



Internal Language Model Adaptation with Text-Only Data for End-to-End Speech Recognition

Zhong Meng¹, Yashesh Gaur¹, Naoyuki Kanda¹, Jinyu Li¹, Xie Chen¹, Yu Wu², Yifan Gong¹

¹Microsoft Corporation, USA ²Microsoft Research Asia, China
{zhme, yagaur, nakanda, jinyuli, xieche, yuwu1, ygong}@microsoft.com

Abstract

Text-only adaptation of an end-to-end (E2E) model remains a challenging task for automatic speech recognition (ASR). Language model (LM) fusion-based approaches require an additional external LM during inference, significantly increasing the computation cost. To overcome this, we propose an internal LM adaptation (ILMA) of the E2E model using *text-only* data. Trained with audio-transcript pairs, an E2E model implicitly learns an internal LM that characterizes the token sequence probability which is approximated by the E2E model output after zeroing out the encoder contribution. During ILMA, we fine-tune the internal LM, i.e., the E2E components excluding the encoder, to minimize a cross-entropy loss. To make ILMA effective, it is essential to train the E2E model with an internal LM loss besides the standard E2E loss. Furthermore, we propose to regularize ILMA by minimizing the Kullback-Leibler divergence between the output distributions of the adapted and unadapted internal LMs. ILMA is the most effective when we update only the last linear layer of the joint network. ILMA enables a fast text-only adaptation of the E2E model without increasing the run-time computational cost. Experimented with 30K-hour trained transformer transducer models, ILMA achieves up to 34.9% relative word error rate reduction from the unadapted baseline.

Index Terms: Speech recognition, internal language model, text-only adaptation

1. Introduction

End-to-end (E2E) models have achieved the state-of-art performance for automatic speech recognition (ASR) [1, 2, 3, 4]. With the goal of directly mapping speech features into word sequences using a single neural network (NN), the most popular E2E models include connectionist temporal classification [5, 6, 7], recurrent neural network transducer (RNN-T) [8, 2, 3], and attention-based encoder-decoder (AED) models [9, 10, 11]. However, the performance of E2E models degrades when a mismatch exists between the source-domain training data and the target-domain test data. Many ideas have been proposed to adapt ASR models, such as regularization methods [12, 13, 14, 15, 16], teacher-student learning [17, 18, 19, 20], transformation methods [21, 22], and adversarial learning [23, 24, 25, 26]. However, all these methods require audio data for adaptation when applied to E2E models [27, 28, 29]. To overcome this, the most promising approach is to adapt the E2E model using *text-only* data because it is easy to collect orders of magnitude more text-only data than the audio-transcript pairs in the target domain.

The most common solution is to train an LM using text-only adaptation data and integrate it into the E2E model during inference. The simplest LM fusion method is Shallow Fusion [6, 30] which combines the E2E model score and the external LM score in the log-linear domain at each step of the beam search. To improve Shallow Fusion, a Density Ratio method [31, 32] and an internal LM estimation-based Fusion [33, 34] were proposed in which a source-domain LM score and an internal LM score

are subtracted from the Shallow Fusion score, respectively, during inference. Variants of internal LM estimation were proposed in [35, 36]. Minimum word error rate training with LM fusion [37, 38, 39, 40] was conducted to obviate the need for LM weights tuning. Furthermore, structural LM fusion methods such as Deep Fusion [30], Cold Fusion [41] and Simple Fusion [42] jointly train an E2E model with an external LM to learn a sophisticated combination between the two models. However, all these fusion methods involve an additional external LM during inference which significantly increases the run-time computational cost.

Other solutions include synthesizing speech from the text-only adaptation data using a text-to-speech (TTS) model, and then fine-tuning the E2E model with the synthesized audio-transcript pairs [43, 40, 44]. During inference, only the adapted E2E model is needed. However, these approaches require additional transcribed speech to train a TTS model. Training such a reliable multi-speaker TTS model and generating synthesized speech from it are both computationally expensive. Moreover, the adaptation cost with audio-transcript pairs is typically much higher than that with text-only data. Therefore, it is hard to apply these methods to the fast on-device adaptation where the target-domain text is frequently updated. Recently, [45] shows that an RNN-T can be adapted with text-only data by training an auxiliary LM output layer and using it as a regularizer to fine-tune the prediction network. However, learning the additional LM output layer complicates the adaptation process, leading to increased adaptation time and computational cost. [46] factorizes the blank and vocabulary predictions in a transducer and effectively adapts the vocabulary predictor with text as adapting an LM. However, the additional blank predictor increases the model size and computational cost.

