2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) | 979-8-3503-0689-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASRU57964.2023.10389682

EFFICIENT TEXT-ONLY DOMAIN ADAPTATION FOR CTC-BASED ASR

Chang Chen, Xun Gong, Yanmin Qian†

MoE Key Lab of Artificial Intelligence, AI Institute
Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{chenchang,gongxun,yanminqian}@sjtu.edu.cn

ABSTRACT

For connectionist temporal classification (CTC) based speech recognition (ASR) models, text-only domain adaptation still faces several challenges. In this study, we propose an efficient text-only domain adaptation method for CTC-based models. We introduce the assistant textual adapter (ATA) to learn textual features and transform them into the latent space of the acoustic encoder. With the help of the ATA module, the adaptation is achieved by fine-tuning the top layers of the acoustic encoder with the target domain text. Meanwhile, further improvement can be obtained by the integration with shallow fusion (SF). Adapted from LibriSpeech, experiments show that the proposed method can achieve averaged 29.7% relative WER reduction (WERR) compared with the unadapted baseline on WSJ, and 10.5% WERR compared to SF as well. Moreover, it also shows 15.4∼37.1% WERR for 10 GigaSpeech target domains test sets compared to the unadapted baseline, and also 6.5% WERR on average compared with SF.

Index Terms— text-only domain adaptation, connectionist temporal classification, end-to-end speech recognition

1. INTRODUCTION

End-to-end (E2E) speech recognition (ASR) takes audio feature sequences as the input and generates text sequences straightforward as the output, such as connectionist temporal classification (CTC) [1], attention-based encoder-decoder (AED) [2, 3, 4] and recurrent neural network transducer (RNN-T) [5]. Particularly, CTC is a non-autoregressive model which can infer the entire sequence at the same time efficiently instead of iterative computing depending on the previous prediction. However, when E2E ASR models come into use, a mismatch between the source and target domains could cause dramatic degradation in their performance.

To tackle this problem, many approaches for domain adaptation have been proposed. Conventional domain adaptation methods require speech-text paired data in the target domain, such as teacher-student learning [6], adversarial learning [7], adapter [8, 9] and the mixture of experts [10]. But it is not practical to obtain sufficient paired target domain data all the time. Text-only approaches can get rid of this data limitation by using text data of the target domain which are much easier to collect.

Two traditional approaches for text-only domain adaption, language model integration methods such as shallow fusion [11] and text-to-speech (TTS) [12], can alleviate the problem. Shallow fusion [11] introduces an external language model (LM), which is trained with text data in the target domain, to work with the E2E model during the beam search. But the external LM needs solitary training and takes extra memory while inferring. Meanwhile, TTS models can generate paired data from texts in the target domain. However, an effective TTS model is usually expensive in computation.

Recently, although there are research to develop the text data in AED [13] or RNN-T [14] architecture, few works have been done based on CTC architecture. As the decoder of AED or the predictor of RNN-T can be regarded as a fully linguistic part, efficient methods of text-only domain adaptation are hardly implemented on CTC architecture, as it only has an acoustic encoder. An early approach by Hiroaki [15] embeds acoustic and linguistic information in the same latent space based on intermediate CTC [16]. However, such a technique has its limitations, as it requires intermediate CTC [16] as the backbone model and large fine-tuned parameters for each target domain.

In this work, we proposed an efficient text-only domain adaptation method that simplifies the above adaptation procedure using a redesigned assistant textual adapter (ATA). Our two-stage adaptation method first trains the ATA module with CTC sequences predicted by the acoustic encoder in the source domain. This gives ATA the ability to learn the transformation from text to the acoustic encoder's latent features. Then, partial of the encoder layers are fine-tuned to learn the target domain linguistic information, with generated pseudo sequence from text data. Meanwhile, we also add a penalty function to avoid overfitting during adaptation. Furthermore, the final performance can be boosted by combining it with the

† Corresponding author

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

shallow fusion technique [11]. Experiments are conducted on the text-only adaptation from LibriSpeech [17] to WSJ [18] and GigaSpeech [19] and show advanced results compared with the baseline. Results show that our method improves the ability of the adaptive module to synthesize hidden features based on the text data effectively.

