APSIPA Transactions on Signal and Information Processing, 2023, 12, e27

Original Paper

Speech-and-Text Transformer: Exploiting Unpaired Text for End-to-End Speech Recognition

Qinyi Wang[1*], Xinyuan Zhou[2] and Haizhou Li[1,3]

[1] *National University of Singapore, Singapore*
[2] *Shanghai Normal University, Shanghai, China*
[3] *The Chinese University of Hong Kong (Shenzhen), Shenzhen, China*

ABSTRACT

End-to-end automatic speech recognition (ASR) models are typically data-hungry, which depend on a large paired speech-text dataset for the models to be effective. It remains an active area how to increase the linguistic competence of such ASR models with unpaired text data. The conventional techniques that employ an external language model (LM) suffer from high decoding complexity. Pre-training methods have problems of catastrophic forgetting and model capacity gap between the pre-trained modules and the actual tasks. This paper introduces a speech-and-text Transformer to leverage unpaired text and address the above issues. The decoder of the proposed speech-and-text Transformer contains three parallel branches to learn strong text representations from unpaired text and reduce the mismatch between the speech and text representations. An on-demand dual-modality attention mechanism is proposed to automatically select one or two modalities to learn from. Besides, we introduce a novel alternate training algorithm to load speech and text batches alternately and accumulate their gradients. The proposed model is trained with an auxiliary language modeling task. Intra-domain and cross-domain speech recognition experiments are conducted on AISHELL-1, LibriSpeech, and WenetSpeech corpora.

[*] Corresponding author: Qinyi Wang, qinyi@u.nus.edu.

Received 03 January 2023; Revised 13 April 2023
ISSN 2048-7703; DOI 10.1561/116.00000001

Results show competitive performance to the conventional shallow fusion method with negligible computation overheads during inference.

Keywords: Speech recognition, language model, unpaired data, transformer, semi-supervised learning

1 Introduction

Deep learning has significantly advanced the landscape of speech recognition research with the scalability and prediction power of deep neural networks. The main goal of speech recognition is to build a model to infer the text sequence $\boldsymbol{Y}$ from the acoustic feature sequence $\boldsymbol{X}$. Traditionally, a statistical automatic speech recognition (ASR) system contains two major components – an acoustic model (AM) for estimating the likelihood of a sequence of speech features given a sequence of text tokens $P(\boldsymbol{X}|\boldsymbol{Y})$ and a language model (LM) for evaluating the prior probability of the text sequence $P(\boldsymbol{Y})$ [28, 30]. The AM component finds the possible text sequences that match the input acoustic features, while the LM component imposes linguistic constraints on these selected text sequences. The AM-LM decoding mechanism is formulated under the hidden Markov modeling (HMM) framework [64]. While the HMM-based ASR model is not as effective as the end-to-end ASR model, the former employs a modular architecture that allows the AM and LM to be trained separately on the unpaired dataset.

The end-to-end (E2E) ASR model directly maps acoustic sequence to label sequence, obviating the need to train different modeling components with separate datasets. There are three prevailing types of end-to-end speech recognition architecture: connectionist temporal classification (CTC) [20, 24], recurrent neural network Transducer (RNN-T) [19, 21], and attention-based encoder-decoder (AED) [1, 10]. The early AED models use the recurrent neural network (RNN) as the building block to realize the encoder-decoder framework. Therefore, they are referred to as the RNN-based encoder-decoder hereafter. One of the latest advances of the attention-based encoder-decoder approaches is the adoption of Transformer [14, 56], which replaces all recurrent connections in the early AED models with the self-attention mechanism to capture long-range dependence and allow for parallelization. This model is referred to as the Transformer-based encoder-decoder hereafter. Augmented with CNN, Conformer-based ASR models have achieved state-of-the-art performance on most ASR benchmarks [22].

Despite overwhelming success, one critical problem of E2E solutions is that they are *data-hungry*, i.e., it requires thousands of hours of labeled speech for them to be effective [39]. The performance of an E2E ASR system may dramatically deteriorate as the amount of paired speech-text training data reduces [50]. Traditional module-based ASR systems contain a powerful LM, which is separately trained on massive text-only data to inject domain-specific linguistic knowledge into the ASR system and enhance the fluency of generated texts. In contrast, E2E ASR models entirely rely on paired speech-text training data, that are not always available. Therefore, the scope of application of the end-to-end ASR models is highly constrained. How to enhance the linguistic competence of E2E speech recognition systems with abundant unpaired text data is still an open research problem.

A common strategy to improve the linguistic competence of an E2E ASR system is to leverage the knowledge contained in an LM trained with unpaired text data. Shallow fusion [12] is the most popular approach in this direction, which applies the pre-trained LM during ASR's decoding stage in a post-processing beam search process. The use of pre-trained LM during ASR's training stage has also been investigated in techniques such as cold fusion [52], component fusion [51], and memory attentive fusion [29]. One crucial problem of those fusion techniques is that the additional LM module increases the model's computational complexity and decoding time, making it a great challenge to deploy them in real-time applications [27]. Liu *et al.* [38] use a Criticizing LM to distinguish the real text (both paired and unpaired text data) from the fake text generated by the ASR system and force the ASR to generate transcriptions of better quality. However, the adversarial training method often leads to overfitting as the ASR model might overly depend on the output distribution of an LM and less on the acoustic evidence.

Instead of using a separate LM, it has also been studied to pre-train end-to-end speech recognition systems directly with text data [6, 62, 67]. Gao *et al.* [18] treat the decoder of a Transformer-based encoder-decoder model as a neural LM and pre-train it using pure text data. After pre-training, the whole ASR model is fine-tuned with paired speech-text data. Similarly, Deng *et al.* [13] design a one-cross decoder to relax its dependence on acoustic inputs for a Transformer-based encoder-decoder model and initialize the encoder and decoder of the system with pre-trained AM and LM separately. Fan *et al.* [16] pre-train the encoder and decoder of a Transformer-based encoder-decoder model with unpaired audio data and synthesized speech-text data separately, then fine-tune the whole system with genuine paired data.