To address these limitations, we propose an internal LM adaptation (ILMA) of an E2E model using the *text-only* data to improve the ASR performance on the target-domain test data. Trained with audio-transcript pairs, an E2E model implicitly learns an internal LM that characterizes token sequence distribution. The internal LM probability is estimated by the E2E model output after eliminating the encoder contribution from the network [33, 34]. Therefore, we define the E2E model components excluding the encoder as an *internal LM*. Specifically, the internal LM refers to the prediction network followed by the joint network for a transducer model, or to the decoder for an AED model. During ILMA, we fine-tune the internal LM parameters to maximize the log probability of the adaptation sentences. Before ILMA, it is essential to perform internal LM training (ILMT) [47] to minimize a cross-entropy internal LM loss in addition to the standard E2E loss. ILMT learns a *dual-mode* internal LM that acts as a standalone LM without losing its capability of generating the correct E2E scores together with the other components of the E2E model. Kullback-Leibler divergence (KLD) [48] regularization is performed during ILMA to prevent the adapted E2E model from overfitting the text-only data. We show that, for a transducer model, ILMA is the most effective when we update only the last linear layer of the joint network. Compared to LM fusion and TTS-based adaptation, ILMA enables a fast text-only adaptation without increasing run-

time computational cost during inference. Evaluated with 30 thousand (K)-hour trained transformer transducers (T-Ts) [49], ILMA achieves up to 34.9% relative word error rate (WER) reduction from the unadapted T-T on Microsoft production test sets.

2. Related Work

2.1. E2E Models for ASR

Given a sequence of speech features $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ where $\mathbf{x}_t \in \mathbb{R}^d$, an E2E model is trained to minimize the summation of the negative log posteriors of the reference token sequences $\mathbf{Y} = \{y_1, \dots, y_U\}$ over the training corpus \mathcal{D}_T where $y_u \in \mathcal{V}$ and \mathcal{V} is the set of non-blank output tokens, e.g., word pieces,

$$\mathcal{L}_{\text{E2E}}(\theta_{\text{E2E}}) = - \sum_{(\mathbf{x}, \mathbf{Y}) \in \mathcal{D}_T} \log P(\mathbf{Y}|\mathbf{X}; \theta_{\text{E2E}}), \quad (1)$$

where θ_{E2E} is the parameters of an E2E model.

A transducer model [8] comprises an encoder, a prediction network and a joint network. The encoder maps the input speech features \mathbf{X} to a sequence of hidden states $\mathbf{H}^{\text{enc}} = \{\mathbf{h}_1^{\text{enc}}, \dots, \mathbf{h}_T^{\text{enc}}\}$. The predictor takes in the embedding vectors of the previous *non-blank* token $\mathbf{Y}_{0:u-1}$ to generate the hidden state $\mathbf{h}_u^{\text{pred}}$. The joint network is a feed-forward network that combines the outputs of the encoder and predictor to predict the conditional distribution over the next possible token $k \in \mathcal{V} \cup \langle \text{b} \rangle$, i.e.,

$$\mathbf{z}_{t,u} = W_j \phi(W_e \mathbf{h}_t^{\text{enc}} + W_p \mathbf{h}_u^{\text{pred}} + \mathbf{b}_e + \mathbf{b}_p) + \mathbf{b}_j, \quad (2)$$

$$[P(k|t, u; \theta_{\text{E2E}})]_{k \in \mathcal{V} \cup \langle \text{b} \rangle} = \text{softmax}(\mathbf{z}_{t,u}), \quad (3)$$

where $\langle \text{b} \rangle$ denotes a blank symbol, ϕ is a non-linear function, e.g., tanh or ReLU. W_j , W_e , W_p are weight matrices, and \mathbf{b}_e , \mathbf{b}_p , \mathbf{b}_j are biases. $\mathbf{z}_{t,u}$ is a $|\mathcal{V}| + 1$ dimensional logit vector.

2.2. Internal LM of E2E Model

The E2E model implicitly learns an internal LM via the E2E training with paired audio and transcripts. The internal LM characterizes the distribution of token sequences given θ_{E2E} . The internal LM probability of a token sequence \mathbf{Y} is

$$P(\mathbf{Y}; \theta_{\text{E2E}}) = \prod_{u=1}^U P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{E2E}}), \quad (4)$$

where $P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{E2E}})$ is estimated as the E2E model output at the step u after zeroing out the encoder hidden states \mathbf{H}^{enc} [33, 34]. For a transducer, the internal LM probability is the model output after feeding the previous tokens $\mathbf{Y}_{0:u-1}$ through the internal LM, i.e., the prediction network followed by the joint network