In the rest of the paper, we first review the Conformer-CTC architecture in Section 2. Then we describe our text-only domain adaptation approach in Section 3. Experimental results are shown to demonstrate the superiority of the proposed approach in two datasets in Section 4. Finally, we draw our conclusion in Section 5.

2. CONFORMER-BASED CTC

2.1. Conformer Encoder

The convolution-augmented transformer (conformer) [20] has shown its impressive performance in ASR. Firstly, a down-sampling module is passed through to reduce the audio sequence length from T to T'. Then the feature is fed into the consecutive conformer blocks, and each block structure is as follows:

$$\boldsymbol{H} = \boldsymbol{H} + \frac{1}{2}\text{FFN}(\boldsymbol{H}), \tag{1}$$

$$\boldsymbol{H} = \boldsymbol{H} + \text{Conv}(\boldsymbol{H} + \text{Att}(\boldsymbol{H})), \tag{2}$$

$$\boldsymbol{H} = \text{LayerNorm}(\boldsymbol{H} + \frac{1}{2}\text{FFN}(\boldsymbol{H})), \tag{3}$$

where FFN denotes the feed-forward modules, Att denotes the multi-head self-attention module and Conv denotes the convolution module. As a combination of the convolution neural network (CNN) and transformer, the conformer encoder has both advantages. It is adept at capturing the global relations within the context like transformers as well as can extract the local features well like CNN.

2.2. Connectionist Temporal Classification (CTC)

The E2E ASR system is to find a mapping from a speech feature sequence $\boldsymbol{O} = [\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T]$ to a token sequence $\boldsymbol{y} = [y_1, y_2, \ldots, y_L]$ where $\boldsymbol{o}_i$ is the acoustic feature (such as filter-bank), $y_i \in \mathcal{V}$ and $\mathcal{V}$ is the vocabulary set.

For the CTC-based ASR system, it is composed of a front-end encoder and a classifier. The classifier consists of a linear layer and a softmax function. Firstly, the front-end encoder gets speech input $\boldsymbol{O}$ and takes hidden features of the final layer $\boldsymbol{H}_{\text{final}}$ as the output. Then the output of the encoder will go through the linear classifier to produce $\boldsymbol{Z}$, a probability distribution of each candidate symbol at each time:

$$\boldsymbol{H}_{\text{final}} = \text{Encoder}(\boldsymbol{O}), \tag{4}$$

$$p(\boldsymbol{Z}|\boldsymbol{O}) = \text{Classifier}(\boldsymbol{H}_{\text{final}}). \tag{5}$$

Any valid CTC sequence $\boldsymbol{z}$ can be decoded from $\boldsymbol{Z}$. Since the length of $\boldsymbol{z}$ equals to that of the input sequence which is usually different from the label text's ($T \neq L$), CTC introduces a special token <B>into the candidate symbol list to map with silent or unrecognizable audio frame. In this way, the CTC model can bring out text sequence $\boldsymbol{y}$ of a flexible length L ($L \leq T$). A function $\mathcal{F}$ works to merge the continuously repetitive tokens and remove <B>from $\boldsymbol{z}$ to produce the final predicted text. Thus $\mathcal{F}^{-1}$ provides all the valid CTC sequences that lead to a certain text. The CTC loss follows a conditional independence assumption that the prediction at time t is not dependent on any other prediction.

$$\mathcal{L}_{CTC} = -\log p(\boldsymbol{y}|\boldsymbol{O}) \tag{6}$$

$$= -\log \sum_{\boldsymbol{z} \in \mathcal{F}^{-1}(\boldsymbol{y})} \prod_{t=1}^{T} p(\boldsymbol{z}_t|\boldsymbol{O}) \tag{7}$$

where z_t is the t-th token in $\boldsymbol{z}$.