There are two significant issues with these pre-training methods. One is *catastrophic forgetting*, i.e., a model forgets previously learned knowledge when learning new information. During the fine-tuning stage, the pre-trained ASR models learn speech-text alignments at the price of forgetting linguistic knowledge learned in the previous stage. As a result, text-derived knowledge is not fully utilized in such pre-trained ASR models. Another issue lies in the model capacity gap between the powerful pre-trained LM part and the randomly initialized acoustic part of the E2E model at the beginning of the fine-tuning stage. This model capacity gap can cause mismatches between their generated speech and text representations, degrading the learning of speech-text alignment in the fine-tuning stage.

In this paper, we propose a novel encoder-decoder ASR architecture, called speech-and-text Transformer, to acquire linguistic knowledge directly from the unpaired text for E2E ASR systems and address the above problems. We make three significant amendments to the decoder architecture of the vanilla Transformer: (1) Instead of having one single branch with two attention modules connected in series, our decoder has three parallel branches – a deep acoustic branch for capturing multiple levels of acoustic abstractions, a speech decoding branch for learning speech-text alignment from previous deep acoustic states and decoder states, and an inner-LM branch for acquiring linguistic knowledge from unpaired text data. (2) In the speech decoding and inner-LM branches, we introduce an on-demand dual-modality attention mechanism to enable the model to automatically learn from one or two modalities, depending on their availability. (3) We share parameters between the speech decoding and inner-LM branch to leverage the strong text representation learned in the inner-LM branch for speech decoding and reduce model size.

To overcome catastrophic forgetting and balance the model capabilities between the inner-LM and other branches of the decoder, we introduce an alternate training algorithm that involves loading several batches of text data and a batch of paired speech-text data in an alternating manner. Besides, we use a *text gradient accumulation* mechanism in the alternate training algorithm to accumulate gradients from the text and speech-text batches to simulate a big batch for stabilizing training. Another highlight of this algorithm is the use of text ratio, which controls the proportion of the unpaired text to paired speech-text data when updating the parameters. We employ multi-objective learning to optimize the speech-and-text Transformer with a main hybrid CTC/attention objective and an auxiliary language modelling objective. By controlling the LM training weight in the joint loss function, we balance the contributions from the paired speech-text data and the unpaired text, thus avoiding overfit to one of the two training data.

The contributions of this work are summarized as follows.

- We introduce a unified end-to-end speech recognition model, namely speech-and-text Transformer, that is designed to enhance its language modeling ability directly from unpaired text data.
- We overcome catastrophic forgetting of pre-training methods and close the model capacity gap within the ASR model by using a novel alternate training algorithm.
- Our methods achieve comparable performance to the conventional shallow fusion method in both in-domain and out-of-domain speech recognition experiments.

- Our methods only incur negligible computational overhead during inference, making it possible to deploy our model to real-time applications.

The rest of this paper is organized as follows. Section 2 reviews the background of the Transformer-based encoder-decoder ASR. The architecture and training schemes of the proposed speech-and-text Transformer model are introduced in Section 3. Section 4 describes related work. Experiment setups and results are presented in Sections 5 and 6. Section 7 provides analyses and some insights into our experiment results. Finally, Section 8 concludes the paper and discusses future work.

2 Preliminaries

We first discuss the fundamentals relevant to the proposed Transformer-based encoder-decoder ASR architecture.

2.1 Transformer-based ASR Model

Transformer draws dependencies between inputs and outputs by relying entirely on attention mechanisms. In Figure 1, we illustrate a standard Transformer-based encoder-decoder speech recognition architecture. Let's define a T-length input acoustic feature sequence as $\boldsymbol{X} = \{x_1, \ldots, x_t, \ldots, x_T\}$ and an N-length output text token sequence as $\boldsymbol{Y} = \{y_1, \ldots, y_n, \ldots, y_N\}$. We also define $\Theta = \{\theta_{enc}, \theta_{dec}\}$ as the set of trainable parameters of the Transformer model, with θ_{enc} being the parameters of the encoder and θ_{enc} being the parameters of the decoder.

The Transformer encoder is composed of a stack of J identical encoder blocks, each of which takes input from the hidden states of its previous block and generates higher-level encoder hidden states. Together with an acoustic embedding and a positional encoding module, these encoder blocks extract information from the input acoustic feature sequence $\boldsymbol{X}$ and convert it to a high-level acoustic representation $\boldsymbol{H}$ as follows,

$$\boldsymbol{X}^{(0)} = \text{PosEnc}(\text{AousticEmb}(\boldsymbol{X})), \tag{1}$$

$$\boldsymbol{X}^{(j)} = \text{EncBlock}(\boldsymbol{X}^{(j-1)}, \theta_{enc}), \tag{2}$$

$$\boldsymbol{H} = \text{LayerNorm}(\boldsymbol{X}^{(J)}), \tag{3}$$

where EncBlock$(\cdot)$ is the Transformer encoder block containing a multi-head attention sub-layer and a position-wise feed-forward network sub-layer, with a residual connection and a layer normalization adopted for each sub-layer.