$$\mathbf{z}_u^{\text{ILM, NB}} = \mathbf{W}_j^{\text{NB}} \phi(\mathbf{W}_p \mathbf{h}_u^{\text{pred}} + \mathbf{b}_p) + \mathbf{b}_j^{\text{NB}}, \quad (5)$$

$$P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{pred}}, \theta_{\text{joint}}) = \text{softmax}(\mathbf{z}_u^{\text{ILM, NB}}), \quad (6)$$

where NB denotes non-blank, \mathbf{W}_j^{NB} and \mathbf{b}_j^{NB} are created from \mathbf{W}_j and \mathbf{b}_j , respectively, by removing the rows corresponding to the blank, and $\mathbf{z}_u^{\text{ILM, NB}}$ is a logit vector of dimension $|\mathcal{V}|$ without the blank logit. θ_{pred} and θ_{joint} are the parameters of the prediction network and joint network, respectively.

2.3. Internal LM Training (ILMT)

The standard E2E training learns a weak internal LM with a high perplexity because it maximizes the posterior $P(\mathbf{Y}|\mathbf{X}; \theta_{\text{E2E}})$ instead of the internal LM probability $P(\mathbf{Y}; \theta_{\text{E2E}})$. To address this limitation, ILMT was proposed in [47] where the E2E model is trained to minimize an internal LM loss in addition to the standard E2E loss.

The internal LM loss is the summation of negative log internal LM probabilities over the training corpus \mathcal{D}_T as follows

$$\mathcal{L}_{\text{ILM}}(\theta_{\text{ILM}}; \mathcal{D}_T) = - \sum_{\mathbf{Y} \in \mathcal{D}_T} \sum_{u=1}^U \log P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}), \quad (7)$$

where θ_{ILM} denotes the internal LM parameters. θ_{ILM} equals to $\{\theta_{\text{pred}}, \theta_{\text{joint}}\}$ for a transducer model, and denotes the decoder parameters for an AED model. The ILMT loss is a weighted sum of the E2E loss in Eq. (1) and the internal LM loss in Eq. (7)

$$\mathcal{L}_{\text{ILMT}}(\theta_{\text{E2E}}; \mathcal{D}_T) = \mathcal{L}_{\text{E2E}}(\theta_{\text{E2E}}; \mathcal{D}_T) + \alpha \mathcal{L}_{\text{ILM}}(\theta_{\text{ILM}}; \mathcal{D}_T), \quad (8)$$

where α is the weight of the internal LM loss. By minimizing the ILMT loss, we maximize the internal LM probability of the E2E training transcripts by updating only the internal LM while maximizing the conditional probability of the training transcripts given input speech by updating the entire E2E model.

ILMT learns a strong internal LM with significantly reduced perplexity without losing the ASR performance. It encourages the internal LM to behave also like a standalone NN-LM [50], increasing the modularity of the E2E model.

3. Internal LM Adaptation (ILMA)

Our goal is to adapt the E2E model using *text-only* data so that it achieves improved ASR performance in the target domain without any increase of the run-time computational cost. Since an internal LM of an E2E model acts exactly the same as a standard NN-LM, one possible way is to adapt the internal LM using the text-only data. Theoretically, any NN-LM adaptation methods are applicable to ILMA. The simplest solution is to re-train the internal LM to minimize the cross-entropy internal LM loss as follows

$$\mathcal{L}_{\text{ILM}}(\theta_{\text{ILM}}; \mathcal{D}_A) = - \sum_{\mathbf{Y} \in \mathcal{D}_A} \sum_{u=1}^U \log P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}). \quad (9)$$

Different from ILMT where the internal LM loss in Eq. (7) is computed with the transcripts of the training data \mathcal{D}_T , the internal LM loss of ILMA is computed with the sentences in the text-only adaptation set \mathcal{D}_A .

In the standard E2E model training, the internal LM is updated together with the encoder to minimize the internal LM loss. However, in Eq. (9), the internal LM is re-trained separately to become a standalone NN-LM by maximizing the log probability of the adaptation sentences. After ILMA of a standard E2E model, it is probable that the adapted internal LM deviates too much from the unadapted one such that it forgets how to work together with the encoder to generate reasonable E2E model probabilities during inference. To circumvent this, we perform ILMT of the E2E model to minimize an ILMT loss in Eq. (8) before adapting the internal LM towards text-only data. Through ILMT, a *dual-mode* internal LM is learned to behave like a standalone LM while maintaining its capability to collaborate with the encoder to compute the correct E2E model probabilities. During ILMA, only the ‘‘LM mode’’ of the internal LM is adapted towards the text-only data while its ‘‘E2E component mode’’ remains unchanged, leading to improved performance on the target-domain test data.