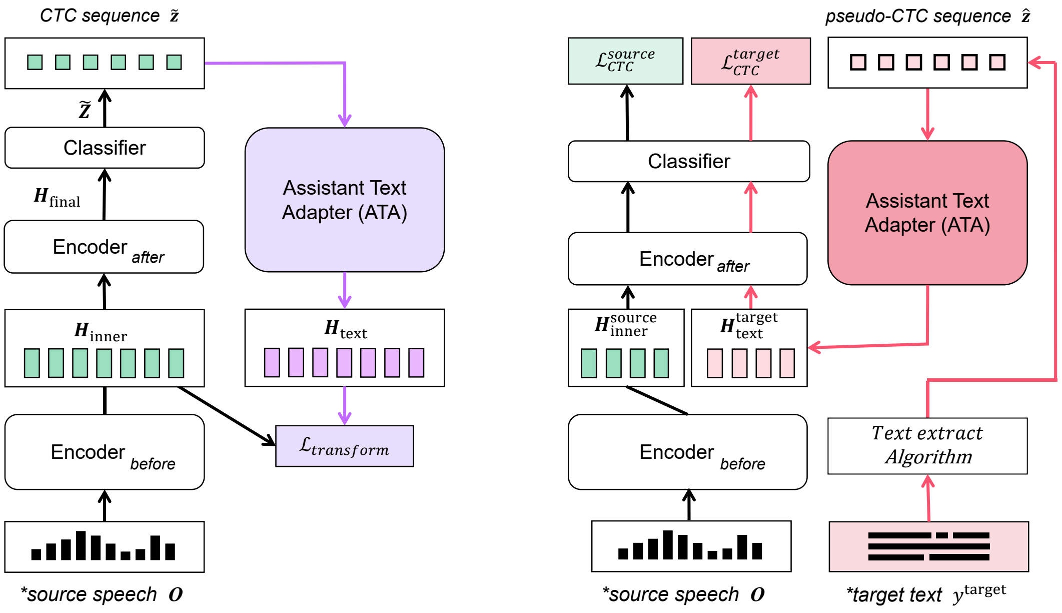
3. EFFICIENT TEXT-ONLY DOMAIN ADAPTATION FOR CTC

In this section, we present a two-stage text-only domain adaptation method for CTC-based ASR shown in Figure 1. In the first preparation stage, we incorporate an assistant textual adapter into the top layers of the acoustic encoder to transform text input into the hidden feature latent space. In the second adaptation stage, we adapt the top layers of the encoder to the target domain utilizing the unpaired text data, and then we evaluate the adapted model.

3.1. Textual Space Transformation with Assistant Textual Adapter

As we'd like to transform the textual input into the hidden feature latent space, there are two key points. The first is how to match the sequence length and the second is which latent layer is the most effective. As the latent features' length is equal to the CTC symbol sequence, which is also an inverse mapping from token distribution sequence $\boldsymbol{y}$ mentioned in Section 2.2. The extracted sequence $\tilde{\boldsymbol{z}}$ from the baseline is regarded as the extended textual input.

Illustrated in Figure 1(a), the Conformer encoder is divided into two parts Encoder$_{\text{before}}$ and Encoder$_{\text{after}}$, and the latent space after Encoder$_{\text{before}}$ is selected as the target of the transformation. Encoder$_{\text{before}}$ includes the bottom layers of the conformer encoder, which directly accepts the speech input $\boldsymbol{O}$ and extracts latent features $\boldsymbol{H}_{\text{inner}}$. Encoder$_{\text{after}}$ represents the other top layers. With the linear classifier following, it is where the final posterior probabilities $\log p(z)$ are pro-


CTC sequence $\tilde{z}$
$\tilde{Z}$
Classifier
H_{final}
Encoder after
H_{inner}
Encoder before
*source speech O
Assistant Text Adapter (ATA)
H_{text}
$\mathcal{L}_{transform}$
$\mathcal{L}_{CTC}^{source}$
$\mathcal{L}_{CTC}^{target}$
pseudo-CTC sequence $\hat{z}$
Classifier
Encoder after
Assistant Text Adapter (ATA)
$H_{\text{inner}}^{\text{source}}$
$H_{\text{text}}^{\text{target}}$
Encoder before
Text extract Algorithm
*source speech O
*target text y^{target}

Stage 1: Training the Assistant Text Adapter

Stage 2: Efficient Text-only Domain Adaptation

Fig. 1: The proposed two-stage text-only domain adaptation method. In stage one, we train an assistant text adapter (ATA) for textual space transformation where the source textual data is passed through the purple pipe. In stage two, parameters of Encoder$_{\text{after}}$ and classifier are tuned for adaptation, and the target textual data is passed through the red pipe with a source penalty $\mathcal{L}_{source}$.