The Transformer decoder also consists of a stack of K identical decoder blocks, each of which takes input from the hidden states of its previous decoder

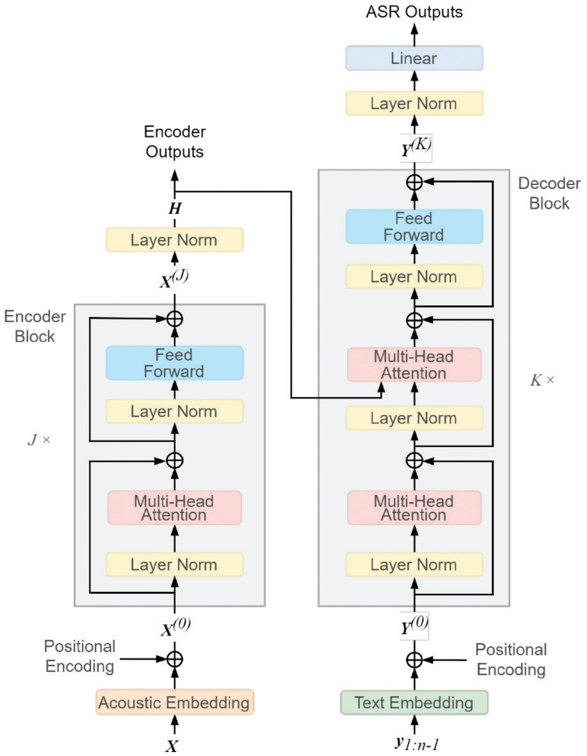

Figure 1: The architecture of a Transformer-based encoder-decoder ASR.

block and the same acoustic representation $\boldsymbol{H}$ obtained from the encoder to yield higher-level decoder hidden states. The Transformer decoder calculates the probability of the next text token given the whole acoustic feature sequence and its previous text token sequence $P(y_n|\boldsymbol{X}, \boldsymbol{y}_{1:n})$ as follows,

$$\boldsymbol{Y}^{(0)} = \text{PosEnc}(\text{TextEmb}(\boldsymbol{y}_{1:n-1})), \tag{4}$$

$$\boldsymbol{Y}^{(k)} = \text{DecBlock}(\boldsymbol{Y}^{(k-1)}, \boldsymbol{H},\ \theta_{dec}), \tag{5}$$

$$P(y_n|\boldsymbol{X}, \boldsymbol{y}_{1:n}) = \text{Softmax}(\text{Linear}(\text{LayerNorm}(\boldsymbol{Y}^{(K)}))), \tag{6}$$

where DecBlock$(\cdot)$ is the Transformer decoder block containing two multi-head attention sub-layers and a position-wise feed-forward network sub-layer, with a residual connection and a layer normalization adopted for each sub-layer.

2.2 Multi-Head Attention

Multi-head attention is the core module of Transformer, which allows the model to acquire dependency information from multiple representation sub-spaces jointly. The left part of Figure 2 illustrates a multi-head attention module that employs the scaled dot-product attention function.

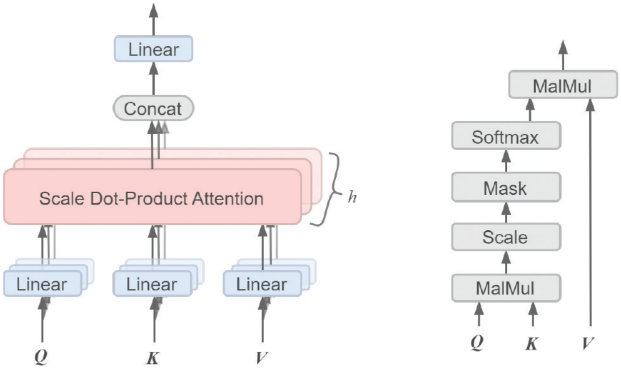

Figure 2: (left) A multi-head attention module that adopts scaled dot-product attention as its attention function. (right) Scaled dot-product attention.

Scaled dot-product attention is the most common choice for the attention function used in the multi-head attention mechanism. Define $\boldsymbol{Q} \in \mathbb{R}^{l_k \times d_{model}}$, $\boldsymbol{K} \in \mathbb{R}^{l_k \times d_{model}}$ and $\boldsymbol{V} \in \mathbb{R}^{l_v \times d_{model}}$ as the query, key, and values, where l_* are the lengths of these matrices and d_{model} is the output dimension of the Transformer model. Normally, $l_k = l_v$. As shown on the right side of Figure 2, the scaled dot-product attention function is formulated as follows,

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V}, \tag{7}$$

where dividing the dot products by $\sqrt{d_k}$ is to ensure the generated attention score has variance 1.

In the multi-head attention mechanism, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are first projected into h sub-spaces by using different linear layers, with h being the number of heads. The attention function $\text{Attention}(\cdot)$ is then performed h times to extract information from h representation sub-spaces. Lastly, the calculated attention scores are concatenated and projected to yield the final attention score. The multi-head attention mechanism is formulated as follows,

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h \boldsymbol{W}^O, \tag{8}$$

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V), \tag{9}$$

where $\boldsymbol{W}^*$ are the learned linear projection matrices, with $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_\text{model} \times d_k}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d_\text{model} \times d_k}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d_\text{model} \times d_v}$, $\boldsymbol{W}^O \in \mathbb{R}^{d_o} \times d_\text{model}$, and d_model is the output dimension of the Transformer model. Usually, $d_k = d_v = d_{model}/h$ and $d_o = h \cdot d_v = d_{model}$.

3 Speech-and-Text Transformer

We propose a neural architecture where we directly use unpaired text data in ASR training to enhance the linguistic competence of a Transformer-based encoder-decoder speech recognizer.