Further, to prevent the adapted E2E model from overfitting the text-only adaptation data, we minimize the KLD between the output distribution of the adapted internal LM and that of unadapted one below in addition to the cross-entropy internal LM loss

$$\mathcal{KL} [P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}^*) || P(y_u | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}})] = \sum_{v \in \mathcal{V}} P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}^*) \log \left[\frac{P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}^*)}{P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}})} \right], \quad (10)$$

where θ_{ILM}^* is the internal LM parameters before ILMA. With KLD regularization, the ILMA loss becomes

$$\begin{aligned} \mathcal{L}_{\text{ILMA}}(\theta_{\text{ILM}}; \mathcal{D}_A) &= (1 - \rho)\mathcal{L}_{\text{ILM}}(\theta_{\text{ILM}}; \mathcal{D}_A) \\ &- \rho \sum_{\mathbf{Y} \in \mathcal{D}_A} \sum_{u=1}^U \sum_{v \in \mathcal{V}} P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}^*) \log P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}) \\ &= - \sum_{\mathbf{Y} \in \mathcal{D}_A} \sum_{u=1}^U \sum_{v \in \mathcal{V}} [(1 - \rho)\delta(v = y_u) + \rho P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}^*)] \\ &\quad \log P(v | \mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}), \end{aligned} \quad (11)$$

where $\rho \in [0, 1]$ is the regularization weight and $\delta(v = y_u)$ is the Kronecker function which equals 1 when $v = y_u$ and 0 otherwise. Note that ILMA with KLD regularization in Eq. (11) reduces to the cross-entropy ILMA in Eq. (9) when $\rho = 0$.

During ILMA, different parts of the internal LM parameters can be updated to minimize the ILMA loss. We find that, for a transducer model, ILMA is the most effective when updating only $\{\mathbf{W}_j^{\text{NB}}, \mathbf{b}_j^{\text{NB}}\}$ of the joint network. One possible reason is: The adaptation text does not include the blank token $\langle b \rangle$, so the best we can do is to *only* adapt the transducer’s capability of predicting non-blank tokens *without* sacrificing its capability of predicting $\langle b \rangle$. In a transducer, the only parameters that do not contribute to $P(\langle b \rangle | \mathbf{Y}_{0:u-1}; \theta_{\text{E2E}}^S)$ are $\{\mathbf{W}_j^{\text{NB}}, \mathbf{b}_j^{\text{NB}}\}$. Therefore, adapting $\{\mathbf{W}_j^{\text{NB}}, \mathbf{b}_j^{\text{NB}}\}$ performs the best while adapting any other parameters may hurt the $\langle b \rangle$ predictions.

In summary, ILMA of an E2E model with text-only data includes the following steps:

1. Perform ILMT of the E2E model with audio-transcript pairs in the training set \mathcal{D}_T by minimizing the ILMT loss in Eq. (8) to generate the model M_1 .
2. Perform ILMA of the E2E model M_1 with *text-only* data \mathcal{D}_A by minimizing the ILMA loss defined in Eq. (11) to generate the model M_2 .
3. Perform inference with the adapted E2E model M_2 on the target-domain test data.

ILMA enables a fast text-only adaptation of the internal LM without losing its functionality to work seamlessly with the other components of an E2E model. Contrary to LM fusion methods, ILMA does *not* increase any computational cost during inference because only the adapted E2E model is needed to run the beam search decoding. Compared to Fast Text-only Adaptation (FTA) [45], ILMA saves one step of training the additional output LM component, effectively reducing the training time and the computational cost. Most importantly, ILMA has the flexibility to adapt only the joint network (the most effective) while FTA does not.

4. Experiments

We perform text-only ILMA of the T-T models trained with a transducer loss or a ILMT loss and evaluate the adapted models on intra-domain and cross-domain test sets. During inference, we perform beam search with a beam size of 5 *without* using any external LM. 3999 word-piece units generated by byte-pair encoding [51] are used as the output token set \mathcal{V} for all T-T models.

4.1. Data Preparation

We train T-T models with 30K hours of anonymized and transcribed data as in [34, 47] collected from Microsoft services, including voice search, short message dictation, conversations, etc. In addition, we collect two *text-only* adaptation sets as follows

- **Command & Control:** 50 million (M) words of simulated text generated by a command & control grammar and 400K

words of real anonymized text from Microsoft command & control services.

- **Multi-Domain:** 2 billion (B) words of anonymized text corpus comprising conversational data, voice search and short message dictation from Microsoft services.

Correspondingly, we gather two target-domain test sets whose transcripts do not overlap with the adaptation text.