duced:

$$\boldsymbol{H}_{\text{inner}} = \text{Encoder}_{\text{before}}(\boldsymbol{O}) \tag{8}$$

$$\boldsymbol{H}_{\text{final}} = \text{Encoder}_{\text{after}}(\boldsymbol{H}_{\text{inner}}), \tag{9}$$

$$\log p(\tilde{\boldsymbol{Z}}|\boldsymbol{O}) = \log \text{Classifier}(\boldsymbol{H}_{\text{final}}), \tag{10}$$

where $\tilde{\boldsymbol{Z}}$ has a shape of $T' \times (|\mathcal{V}|+1)$, and the greedy-decoded CTC sequence is $\tilde{\boldsymbol{z}} = \arg\max \log p(\tilde{\boldsymbol{Z}}|\boldsymbol{O})$. Then, the sequence $\tilde{\boldsymbol{z}}$ is fed into the assistant textual adapter and extracted as latent features $\boldsymbol{H}$:

$$\boldsymbol{H}_{\text{text}} = \text{ATA}(\tilde{\boldsymbol{z}}), \tag{11}$$

where the embedding layer is needed in ATA with traditional positional embedding the same as transformer [3]. Then ATA is optimized by minimizing the frame-level L2 norm between the latent features $\boldsymbol{H}_{\text{inner}}$ from speech and the latent features from text $\boldsymbol{H}_{\text{text}}$:

$$\mathcal{L}_{transform} = \frac{1}{T'} \sum_{t=1}^{T'} \sqrt{||\boldsymbol{H}_{\text{inner}}^{t} - \boldsymbol{H}_{\text{text}}^{t}||^2}), \tag{12}$$

where T' denotes the length of the latent features.

3.2. Text-Only Domain Adaptation with Assistant Textual Adapter

As mentioned above, the target domain text data have to be extracted to match the length of features $\boldsymbol{H}_{\text{inner}}$ and therefore we propose a text extraction algorithm in 1, inspired by [16, 15]. First, we prepare statistical information for the extraction, where p_b, p_{nb} are the probabilities of consecutive blanks and symbols in symbol sequences set $\boldsymbol{Z}$. $p_b(n)$, denotes the occurrence probability that n consecutive <B>occurs in the whole training dataset, and the symbol sequences set $\boldsymbol{Z}$ is generated using greedy search. Using the probabilities above, we generate the pseudo sequence $\hat{\boldsymbol{z}}$ based on the token sequence $\boldsymbol{y}^{\text{target}} = [y_1, y_2, \cdots, y_L]$ in the target domain text, where $\boldsymbol{y}^{\text{target}} = \mathcal{F}(\hat{\boldsymbol{z}})$). For example, we extract the text 'LOOK' in the target domain to a possible symbol sequence 'L <B>O <B>O <B>K K <B><B>'.

During the adaptation stage, the loss function is the same as Equation 7 where Encoder$_{\text{after}}$ and the classifier's parameters will be updated:

$$\mathcal{L}_{\text{target}} = \mathcal{L}_{CTC}^{\text{target}}(\boldsymbol{H}_{\text{text}}^{\text{target}}, \boldsymbol{y}^{\text{target}}). \tag{13}$$

Although the above Algorithm 1 can significantly enhance the robustness of the model when dealing with diverse patterns of CTC symbol sequences, it limits the original ability during text-only domain adaptation. We add the CTC loss from the source domain as a penalty:

$$\mathcal{L}_{\text{source}} = \mathcal{L}_{CTC}^{\text{source}}(\boldsymbol{O}^{\text{source}}, \boldsymbol{y}^{\text{source}}), \tag{14}$$

where the regular supervised learning is conducted with the speech and text data in the source domain to maintain the

