-perceived latency low while obtain-



**Fig. 2:** Diagram of training the distilled 5M-parameter BERT model with the fusion strategy described in Section 2.2.3; the 170M BERT model is the *teacher* for calculating PLL scores.

**Table 2:** Performance and latency study of our approach using BERT of different parameter sizes on three internal datasets.

(a) Relative improvements over LSTM (4M) on three internal datasets, using the best setting (MD-MWER/MWED) according to Table 1b for each dataset.

BERT	Info (en)	Info (ja)	Nav (ja)
170M	<b>6.6%</b>	<b>12.0%</b>	<b>12.7%</b>
17M	3.5%	9.1%	9.4%
5M	3.1%	7.8%	7.8%

(b) Averaged latency (in ms) of each batch using 2 threads on a CPU, with a batch size of 5 hypotheses. SL: input sequence/hypothesis length. Parentheses indicate relative latency compared to LSTM.

SL	LSTM (4M)	BERT (5M)	BERT (17M)	BERT (170M)
16	9.7	<b>7.6 (78%)</b>	17.5 (180%)	180 (1.8k%)
32	18.7	<b>11.0 (59%)</b>	26.3 (141%)	270 (1.4k%)

ing the accuracy gains. We next examine how to further distill the model to reduce its compute footprint and hence, achieve even lower latency. We focus on the internal datasets for this task. As described in Section 3.2, in addition to the 170M BERT model, we have 17M and 5M BERT models distilled [19] from it. To achieve low latency, we can perform MD-MWER or MD-MWED starting from this smaller model, except that to maintain WER gains, we use the 170M BERT model to compute PLL scores in MD training, as well as in the final combined loss. This training process is illustrated in Figure 2.

Table 2a shows the relative improvements using BERT models with 170M, 17M, and 5M parameters described in Section 3.2 with the best settings for each dataset. Smaller BERT is expected to yield less improvement; nevertheless, the degradation is within a relatively small margin even for the much smaller 5M version that has only  $\sim 3\%$  the parameters of the 170M model, with still nearly 8% improvement on two ja datasets and 3+ on Info (en).

Table 2b shows the latency comparison among three BERT models and the NCE-based LSTM of 4M parameters, using the PyTorch benchmarking tool under direct model inference in Python using 2 threads on a CPU. 5M BERT is shown to be faster than LSTM, while 17M BERT is slower but also appears comparable. Overall, Table 2 shows that our proposed approach can be a superior substitute for LSTM scoring in deployed systems. In particular, from Table 2a and 2b, 5M BERT significantly outperforms a similar-sized 4M LSTM, both in WER (by 3.1%/7.8%/7.8%) and in latency (by 22%/41%).

## 5. CONCLUSION

We have proposed a method to train a BERT rescoring model with discriminative objective functions. We show that discriminative training can significantly improve BERT rescoring on a variety of datasets: 6.6%/3.4% relative WER improvement on LibriSpeech and 4%/8.3%/7.1% relative WER improvement on internal voice assistant datasets. The proposed fusion strategy to incorporate MLM into discriminative training is found to further reduce WER. We also propose a new discriminative loss MWED that is a strong alternative to the standard MWER loss, yielding 1.3% and 1.7% relative WER improvement over MWER loss on LibriSpeech and 2.6% relative improvement on one internal dataset. Lastly, we show how to further distill the model to achieve even lower latency for streaming applications while preserving WER improvements. We achieve 3-8% relative WER improvement and lower latency compared to a baseline LSTM model on internal data.

## 6. REFERENCES

- [1] Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu, “Deliberation networks: Sequence generation beyond one-pass decoding,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.
- [2] Tara N. Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, Ian McGraw, and Chung-Cheng Chiu, “Two-Pass End-to-End Speech Recognition,” in *Proc. Interspeech*, 2019, pp. 2773–2777.
- [3] Ke Hu, Tara N Sainath, Ruoming Pang, and Rohit Prabhavalkar, “Deliberation model based two-pass end-to-end speech recognition,” in *Proc. IEEE ICASSP*, 2020, pp. 7799–7803.
- [4] Ankur Gandhe and Ariya Rastrow, “Audio-attention discriminative language model for asr rescoring,” in *Proc. IEEE ICASSP*, 2020, pp. 7944–7948.
- [5] Ke Hu, Ruoming Pang, Tara N Sainath, and Trevor Strohman, “Transformer based deliberation for two-pass speech recognition,” in *Proc. IEEE Spoken Language Technology Workshop*, 2021, pp. 68–74.
- [6] Takaaki Hori, Chiori Hori, Shinji Watanabe, and J. Hershey, “Minimum word error training of long short-term memory recurrent neural network language models for speech recognition,” in *Proc. IEEE ICASSP*, 2016, pp. 5990–5994.
- [7] Rohit Prabhavalkar, Tara N Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjali Kannan, “Minimum word error rate training for attention-based sequence-to-sequence models,” in *Proc. IEEE ICASSP*, 2018, pp. 4839–4843.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, Minneapolis, Minnesota, June 2019, pp. 4171–4186.
- [9] Joonbo Shin, Yoonhyung Lee, and Kyomin Jung, “Effective sentence scoring method using BERT for speech recognition,” in *Asian Conference on Machine Learning*. PMLR, 2019, pp. 1081–1093.
- [10] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff, “Masked language model scoring,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 2699–2712, Association for Computational Linguistics.
- [11] Shih-Hsuan Chiu and Berlin Chen, “Innovative BERT-based reranking language models for speech recognition,” in *Proc. IEEE Spoken Language Technology Workshop*, 2021, pp. 266–271.
- [12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, “Language models are unsupervised multitask learners,” OpenAI Blog 1(8), 2019.
- [13] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huihui Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “On using monolingual corpora in neural machine translation,” *arXiv preprint arXiv:1503.03535*, 2015.
- [14] Felix Stahlberg, James Cross, and Veselin Stoyanov, “Simple fusion: Return of the language model,” in *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium, Oct. 2018, pp. 204–211, Association for Computational Linguistics.
- [15] Xie Chen, Anton Ragni, Xunying Liu, and Mark JF Gales, “Investigating bidirectional recurrent neural network language models for speech recognition,” in *Proc. Interspeech*. International Speech Communication Association (ISCA), 2017, pp. 269–273.
- [16] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “LibriSpeech: an ASR corpus based on public domain audio books,” in *Proc. IEEE ICASSP*, 2015, pp. 5206–5210.
- [17] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, “Listen, Attend and Spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. IEEE ICASSP*, 2016, pp. 4960–4964.
- [18] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziq Alvarez, Ding Zhao, David Rybach, Anjali Kannan, Yonghui Wu, Ruoming Pang, et al., “Streaming end-to-end speech recognition for mobile devices,” in *Proc. IEEE ICASSP*, 2019, pp. 6381–6385.
- [19] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean, “Distilling the knowledge in a neural network,” in *NIPS Deep Learning and Representation Learning Workshop*, 2015.
- [20] Anirudh Raju, Denis Filimonov, Gautam Tiwari, Guitang Lan, and Ariya Rastrow, “Scalable multi corpora neural language models for ASR,” in *Proc. Interspeech*, Gernot Kubin and Zdravko Kacic, Eds. 2019, pp. 3910–3914, ISCA.