- **Command & Control:** 1K utterances collected from Microsoft command & control services.
- **Voice Search:** 18K far-field voice search utterances collected from Microsoft services.

Note that Command & Control is a *cross-domain* evaluation set because it is not covered by the 30K-hour training data while Voice Search is an *intra-domain* one because 30K-hour training data includes voice search utterances. We extract 80-dimensional (dim) log Mel filter bank features from the speech signal for training and test sets every 10 ms over a 25 ms window.

4.2. Baseline Systems

T-T has a streaming encoder [52] with four 2D convolution layers sub-sampling the 80-dim log Mel filter bank features in time by a factor of 4. On top of that is a transformer with 18 layers, each layer has an 8-head attention sub-layer with relative positional encoding [53] and a 2048-dim fully-connected sub-layer. The encoder has a look-ahead of 360 ms on average. The T-T prediction network is a 2-layer transformer with each layer containing a 4-head attention sub-layer followed by a 1024-dim fully-connected sub-layer. The input to the prediction network are 512-dim word-piece embedding vectors with positional encoding. The attention dimension is fixed at 512. The outputs of encoder and prediction network are projected to 512-dim vectors. Dropout with a probability of 0.1 is deployed. T-T has 67 M parameters.

We first train a T-T model to minimize the transducer loss using the 30K-hour data as the baseline model. In Table 1, the baseline T-T achieves 12.81% and 4.23% WERs on Command & Control and Voice Search test sets, respectively. To achieve an effective ILMA, we also perform the ILMT of T-T to minimize the ILMT loss in Eq. (8) with the 30K-hour data before ILMA. The ILMT-ed T-T achieves 12.55% and 4.22% WERs on Command & Control and Voice Search test sets, performing slightly better than the baseline T-T. With ILMT, the token-level perplexity of the internal LM reduces from 131.2 (baseline) to 60.7 on the validation set of the 30K-hour training data. This shows that a dual-mode internal LM with significantly lower perplexity and slightly better ASR performance is learned via ILMT.

To compare with ILMA, we perform FTA [45] on the baseline T-T. The output LM component is a feed-forward layer followed by a softmax. In Table 1, FTA achieves 10.04% and 4.15% WERs on Command & Control and Voice Search test sets, respectively. Moreover, we train two long short-term memory (LSTM) [54, 55]-LMs with Command & Control and Multi-Domain adaptation text, respectively. Both LSTM-LMs have two hidden layers and 2048 hidden units at each layer. Each LSTM-LM consists of 58M parameters. We perform Shallow Fusion of the ILMT-ed T-T and the LSTM-LM during inference. In Table 1, Shallow Fusion achieves 8.71% and 4.04% WERs for cross-domain and intra-domain evaluations, respectively.

4.3. Internal LM Adaptation

4.3.1. Cross-Domain Evaluation

We adapt the baseline T-T with text-only data in the Command & Control adaptation set by minimizing the ILMA loss in Eq. (11). Different components of the internal LM are updated during ILMA. We also adjust the KLD regularization weight ρ to

Method	External LM	Model Params	Comd. Control	Voice Search
T-T			12.81	4.23
ILMT	No	67M	12.55	4.22
FTA			10.04	4.15
ILMT + SF	Yes	125M	8.71	4.04
ILMT + ILMA	No	67M	8.34	3.95

Table 1: Compare WERs (%) and model sizes of 3 adaptation methods: Fast Text-only Adaptation (FTA) [45], Shallow Fusion (SF) of ILMT-ed T-T and NN-LM, proposed ILMA after ILMT.

Method	Adapted Params	KLD Regularization Weight ρ			
		0.0	0.2	0.5	0.8
T-T	-	12.81			
+ ILMA	ILM	12.30	11.93	12.44	12.29
	Predictor	12.32	12.01	12.34	12.83
	Joiner	10.85	10.36	10.53	11.09
ILMT	-	12.55			
+ ILMA	ILM	10.34	9.67	10.17	11.59
	Predictor	10.47	10.02	10.42	11.50
	Joiner	8.96	8.34	8.51	9.65

Table 2: The WERs (%) of T-T models on the **cross-domain Command & Control** test set. T-Ts are first trained with a standard transducer loss (T-T) or an ILMT loss, and then adapted with the ILMA loss using Command & Control text-only adaptation data. Joiner represents $\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}$, i.e., the joint network output layer after removing the rows corresponding to the blank.

prevent the T-T from overfitting the adaptation text. As in Table 2, when updating the entire internal LM, the prediction network alone or $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ of the joint network (Eq. (5)), we obtain the largest relative WER reductions 6.8%, 6.2% and 19.1%, respectively, from the baseline T-T at a common weight $\rho = 0.2$. Adapting $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ alone performs significantly better than adapting the entire internal LM or the predictor alone.