3.1 Architecture

The overall architecture of the proposed speech-and-text Transformer is shown in Figure 3. The speech-and-text Transformer employs the same encoder structure as the vanilla Transformer-based encoder-decoder model. We denote the encoder output as $\boldsymbol{H}^{(0)}$ instead of $\boldsymbol{H}$ to distinguish different levels of acoustic representations in the decoder. The decoder consists of a stack of

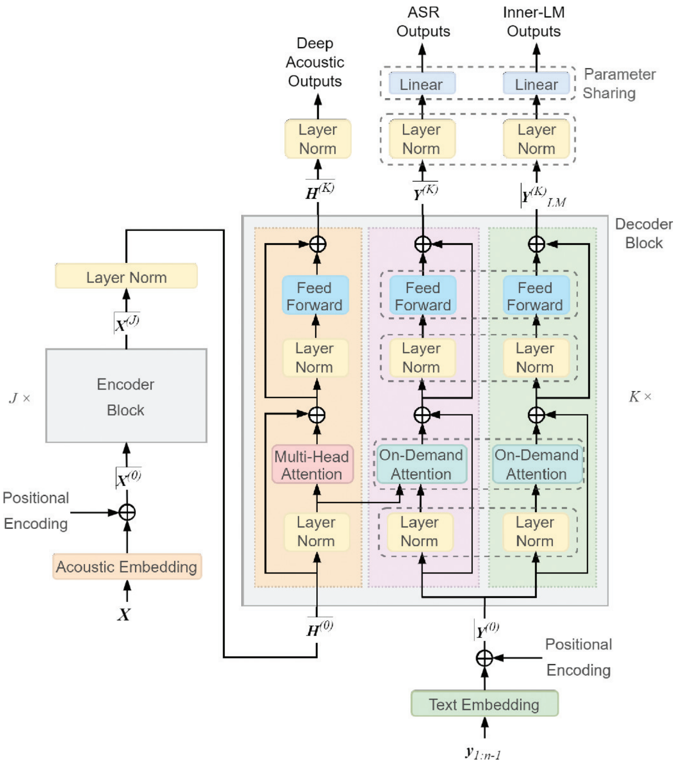

Figure 3: The architecture of the proposed speech-and-text Transformer. The orange area inside the decoder block indicates the deep acoustic branch, the pink area inside the decoder block indicates the speech decoding branch, and the green area inside the decoder block indicates the inner-LM branch. Modules inside the grey-dotted rounded rectangle share the same set of parameters.

K identical decoder blocks. Each decoder block contains three branches of different roles in parallel to facilitate the learning of speech-and-text alignments: a deep acoustic branch, a speech decoding branch, and an inner-LM branch. Define STDecBlock$(\cdot)$ as the decoder block of the speech-and-text Transformer and $\theta_{dec} = \{\theta_{dec\_s}, \theta_{dec\_d}, \theta_{dec\_t}\}$ as the set of trainable parameters of the speech-and-text decoder, with θ_{dec_s} being the parameters of the deep acoustic branch, θ_{dec_d} being the parameters of the speech decoding branch, and θ_{dec_t} being the parameters of inner-LM branch.

(1) Deep Acoustic Branch: In the vanilla Transformer decoder, the same encoder output is used as the acoustic representation to establish the speech-text alignments for all decoder blocks. However, this fixed acoustic state is considered a relatively low-level abstraction than the various text representations generated in the decoder blocks. In order to help the model to learn better alignments with acoustic and text representations of the same abstraction level, we adopt deep acoustic structure (DAS) [68] in the deep acoustic branch for extracting deeper levels of acoustic representations in the decoder blocks. The deep acoustic branch contains a multi-head attention sub-layer and a feed forward sub-layer. The deep acoustic branch generates new deep acoustic states $\boldsymbol{H}^{(k)}$ by attending to the deep acoustic states from its previous block as follows,

$$\boldsymbol{H}^{(k)} = \mathrm{STDecBlock}(\boldsymbol{H}^{(k-1)}, \theta_{dec\_s}). \tag{10}$$

(2) Speech Decoding Branch: The speech decoding branch is responsible for predicting the next text token from deep acoustic representations and previously decoded text tokens. This requires the model to understand the dependency between the audio and text sequence and within the text sentence itself. We use a novel on-demand dual-modality attention module that attends to two modalities simultaneously to generate decoder states. This attention mechanism is detailed in the next subsection. The speech decoding branch is composed of an on-demand multi-modality attention sub-layer and a feed forward sub-layer. The speech decoding branch generates decoder states $\boldsymbol{Y}^{(k)}$ by attending to the deep acoustic states and the decoder states from its previous block as follows,

$$\boldsymbol{Y}^{(k)} = \mathrm{STDecBlock}(\boldsymbol{Y}^{(k-1)}, H^{(k-1)}, \theta_{dec\_s}, \theta_{dec\_d}), \tag{11}$$

and the probability of the next text token given the whole acoustic feature sequence and its previous text token sequence is calculated as follows,

$$P(y_n|\boldsymbol{X}, \boldsymbol{y}_{1:n}) = \mathrm{Softmax}(\mathrm{Linear}(\mathrm{LayerNorm}(\boldsymbol{Y}^{(K)}))). \tag{12}$$

(3) Inner-LM Branch: The inner-LM branch is an acoustic-independent branch that aims to take pure text as input to gain linguistic knowledge for our model. It has the same modular structure as the speech decoding branch,

except that there is only text input for the on-demand dual-modality attention module. The same modular structure enables easy linguistic knowledge transfer between the speech decoding and inner-LM branch through parameter sharing. We share parameters among all modules between the speech decoding and inner-LM branch to leverage the high-quality text representations generated by the inner-LM for recognizing speech and reducing model size. The calculation of inner-LM states $\boldsymbol{Y}_{LM}^{(k)}$ is formulated as follows,

$$\boldsymbol{Y}_{LM}^{(k)} = \text{STDecBlock}(\boldsymbol{Y}^{(k-1)}, \theta_{dec\_t}). \tag{13}$$

After obtaining the final LM hidden state $\boldsymbol{Y}_{LM}^{(K)}$, the probability of the next token conditioned on the previous token history is calculated as follows,

$$P(y_n|\boldsymbol{y}_{1:n}) = \text{Softmax}(\text{Linear}(\text{LayerNorm}(\boldsymbol{Y}_{LM}^{(K)}))). \tag{14}$$

The inner-LM branch is only used during training for introducing external linguistic knowledge to the model with unpaired text data. During decoding, only the deep acoustic and speech decoding branches are responsible for utterance decoding, which reduces the model's inference cost.

