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] [2] [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] [5] [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] 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,
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 $E = (e_1, ..., e_{|E|})$ be a sequence of tokens. Similar to the log-likelihood of a sequence as 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:

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

(1)

where $E_{\setminus t} = (e_{t-1}, \text{[MASK]}, e_{t+1}, ...)$ 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 $|E|^2$ 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 $E_t$ is individually encoded by BERT, represented by the hidden state of the CLS token in the last Transformers layer, denoted by $g_t$. An additional layer is stacked on top of the CLS hidden states to produce the score $s_t^i$ for $E_t$. The distillation is achieved by training the model to mimic PLL scores using mean squared error (MSE) regression loss:

$$g_t = \text{BERT}^{\text{CLS}}(E_t)$$

(2)

$$s_t^i = \text{FFNN}(g_t)$$

(3)

$$L_{\text{MD}} = |s_t^i - PLL(E_t)|^2$$

(4)

FFNN denotes the learnable feed-forward neural network, $s_t^i$ is the predicted PLL approximation, and PLL($E_t$) 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 $\hat{E} = \{E_1, ..., E_n\}$ be the n-best hypotheses obtained from beam search in the first-pass decoder. For any $E_i \in \hat{E}$, let $s_i^k$ be its given score from the first pass, and $s_i^l$ be the score from the second pass (same as Eq. (5)), with lower scores for more likely hypotheses for both; let $e_i$ 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:

$$s_i = s_i^k + \beta \cdot s_i^l$$

(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:

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

(6)

$$\hat{\tau}_H = \frac{1}{n} \sum_{i=1}^n \epsilon_i$$

(7)

$$L_{\text{MWER}} = \sum_{i=1}^n P_i \cdot (\epsilon_i - \hat{\tau}_H).$$

(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 $L_{\text{MWER}}$ represents the expected number of relative word errors, with $\hat{\tau}_H$ 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:

$$d_i^e = \frac{e^{\epsilon_i}}{\sum_{j=1}^n e^{\epsilon_j}}$$

(9)

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

(10)

$$L_{\text{MWED}} = - \sum_{i=1}^n d_i^e \log d_i^s$$

(11)

d_i^e$ and $d_i^s$ represent the relative distribution of word errors and predicted scores over the n-best list. $L_{\text{MWED}}$ 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 $s_i^k$ 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 $T = \frac{1}{\sum_{i=1}^n s_i/\sum_{i=1}^n \epsilon_i}$ 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, 5], 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 $L_{\text{MWER}}$ or $L_{\text{MWED}}$ defined earlier.

In Section 2.2.2, we show results using this approach with $L_{\text{MWED}}$, labeled “MWER Only”, where we perform MWER training on a pretrained 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 $L$ to replace $L_{\text{MWER}}$ or $L_{\text{MWED}}$ in the discriminative training step:

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

where $L_{\text{Discriminative}}$ is the discriminative loss that can be either $L_{\text{MWER}}$ or $L_{\text{MWED}}$. $L_{\text{MD}}$ 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 $L_{\text{MWER}}$ and $L_{\text{MWED}}$, 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 [19].

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), ~100/5 hours of utterances are used as the MWER training/dev set; the test set has ~2 hours of long-tail info utterances. For Info and Nav in ja, we use a single MWER training/dev set that consists of ~220/10 hours of utterances in multiple domains including Info and Nav domains. The test set has ~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_base for our experiments to enable direct comparison with previous work. For internal datasets, we use an in-house multilingual BERT of ~170M 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 ~17M/5M 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 2.2.2 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
discriminative training with MWER and MWED loss functions (as in Section 3.2.1), respectively.

Based on these results, we can make three observations. First, discriminative training significantly improves upon nondiscriminative 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 REL 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 obtaining 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 1 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 ∼3% the parameters of the 170M model, with still nearly 8% improvement on two ja datasets and 3+% on Info (en).

Table 2 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 2(a) and 2(b), 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