We then perform ILMA of the ILMT-ed T-T using the text-only data from the Command & Control adaptation set. ILMA achieves 24.5%, 21.8% and 34.9% relative WER reductions from the baseline T-T by updating the entire internal LM, the predictor alone or $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ of the joint network, respectively, when $\rho = 0.2$. Among the three update schemes, adapting $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ alone performs the best, achieving 13.8% and 16.8% relative WER reductions from adapting the whole internal LM and adapting only the prediction network, respectively. It also outperforms ILMT-ed T-T and FTA by 33.5% and 16.9% relatively in terms of lower WER, respectively. Moreover, in Table 1, ILMA achieves a 4.2% relative WER reduction from Shallow Fusion with 46.4% fewer model parameters and 20.7% shorter inference time.

4.3.2. Intra-Domain Evaluation

We perform ILMA of the baseline T-T using the Multi-Domain adaptation text. However, as in Table 3, adapting the entire internal LM or the predictor alone degrades the WER of the baseline T-T for all regularization weights. Even adapting $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ alone achieves little relative WER reduction from the baseline T-T.

We then adapt the internal LM of the ILMT-ed T-T using the Multi-Domain text. ILMA achieves 2.6%, 3.3% and 6.6% relative WER reductions from the baseline T-T when updating the entire internal LM, the predictor alone and $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ of the joint network, respectively, when $\rho = 0.5$. Adapting $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ performs the best among all three update schemes, with 6.4% and 4.8% relative WER reductions from ILMT-ed T-T and FTA, respectively. Furthermore, in Table 1, ILMA achieves a 2.2% rela-

Method	Adapted Params	KLD Regularization Weight ρ			
		0.0	0.2	0.5	0.8
T-T	-	4.23			
+ ILMA	ILM	4.31	4.31	4.32	4.29
	Predictor	4.28	4.30	4.30	4.31
	Joiner	4.19	4.19	4.19	4.18
ILMT	-	4.22			
+ ILMA	ILM	4.13	4.12	4.12	4.13
	Predictor	4.11	4.10	4.09	4.10
	Joiner	4.07	4.01	3.95	4.00

Table 3: The WERs (%) of T-T models on the **intra-domain Voice Search** test set. T-Ts are first trained with a standard transducer loss (T-T) or an ILMT loss, and then adapted with the ILMA loss using Multi-Domain text-only adaptation data. Joiner represents $\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}$, i.e., the joint network output layer after removing the rows corresponding to the blank.

tive WER reduction from Shallow Fusion with 46.4% fewer model parameters and 25.6% shorter inference time.

4.4. Result Analysis

T-T + ILMA results in Table 3 indicate that the ILMA of a T-T trained with standard transducer loss may degrade the ASR performance because minimizing the ILMA loss pushes the internal LM towards a pure NN-LM that is incompatible with the other components of the E2E model. By comparing the T-T + ILMA and ILMT + ILMA results in both tables, we see that the ILMA of a T-T trained with ILMT loss achieves much larger relative WER reductions than that of a T-T trained with a standard transducer loss for both cross-domain and intra-domain evaluations. This shows that the dual-mode internal LM learned via ILMT can be adapted effectively as a standalone LM using *text-only* data while its full functionality as an essential T-T component is well maintained.

ILMT + ILMA results in Tables 2 and 3 suggest that adding a proper KLD regularization ($\rho > 0$) always reduces WER by preventing the E2E model from overfitting the adaptation text. Furthermore, ILMA by updating only $\{\mathbf{W}_j^{NB}, \mathbf{b}_j^{NB}\}$ of the joint network shows consistently and significantly better performance than updating the entire internal LM or the prediction network alone.

Compared to FTA [45], the proposed ILMA saves one step of output LM component training, achieving *simpler* and *faster* text-only adaptation. Most importantly, as in Table 1, ILMA achieves 4.8% - 16.9% relatively lower WERs than FTA. Further, ILMA consistently outperforms Shallow Fusion with significantly reduced model parameters, computational cost and inference time.

5. Conclusion

In this work, we propose an internal LM adaptation of the E2E model using text-only data. During ILMA, the internal LM of the E2E model is fine-tuned with adaptation text to minimize a cross-entropy internal LM loss. The internal LM training of the E2E model is performed before ILMA to ensure that the internal LM behaves like a standalone NN-LM while maintaining its functionality as an inseparable component of the E2E model. The KLD regularization is added to the ILMA loss to avoid overfitting.