3.2 Multi-Input Multi-Head Attention

The multi-head attention mechanism in the vanilla Transformer model can only learn the dependencies between two modalities or inside one modality itself at different times. However, we hope to learn the above two relationships simultaneously in the speech decoding branch and only the latter dependency in the inner-LM branch of our model. Therefore, we propose the on-demand dual-modality attention (ODDMA) mechanism that has the flexibility of mapping a query and two sets of key-value pairs or a query and one set of key-value pairs into an output, depending on the presence of the source sequence.

The left part of Figure 4 illustrates an ODDMA module. Basically, ODDMA is a multi-head attention mechanism with five input matrices - a query and two sets of key-value pairs: target queries $\boldsymbol{Q}_t \in \mathbb{R}^{l_{k_t} \times d_{model}}$, target keys $\boldsymbol{K}_t \in \mathbb{R}^{l_{k_t} \times d_{model}}$, target values $\boldsymbol{V}_t \in \mathbb{R}^{l_{v_t} \times d_{model}}$, source keys $\boldsymbol{K}_s \in \mathbb{R}^{l_{k_s} \times d_{model}}$, and source values $\boldsymbol{V}_s \in \mathbb{R}^{l_{v_s} \times d_{model}}$. Since all target matrices come from the same text representation and all source matrices come from the same speech representation, we denote $l_{k_t} = l_{v_t} = l_t$ and $l_{k_s} = l_{v_s} = l_s$. Like the standard multi-head attention mechanism, ODDMA linearly projects each input matrix into h different sub-spaces. After that, these projected representations go through our proposed on-demand dual-modality scaled dot-product attention function to learn dependencies from two modalities as requested flexibly. Finally, the attention scores of all heads are combined and projected to get

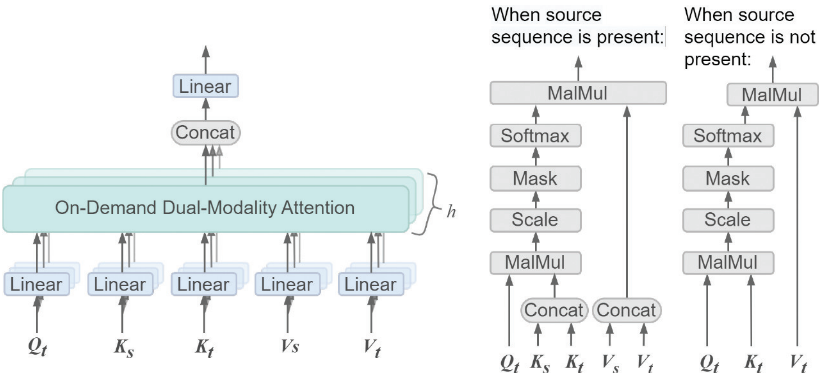

Figure 4: (left) An on-demand dual-modality attention module that adopts on-demand dual-modality scaled dot-product attention. (right) On-demand dual-modality scaled dot-product attention.

the final value. The multi-input multi-head attention is formulated as follows,

$$\mathrm{ODDMA}(\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t, \boldsymbol{K}_s, \boldsymbol{V}_s) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)\boldsymbol{W}^O, \tag{15}$$

$$\mathrm{head}_i = \mathrm{OnDemandAttention}(\boldsymbol{Q}_t\boldsymbol{W}_i^{Q_t}, \boldsymbol{K}_t\boldsymbol{W}_i^{K_t}, \boldsymbol{V}_t\boldsymbol{W}_i^{V_t}, \boldsymbol{K}_s\boldsymbol{W}_i^{K_s}, \boldsymbol{V}_s\boldsymbol{W}_i^{V_s}), \tag{16}$$

where W^* are the learned projection matrices, with $\boldsymbol{W}_i^{Q_t} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{k_t}}$, $\boldsymbol{W}_i^{K_t} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{k_t}}$, $\boldsymbol{W}_i^{V_t} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{v_t}}$, $\boldsymbol{W}_i^{K_s} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{k_s}}$, $\boldsymbol{W}_i^{V_s} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{v_s}}$, $\boldsymbol{W}^O \in \mathbb{R}^{d_o \times d_{\mathrm{model}}}$, and $\mathrm{OnDemandAttention}(\cdot)$ is our introduced on-demand dual-modality scaled dot-product attention function. In this work, we set $d_{k_t} = d_{v_t} = d_{k_s} = d_{v_s} = d_{\mathrm{model}}/h = d$ and $d_o = h \cdot d = d_{\mathrm{model}}$.

The on-demand dual-modality scaled dot-product attention function performs scaled dot-product attention on a concatenation of source and target sequences or on the target sequence itself, depending on the availability of these two modalities. The right part of Figure 4 shows the two cases of the on-demand dual-modality scaled dot-product attention. The on-demand dual-modality scaled dot-product attention function is formulated as follows,

$$\begin{aligned}&\mathrm{OnDemandAttention}(\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t, \boldsymbol{K}_s, \boldsymbol{V}_s)\\ &\quad = \begin{cases} \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_t\boldsymbol{K}_c^T}{\sqrt{d}}\right)\boldsymbol{V}_c, & \text{source sequence is present} \\ \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_t\boldsymbol{K}_t^T}{\sqrt{d}}\right)\boldsymbol{V}_t, & \text{source sequence is not present} \end{cases},\end{aligned} \tag{17}$$

where $\boldsymbol{K}_c \in \mathbb{R}^{(l_t+l_s)\times d}$ is the concatenation of $\boldsymbol{K}_t$ and $\boldsymbol{K}_s$, and $\boldsymbol{V}_c \in \mathbb{R}^{(l_t+l_s)\times d}$ is the concatenation of $\boldsymbol{V}_t$ and $\boldsymbol{V}_s$. During ASR decoding, the concatenation operations entail minimal computation costs.