Algorithm 1: Text Extraction for target domain text $\boldsymbol{y}^{\text{target}}$

```
Function Prepare(baseline, $\boldsymbol{O}$, $\boldsymbol{y}$):
    estimated sequence set $\boldsymbol{Z}$ := greedy(baseline, train data) ;
    $N_b(n)$ = Count(<B>, $\boldsymbol{Z}$) ;
    $N_{nb}(n)$ = $|\boldsymbol{Z}| - N_b(n)$ ;
    $p_b(n) = \frac{N_b(n)}{\sum_{n'} N_b(n')}$ ;
    $p_{nb}(n) = \frac{N_{nb}(n)}{\sum_{n'} N_{nb}(n')}$ ;
    return $p_b(n), p_{nb}(n)$ ;
$p_b(n), p_{nb}(n)$ = Prepare(baseline, $\boldsymbol{O}$, $\boldsymbol{y}$) ;
target_sequence $\hat{\boldsymbol{z}} = []$ ;
for $i \in [1, \cdots, L]$ do
    repeat
        $n$ = Sample($p_b$) ;
    until $i \neq 1$ & $n = 0$ & $y_{i-1} = y_i$ ;
    ;
    Append($\hat{\boldsymbol{z}}$, $n$, $<B>$) ;
    $n$ = Sample($p_{nb}$) ;
    Append($\hat{\boldsymbol{z}}$, $n$, $y_i$) ;
    return $\hat{\boldsymbol{z}}$ ;
end
```

model's ability of speech recognition. And finally the adaptation stage loss is computed as:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{target} + (1 - \alpha) \cdot \mathcal{L}_{source}, \tag{15}$$

where α is the hyper-parameter to adjust the weights of those two losses.

4. EXPERIMENTS

4.1. Experimental Setup

Our experiments are conducted on two different target datasets, WSJ [18] and GigaSpeech [19] . The source domain data is 960h LibriSpeech [17]. The first target domain is WSJ [18], WSJ collects its scripts from the Wall Street Journal and was recorded by the DARPA program to support the research on large vocabulary continuous speech recognition (LVCSR). And train-si-284 text data is used for adaptation and dev93/eval92 are used for evaluation. In order to verify the effectiveness of the method, we select GigaSpeech [19] as another target domain dataset, as it is a multi-domain ASR corpus. We choose the same setup as [13], where 5 YouTube domains (education, entertainment, news, people, science) and 5 different Podcast domains (arts, crime, health, people, science) are chosen and details are shown in Table 2. Each subdomain has a dev set of 5 hours and a test set of 10 hours for the following experiments.

The acoustic encoder of 12 Conformer blocks in the baseline model is divided in half, 6 blocks for the lower layers and the top layers each. In the adaptation from LibriSpeech to WSJ, the assistant textual adapter is set to be 4 blocks of Conformer, the size which outperforms the others in experiments on different ATA layers. In the adaptation from LibriSpeech to GigaSpeech, the ATA layer is set as 6. For acoustic features, 80-dim standard fbank [21] features are extracted with global-level cepstral mean and variance normalization from LibriSpeech, where SpecAugment [22] is used during training as augmentation. 5000 sentence pieces [23] are trained using LibriSpeech 960 hours paired text. We use the pre-trained Conformer-CTC models developed from ESPnet [24]. The subsampling layer is a 2-layer convolution with a downsampling rate of 4, and the number of conformer encoder layers is 12. Each encoder layer has 2048 linear units in each feed-forward module and 8 attention heads in each multi-head self-attention module. The kernel size of each convolution module is 31.

During decoding, a beam search size of 20 is chosen, and shallow fusion factor β is fixed to 0.8 if added. Word error rate (WER) (%) is reported over all evaluation sets.

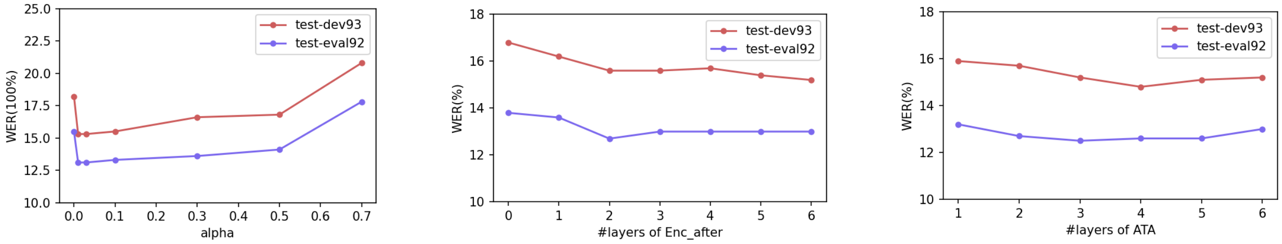
4.2. Text-only adaptation on target WSJ dataset

Model	dev93	eval92	average
baseline	17.5	14.1	15.8
baseline+SF	13.9	10.8	12.4
TDA [15]	-	-	12.2
TDA+SF [15]	-	-	12.0
proposed	14.8	12.6	13.7
proposed+SF	**12.0**	**10.2**	**11.1**

Table 1: Performance (WER) (%) comparison of our proposed method and other advancing methods on WSJ dev93 and eval92 test sets.