Experimented with a 30K-hour trained T-T model, ILMA achieves up to 34.9% and 6.6% relative WER reductions from the baseline T-T on cross-domain and intra-domain test sets, respectively. We also show that ILMT is necessary for an effective and robust ILMA. During ILMA, updating only the last linear layer of the joint network consistently achieves the best performance.

6. References

- [1] C.-C. Chiu, T. N. Sainath *et al.*, “State-of-the-art speech recognition with sequence-to-sequence models,” in *Proc. ICASSP*, 2018.
- [2] T. Sainath, Y. He, B. Li *et al.*, “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in *Proc. ICASSP*, 2020, pp. 6059–6063.
- [3] J. Li, R. Zhao, Z. Meng *et al.*, “Developing RNN-T models surpassing high-performance hybrid models with customization capability,” in *Interspeech*, 2020.
- [4] J. Li, “Recent advances in end-to-end automatic speech recognition,” *arXiv preprint arXiv:2111.01690*, 2021.
- [5] A. Graves, S. Fernández, F. Gomez *et al.*, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML*. ACM, 2006.
- [6] A. Hannun, C. Case *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [7] J. Li, G. Ye, A. Das *et al.*, “Advancing acoustic-to-word CTC model,” in *Proc. ICASSP*, 2018.
- [8] A. Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [9] J. K. Chorowski, D. Bahdanau, D. Serdyuk *et al.*, “Attention-based models for speech recognition,” in *NIPS*, 2015.
- [10] S. Karita, N. Chen *et al.*, “A comparative study on transformer vs RNN in speech applications,” in *Proc. ASRU*, 2019.
- [11] J. Li, Y. Wu *et al.*, “On the comparison of popular end-to-end models for large scale speech recognition,” in *Proc. Interspeech*, 2020.
- [12] D. Yu, K. Yao, H. Su *et al.*, “Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition,” in *Proc. ICASSP*, May 2013.
- [13] Z. Meng, J. Li, and Y. Gong, “Adversarial speaker adaptation,” in *Proc. ICASSP*, 2019.
- [14] H. Liao, “Speaker adaptation of context dependent deep neural networks,” in *Proc. ICASSP*, May 2013.
- [15] Z. Meng, H. Hu, J. Li *et al.*, “L-vector: Neural label embedding for domain adaptation,” in *Proc. ICASSP*. IEEE, 2020.
- [16] P. Swietojanski and S. Renals, “Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models,” in *Proc. SLT*. IEEE, 2014, pp. 171–176.
- [17] J. Li, R. Zhao *et al.*, “Learning small-size DNN with output-distribution-based criteria,” in *INTERSPEECH*, 2014.
- [18] Z. Meng, J. Li, Y. Gong *et al.*, “Adversarial teacher-student learning for unsupervised domain adaptation,” in *Proc. ICASSP*, 2018.
- [19] V. Manohar, P. Ghahremani, D. Povey *et al.*, “A teacher-student learning approach for unsupervised domain adaptation of sequence-trained ASR models,” in *Proc. SLT*. IEEE, 2018.
- [20] Z. Meng, J. Li, Y. Zhao, and Y. Gong, “Conditional teacher-student learning,” in *Proc. ICASSP*, 2019.
- [21] R. Gemello, F. Mana, S. Scanzio *et al.*, “Linear hidden transformations for adaptation of hybrid ANN/HMM models,” *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.
- [22] T. Tan, Y. Qian, M. Yin *et al.*, “Cluster adaptive training for deep neural network,” in *Proc. ICASSP*. IEEE, 2015.
- [23] Z. Meng, J. Li, Z. Chen *et al.*, “Speaker-invariant training via adversarial learning,” in *Proc. ICASSP*, 2018.
- [24] Y. Shinohara, “Adversarial multi-task learning of deep neural networks for robust speech recognition,” in *INTERSPEECH*, 2016.
- [25] D. Serdyuk, K. Audhkhasi *et al.*, “Invariant representations for noisy speech recognition,” in *NIPS Workshop*, 2016.
- [26] Z. Meng, Z. Chen, V. Mazalov, J. Li, and Y. Gong, “Unsupervised adaptation with domain separation networks for robust speech recognition,” in *Proc. ASRU*, 2017.
- [27] T. Ochiai, S. Watanabe *et al.*, “Speaker adaptation for multichannel end-to-end speech recognition,” in *Proc. ICASSP*, 2018.
- [28] Z. Meng, Y. Gaur, J. Li *et al.*, “Speaker adaptation for attention-based end-to-end speech recognition,” *Proc. Interspeech*, 2019.