By sharing the target-related transformations $\boldsymbol{W}_i^{Q_t}$, $\boldsymbol{W}_i^{K_t}$ and $\boldsymbol{W}_i^{V_t}$ for the above two cases, ODDMA can take advantage of the superior text representations learned from pure text data to help establish speech-text alignments from paired data. To prevent the decoder from looking ahead at the future part of the target sentence when predicting the next word in both cases, we apply masks on the attention scores of each head that corresponds to faulty connections.

3.3 Training and Inference

We employ multi-objective learning to train the speech-and-text Transformer. The main task is to minimizes the Kullback-Leibler divergence loss $\mathcal{L}_{att}$ between the predicted conditional probability $P_{att}(\boldsymbol{Y}|\boldsymbol{X})$ and the true label distribution. This objective is optimized on a speech corpus $\mathcal{D}_P$ consisting of acoustic feature and transcription pairs $(\boldsymbol{X}, \boldsymbol{Y})$, where the conditional probability $P_{att}(\boldsymbol{Y}|\boldsymbol{X})$ is factorized as follows for an utterance of N tokens,

$$P_{att}(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{n=1}^{N} P(y_n|\boldsymbol{X}, \boldsymbol{y}_{1:n}). \tag{18}$$

We also adopt a CTC objective as an auxiliary task during training and inference to take advantage of the deep acoustic knowledge and the monotonic alignment property of CTC. The CTC loss $\mathcal{L}_{ctc}$ is applied to the deep acoustic outputs generated by the deep acoustic branch. Let's define a set of distinct text tokens as $\mathcal{U}$. The CTC objective is defined as follows for an utterance of T frames,

$$P_{ctc}(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{Z} \prod_{t=1}^{T} P(z_t|z_{t-1}, \boldsymbol{Y}) P(z_t|\boldsymbol{X}), \tag{19}$$

where $Z = \{z_t \in \mathcal{U} \cup \{< \mathrm{b} >\} \mid t = 1, \ldots, T\}$ is a frame-wise text token sequence with blank symbol <b> added, $P(z_t|z_{t-1}, \boldsymbol{Y})$ is the CTC state transition probability, and $P(z_t|\boldsymbol{X})$ is the frame-wise posterior distribution which is calculated using the final deep acoustic representation $\boldsymbol{H}^{(K)}$ as follows,

$$P(z_t|\boldsymbol{X}) = \mathrm{Softmax}(\mathrm{Linear}(\mathrm{LayerNorm}(\boldsymbol{H}^{(K)}))). \tag{20}$$

Furthermore, we introduce a simple language modeling task that predicts the next token based on previous token history to facilitate our model to learn strong text representations from both the paired text and unpaired text. The LM loss $\mathcal{L}_{lm}$ is defined as the cross entropy loss between the probability $P_{lm}(\boldsymbol{Y})$ predicted by the inner-LM and the true label distribution, normalized by the total number of text tokens N. The joint probability of the text token sequence Y is factorized as follows,

$$P_{lm}(\boldsymbol{Y}) = \prod_{n=1}^{N} P(y_n|\boldsymbol{y}_{1:n}). \tag{21}$$

To address the data imbalance problem between the paired and unpaired text, we construct two loss functions and optimize them at different times to exploit a large amount of unpaired data to its full potential. Given the paired text from the speech-text corpus, we linearly combine the above loss functions into a speech loss function $\mathcal{L}_{speech}$ formulated as follows,

$$\mathcal{L}_{speech} = \alpha\mathcal{L}_{ctc} + (1-\alpha)\mathcal{L}_{att} + \beta\mathcal{L}_{lm}, \tag{22}$$

where α is the CTC training weight and β is the LM training weight that satisfy $0 \leq \alpha, \beta \leq 1$. On the other hand, given unpaired text-only training data, we update the model with a text loss function $\mathcal{L}_{text}$ defined as follows,

$$\mathcal{L}_{text} = \beta\mathcal{L}_{lm}. \tag{23}$$

During inference, we use the conventional CTC/attention joint decoding [60]. Given $P_{att}(\boldsymbol{Y}|\boldsymbol{X})$ and $P_{ctc}(\boldsymbol{Y}|\boldsymbol{X})$ as the sequence probabilities output by the attention and CTC model, the most probable text token sequence $\hat{\boldsymbol{Y}}$ given the input speech is defined as follows,

$$\hat{\boldsymbol{Y}} = \arg\max_{\hat{\boldsymbol{Y}} \in \mathcal{U}} \left\{ \lambda \log P_{att}(\boldsymbol{Y}|\boldsymbol{X}) + (1-\lambda) \log P_{ctc}(\boldsymbol{Y}|\boldsymbol{X}) \right\}, \tag{24}$$

with a tunable parameter λ that satisfies $0 \leq \lambda \leq 1$.

A prevalent challenge encountered in pre-training models is the occurrence of catastrophic forgetting, where a model fails to retain previously acquired knowledge while processing new information. In our case, catastrophic forgetting means forgetting the learned speech-text alignments from paired data when obtaining linguistic knowledge from unpaired text data. Another serious yet common difficulty of using external data to update a partial ASR model is the existence of model capacity gap between two parts of a model due to the different training processes and imbalanced training data size, which often leads to degradation in model performance. To overcome catastrophic forgetting and reduce the model capacity gap between our model's inner-LM and other branches, we propose a training algorithm that alternately loads samples from paired and unpaired data and accumulates gradients. The alternate training algorithm is summarized in Algorithm 1 and is described next.