In this section, we aim to validate the proposed method, explore the best architecture of our model, and evaluate its performance on the adaptation from LibriSpeech to WSJ.

Firstly, we conduct experiments with varying values of hyper-parameter α and different block numbers of the acoustic encoder or the assistant textual adapter (ATA) and tunable encoder $\text{Encoder}_{\text{after}}$ respectively in Figure 2. In Figure 2a, we first validate the effectiveness of the hyper-parameter α mentioned in Equation 15 in stage two. When α equals 0, which means there is no proposed method applied, the performance degraded badly, which is caused by the over-fitting on the source domain. From the figure, WER increase can be observed as α grows from 0.1 to 0.7, and the performance is even worse than the baseline when $\alpha = 0.7$. It can be seen that the basic ability of the ASR model could be impaired if adaptation is too overwhelming. Therefore, smaller α values are explored when $\alpha \in [0, 0.1]$, and results show that our



(a) Different α factor in Equation 15.

(b) Different tunable $\text{Encoder}_{\text{after}}$ layers.

(c) Different tunable ATA layers.

Fig. 2: Ablation Studies on different hyper-parameters and architectures of the proposed method.

Dataset / Model	YouTube (Y)					Podcast (P)				
	education	entertainment	news	people	science	arts	crime	health	people	science
#words (M)	17.33	3.4	3.74	4.24	3.41	0.69	1.17	1.44	1.38	4.08
baseline	15.7	35.0	25.8	27.6	28.7	19.6	19.4	24.9	26.0	24.5
+SF	12.8	30.7	20.2	22.9	21.9	17.3	15.0	19.4	21.7	16.6
+proposed	15.1	34.2	24.3	25.7	28.2	18.4	17.8	23.0	23.5	23.0
+proposed+SF	**12.1**	**29.6**	**19.1**	**21.3**	**21.0**	**16.3**	**13.7**	**17.9**	**19.5**	**15.4**

Table 2: Statistics and performance (WER) (%) comparison on 10 different target domains of GigaSpeech. The proposed method and the target LM are trained separately using only target domain text.

method brings out a decent optimization when $\alpha = 0.01$ with over 10% relative WER reduction (WERR).

To reduce the computational cost, we explore the best architecture with a minimum fine-tuning range in the acoustic encoder $\text{Encoder}_{\text{after}}$ illustrated in Figure 2b, and the classifier layer is always tuned. The results show that the performance of the model generally gets better when the number of layers increases, yet the improvement is limited when the number is larger than 3. Although we choose the last 6 layers of $\text{Encoder}_{\text{after}}$ to be fine-tuned, the acoustic encoder has the potential for further narrowing down its tunable parameters while remaining a competitive performance.

Although the ATA module will not join in the inference process, minimizing its size will reduce the computation cost during the adaptation. The performance of various model scales of ATA is shown in Figure 2c. It shows that increasing ATA scale can lead to slight improvements in accuracy and ATA with more than four layers have similar performance. four-layer ATA is chosen for the following experiments.

Finally, we evaluate the performance of our proposed method in Table 1. Compared with the unadapted baseline, the final proposed method shows a relative WER reduction of 13.3%. It means that our method can provide effective adaptation with text data alone in the target domain. Although the plain method we proposed lags a little bit compared with shallow fusion (SF) or text-only adaptation (TDA)[15] alone, it achieves the best results when combined with SF. With an external language model applied, our model not only outperforms the traditional SF method (baseline+SF) with WERR of 10.5% but also exceeds the result (TDA+SF) by 7.5% WERR.

4.3. Text-only adaptation on various GigaSpeech datasets

To further verify the effectiveness of the proposed method, 10 subsets of GigaSpeech are selected as the target domain adapted from LibriSpeech to make the quantitative evaluation. Optimization in recognition accuracy has shown in all 10 subsets when the proposed method is applied. Compared with the unadapted baseline, the final '+proposed+SF' method can achieve 15.4∼37.1% which is such a huge improvement. Even compared with shallow fusion (baseline+SF), our model can perform better by 6.5% WER reduction. Also we find that arts of podcast dataset improvement is limited compared to the others, as it has such a small text scale. And consistent improvements can be observed when the text scale (#words) grows.