- [29] Z. Meng, J. Li, Y. Gaur, and Y. Gong, “Domain adaptation via teacher-student learning for end-to-end speech recognition,” in *Proc. ASRU*. IEEE, 2019.
- [30] C. Gulcehre, O. Firat, K. Xu *et al.*, “On using monolingual corpora in neural machine translation,” *CoRR*, 2015.
- [31] E. McDermott, H. Sak, and E. Variani, “A density ratio approach to language model fusion in end-to-end automatic speech recognition,” in *Proc. ASRU*. IEEE, 2019, pp. 434–441.
- [32] N. Kanda, X. Lu, and H. Kawai, “Maximum a posteriori based decoding for CTC acoustic models,” in *Interspeech*, 2016.
- [33] E. Variani, D. Rybach, C. Allauzen *et al.*, “Hybrid autoregressive transducer (hat),” *arXiv preprint arXiv:2003.07705*, 2020.
- [34] Z. Meng, S. Parthasarathy, E. Sun *et al.*, “Internal language model estimation for domain-adaptive end-to-end speech recognition,” in *Proc. SLT*. IEEE, 2021.
- [35] A. Zeyer, A. Merboldt, W. Michel *et al.*, “Librispeech transducer model with internal language model prior correction,” *arXiv preprint arXiv:2104.03006*, 2021.
- [36] M. Zeineldeen, A. Glushko, W. Michel *et al.*, “Investigating methods to improve language model integration for attention-based encoder-decoder ASR models,” *arXiv preprint arXiv:2104.05544*, 2021.
- [37] Z. Meng, Y. Wu, N. Kanda *et al.*, “Minimum word error rate training with language model fusion for end-to-end speech recognition,” *Proc. Interspeech*, 2021.
- [38] N. Kanda, X. Lu, and H. Kawai, “Minimum bayes risk training of CTC acoustic models in maximum a posteriori based decoding framework,” in *Proc. ICASSP*. IEEE, 2017.
- [39] C. Weng, C. Yu, J. Cui *et al.*, “Minimum bayes risk training of RNN-transducer for end-to-end speech recognition,” *arXiv preprint arXiv:1911.12487*, 2019.
- [40] C. Peyser, S. Mavandadi, T. Sainath *et al.*, “Improving tail performance of a deliberation E2E ASR model using a large text corpus,” in *INTERSPEECH*, 2020.
- [41] A. Sriram, H. Jun, S. Satheesh *et al.*, “Cold fusion: Training seq2seq models together with language models,” *Proc. Interspeech*, 2018.
- [42] F. Stahlberg, J. Cross, and V. Stoyanov, “Simple fusion: Return of the language model,” *arXiv preprint arXiv:1809.00125*, 2018.
- [43] Y. Huang, J. Li, L. He *et al.*, “Rapid RNN-T adaptation using personalized speech synthesis and neural language generator,” in *INTERSPEECH*, 2020, pp. 1256–1260.
- [44] M. K. Baskar, S. Watanabe, R. Astudillo *et al.*, “Self-supervised sequence-to-sequence ASR using unpaired speech and text,” *arXiv preprint arXiv:1905.01152*, 2019.
- [45] J. Pyllkkönen, A. Ukkonen, J. Kilpikoski *et al.*, “Fast text-only domain adaptation of RNN-transducer prediction network,” *arXiv preprint arXiv:2104.11127*, 2021.
- [46] X. Chen, Z. Meng *et al.*, “Factorized neural transducer for efficient language model adaptation,” in *Proc. ICASSP*. IEEE, 2022.
- [47] Z. Meng, N. Kanda, Y. Gaur *et al.*, “Internal language model training for domain-adaptive end-to-end speech recognition,” in *Proc. ICASSP*. IEEE, 2021.
- [48] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [49] Q. Zhang, H. Lu, H. Sak *et al.*, “Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss,” in *Proc. ICASSP*. IEEE, 2020.
- [50] T. Mikolov, M. Karafiát *et al.*, “Recurrent neural network based language model,” in *Interspeech*, 2010.
- [51] R. Sennrich *et al.*, “Neural machine translation of rare words with subword units,” *arXiv preprint arXiv:1508.07909*, 2015.
- [52] X. Chen, Y. Wu, Z. Wang *et al.*, “Developing real-time streaming transformer transducer for speech recognition on large-scale dataset,” *arXiv preprint arXiv:2010.11395*, 2020.
- [53] A. Vaswani, N. Shazeer, N. Parmar *et al.*, “Attention is all you need,” in *Proc. NIPS*, vol. 30, 2017.
- [54] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Interspeech*, 2014.
- [55] Z. Meng, S. Watanabe, J. R. Hershey *et al.*, “Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition,” in *ICASSP*. IEEE, 2017.