In each iteration, we first load the unpaired text batches and compute the gradients w.r.t. the inner-LM parameters θ_{dec_t} for τ times. τ is an introduced hyperparameter, called text ratio, that controls the number of unpaired text batches used together with a batch of paired speech-text data for parameter

Algorithm 1 The Alternate Training Algorithm

```
Require: paired speech-text dataset $\mathcal{D}_P$, unpaired dataset $\mathcal{D}_U$
Ensure: model parameters $\Theta = \{\theta_{enc}, \theta_{dec\_s}, \theta_{dec\_d}, \theta_{dec\_t}\}$
  Initialize model parameters $\Theta$ randomly;
  repeat
      for $i = 1, \ldots, \tau$ do
          Load a batch of data from $\mathcal{D}_U$ randomly;
          Compute the text loss $\mathcal{L}_{text}$ by Eq.(23);
          Back propagate the text loss to obtain gradients for $\theta_{dec\_t}$;
      end for
      Load a batch of data from $\mathcal{D}_P$ randomly;
      Compute the speech loss $\mathcal{L}_{speech}$ by Eq.(22);
      Back propagate the speech loss to obtain gradients for $\{\theta_{enc}, \theta_{dec\_s}, \theta_{dec\_d}\}$;
      Update model parameters $\Theta$ with all the accumulated gradients;
  until maximum iteration
```

update of the whole model. The use of τ regulates the mixing of two datasets in one simulated big batch. Then, we load a batch of paired speech-text data and compute the gradients w.r.t. the parameters of the encoder θ_{enc}, the deep acoustic branch θ_{dec_s}, and the speech decoding branch θ_{dec_d} for only once. The essential operation of the alternate training algorithm is to accumulate the gradients from several text batches and one speech batch, then update the model parameters with all the accumulated gradients. We call this mechanism *text gradient accumulation*, which we believe helps the speech-and-text Transformer stabilize during training by updating the linguistic-related parameters of our model together with the speech-related parameters.

4 Relations to Prior Work

There have been studies on various neural architectures, learning objectives, and learning procedures with a common goal of leveraging abundant unpaired text data. In what follows, we try to link our work to the related prior work for a better understanding of our proposal.

4.1 Knowledge Distillation

Knowledge distillation describes a group of methods that transfer knowledge from a large and high-performing teacher model to a student model typically trained on a small training data set. Bai *et al.* transfer knowledge from an external RNN language model to an E2E ASR model by guiding the ASR's training with the soft label provided by the teacher LM [3, 5]. Similarly, Futami *et al.* distill knowledge from a bidirectional language model by minimizing the divergence between the soft label and the ASR's prediction [17]. Apart from learning from the teacher model's final prediction, some works focus on supervising the student model with the teacher model's hidden representations as the source of knowledge [4, 11, 35, 63].

Knowledge distillation techniques do not explicitly incorporate the text-only data into the training of the E2E ASR model, but rather rely on transferring knowledge from an external LM in a multi-step training process. Thus, the effectiveness and efficiency of the knowledge distillation become an additional topic of concern. This work is a departure from the knowledge distillation techniques, we study how to allow the ASR model to learn directly on the unpaired text data in the target domain.

4.2 Multi-Task Learning

Multi-task learning is a machine learning paradigm that aims to simultaneously learn multiple related tasks and exploit knowledge and commonalities across tasks.

Sainath *et al.* [49, 58] use a learnable context vector to distinguish between the paired and unpaired examples with a two-stage training scheme. The whole model is trained on the paired text in the first stage, then alternately trained on the paired and unpaired data in the second stage. One critical issue of this method is that the artificial context vector does not contain any speech information as the training is conducted on pure text, which might harm the learning of speech-text alignment in the cross-attention module of the decoder. Tang *et al.* use an extra text encoder and a shared encoder to train the encoder-decoder ASR system with an additional denoising task [53]. The share encoder aims to map inputs of different modalities into the same representation space. Similarly, Yusuf *et al.* use a bank of modality encoder and a shared encoder to co-train the ASR model with a masked language modeling task and two machine translation tasks [65].

Another line of multi-task learning involves jointly training an ASR model with an inverse model to leverage unpaired data and enhance recognition accuracy. A commonly used approach is using a text-to-speech (TTS) system to synthesize speech from unpaired text data, and then training the TTS and ASR systems jointly with both authentic and synthesized speech-text pairs [7, 8, 36, 37, 48, 55]. Rather than creating pseudo speech data, Hayashi *et al.* [26] propose a text-to-encoder (TTE) model to generate synthetic encoder hidden states to use as the speech counterpart of paired data. The use of text-based synthesized data as speech representation such as words or phonemes has also been explored in [46, 61]. To leverage the intermediate representations, Karita *et al.* [32, 33] introduce a speech autoencoder, a text autoencoder, and an inter-domain loss to the system to generate common encoder features for the ASR and TTS models. However, these multi-task learning techniques exhibit certain limitations. Firstly, they rely on external models and require multiple training steps, which increases the training time and complexity of end-to-end models. Secondly, the synthesized data exhibits much less variation than real data, which often leads to failure in generalization in tasks with real data [47, 57].

Unlike the aforementioned methods, our approach does not require any additional artificial context vector, modality conversion component, or inverse model. Instead, we incorporate an inner-LM branch into the ASR model and introduce an on-demand dual-modality attention mechanism to enable the model to be directly trained on unpaired text. Besides, we propose an alternate training algorithm to ensure the proposed model preserves speech-text alignment information while acquiring linguistic knowledge from unpaired textual data.