5. CONCLUSION

In this work, we propose an efficient text-only domain adaptation method with the help of the assistant textual adapter (ATA). The proposed method exploits the differences between linguistics and other latent features inside the encoder. We first train the ATA module in the source domain and then fine-tune partial layers of the encoder to the target domain with only text data. The model adapted by our method needs no more extra parameters during the inference, which is quite potential to be used in mobile devices. What's more, the number of parameters that need to be updated during the text-only adaptation is kept to a minimum. We explore the effectiveness of the proposed method on the adaptation from LibriSpeech to WSJ/GigaSpeech datasets, which both obtain great performance improvements compared with the baseline.

6. ACKNOWLEDGEMENT

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

7. REFERENCES

[1] Alex Graves and Alex Graves, "Connectionist temporal classification," *Supervised sequence labelling with recurrent neural networks*, pp. 61–93, 2012.

[2] William Chan, Navdeep Jaitly, et al., "Listen, attend and spell," *arXiv:1508.01211 [cs, stat]*, Aug. 2015.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP 2018*. IEEE, 2018, pp. 5884–5888.

[5] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[6] Vimal Manohar, Pegah Ghahremani, Daniel Povey, and Sanjeev Khudanpur, "A teacher-student learning approach for unsupervised domain adaptation of sequence-trained asr models," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 250–257.

[7] Zhong Meng, Jinyu Li, et al., "Adversarial teacher-student learning for unsupervised domain adaptation," in *Proc. ICASSP*, 2018, pp. 5949–5953.

[8] Xun Gong, Yizhou Lu, et al., "Layer-wise fast adaptation for end-to-end multi-accent speech recognition," in *Proc. Interspeech*, 2021, pp. 1274–1278.

[9] Yanmin Qian, Xun Gong, and Houjun Huang, "Layer-wise fast adaptation for end-to-end multi-accent speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2842–2853, 2022.

[10] Yizhou Lu, Mingkun Huang, et al., "Bi-encoder transformer network for mandarin-english code-switching speech recognition using mixture of experts," in *Proc. Interspeech*, Oct. 2020, pp. 4766–4770.

[11] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhijeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *ICASSP 2018*. IEEE, 2018, pp. 1–5828.

[12] Yan Deng, Rui Zhao, Zhong Meng, Xie Chen, Bing Liu, Jinyu Li, Yifan Gong, and Lei He, "Improving rnn-t for domain scaling using semi-supervised training with neural tts.," in *Interspeech*, 2021, pp. 751–755.

[13] Xun Gong, Wei Wang, Hang Shao, Xie Chen, and Yanmin Qian, "Factorized aed: Factorized attention-based encoder-decoder for text-only domain adaptive asr," in *ICASSP 2023*. IEEE, 2023, pp. 1–5.

[14] Xie Chen, Zhong Meng, et al., "Factorized neural transducer for efficient language model adaptation," in *Proc. ICASSP*, 2022, pp. 8132–8136.

[15] Hiroaki Sato, Tomoyasu Komori, Takeshi Mishima, Yoshihiko Kawai, Takahiro Mochizuki, Shoei Sato, and Tetsuji Ogawa, "Text-only domain adaptation based on intermediate ctc," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2022, vol. 2022, pp. 2208–2212.

[16] Jaesong Lee and Shinji Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *ICASSP 2021*. IEEE, 2021, pp. 6224–6228.

[17] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP 2015*. IEEE, 2015, pp. 5206–5210.

[18] Douglas B Paul and Janet Baker, "The design for the wall street journal-based csr corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[19] Guoguo Chen, Shuzhou Chai, et al., "GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio," in *Proc. Interspeech*, 2021, pp. 3670–3674.

[20] Anmol Gulati, James Qin, et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, May 2020, pp. 5036–5040.

[21] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 24–29.

[22] Daniel S. Park, William Chan, et al., "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech*, pp. 2613–2617, Sept. 2019.

[23] Philip Gage, "A new algorithm for data compression," *C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[24] Shinji Watanabe, Takaaki Hori, et al., "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, Mar. 2018.