4.3 Fusion Methods

Fusion methods refer to a group of techniques that utilize unpaired text by integrating a separately pre-trained LM into the E2E ASR model. Shallow fusion [25] is the simplest yet most popular one in end-to-end speech recognition. It linearly combines the output scores of an E2E ASR model and an LM during the ASR's decoding stage. Unlike shallow fusion, deep fusion [23] and cold fusion [52] were studied to fuse the final hidden states between a neural LM and an encoder-decoder model with a parametric gating mechanism during the ASR's training stage. Component fusion [51] and memory attentive fusion [29] build on the idea of cold fusion to integrate the last hidden states of an external LM into the decoder layers of the encoder-decoder ASR models. One common issue of these fusion methods is that employing an additional LM during decoding increases the ASR model's decoding time and complexity. Furthermore, component fusion and memory attentive fusion repeatedly use the same LM's final hidden states in every decoder layer, which presents a mismatch with the decoder hidden states at different levels of abstraction.

Departing from the above, we propose a novel Transformer-based encoder-decoder architecture for speech recognition that contains an inner-LM branch learned from scratch with unpaired text to guide the ASR's training with minimum computation overheads during inference. Furthermore, the inner-LM branch has the same modular structure as the speech decoding branch, which allows for easy linguistic knowledge transfer between the two branches through parameter sharing, at the same time, fully utilizes the hidden states from different layers of the inner-LM.

4.4 Domain Adaptation

Domain adaptation for ASR models aims to overcome the acoustic or linguistic mismatch between training and test conditions. As the E2E ASR model tends to memorize the training speech well, such mismatch becomes even more acute. On the other hand, paired speech-text data in the target domain is often insufficient and hard to obtain. Therefore, researchers have studied adapting the E2E ASR model to a target domain with relatively cheaper text-only data.

McDermott *et al.* extend shallow fusion and propose a method called density ratio [40] based on Bayes' rule. They train a source-domain LM using the paired text data and a target-domain LM using separate text-only data. During decoding, the source-domain LM score is subtracted from the linear interpolated score of the source-domain ASR model and the target-domain LM. Similarly, Meng *et al.* propose internal LM estimation [41, 42] approaches to calculate the internal LM score of an E2E ASR model by eliminating the contribution of the encoder. During inference, the estimated internal LM score is removed from the combined score of the ASR model and the target-domain LM. Although these fusion-based domain adaptation methods achieved good results, they still face a common problem: decoding with an additional target-domain LM component increases the model's decoding latency. Another line of work treats the internal LM part of the E2E ASR system as a standalone neural LM, and fine-tunes it with target-domain text data [45, 54]. Unfortunately, these techniques can only be applied to Transducer-based E2E ASR architecture as the decoder in Transformer-based encoder-decoder models highly depends on acoustic input for computing cross-attention.

Different from these domain adaptation approaches, we propose a unified Transformer-based encoder-decoder architecture that can be trained directly with target-domain text-only data. As a result, we can enhance the model's target-domain linguistic knowledge and adapt the model to the target domain without using an external LM component during decoding.

5 Experimental Setup

We conduct experiments on two in-domain and one cross-domain speech recognition task to evaluate the neural architecture. With the two in-domain tasks, we aim to verify the speech-and-text Transformer's ability to leverage unpaired text data of small (5 times the size of the text in the paired speech-text data) and large (250 times the size of the text in the paired speech-text data) sizes. With the cross-domain experiments, we intend to validate if our model can gain target-domain linguistic knowledge from unpaired text data to alleviate the domain mismatch between source-domain training and target-domain testing speech data. For ease of comparison with previous work, the corpora we used are publicly available. All the experiments are implemented with the end-to-end speech processing toolkit ESPNet [59].

5.1 Dataset

Chinese In-domain Speech Recognition: We first conduct in-domain speech recognition experiments with a Mandarin speech corpus:
AISHELL-1 [9] and small-scale unpaired text data from AISHELL-2 [15]. AISHELL-1 is a 178-hours read speech corpus sampled at 16 kHz. It covers five topics: finance, science and technology, sports, entertainments, and news. AISHEEL-2 contains 1,000 hours of speech with application topics similar to AISHELL-1. Since these two corpora contain similar speaking styles and topics, we consider they are from the same domain. We use the *training* set of AISHELL-1 as the paired audio-text data and the unique sentences in AISHELL-2 as the unpaired text data. By removing sentences that appeared in the AISHELL-1 from the unpaired text data, we obtain a text-only dataset of 591,291 sentences[1]. The number of sentences in the unpaired text data is approximately 5 times of the number of utterances in the paired speech data. The data structure of AISHELL-1 and AISHELL-2 after the clean-up is summarized in Table 1. The model's hyperparameters are tuned on the AISHELL-1 *dev.* set and the final model is tested on the AISHELL-1 *test* set.

Table 1: Dataset statistics of in-domain Chinese experiments.

		#Hours	#Utterance	#Uniq. Sent.
AISHELL-1	Training	150	120,098	–
	Dev.	18	14,326	–
	Test	10	7,176	–
AISHELL-2	-	-	-	591,291

English In-domain Speech Recognition: We then perform in-domain speech recognition experiments with an English corpus LibriSpeech [43] and larger text-only datasets. LibriSpeech is a 960-hours read English speech corpus derived from audiobooks and sampled at 16 kHz. The speech training data of LibriSpeech is partitioned into three sets: *train-clean-100*, *train-clean-360*, and *train-other-500*. The corpus also contains 4.3 GB of additional text-only data in the same domain for building language models. We use the *train-clean-100* set as the paired speech data, and use the transcriptions of the remaining 860-hours training data and a subset of the LibriSpeech-LM dataset as the

[1] The selected unpaired text data for the Chinese in-domain speech recognition experiments can be downloaded at: https://www.dropbox.com/sh/9xyyftgvr69hrnw/AADAPieK9PVYSCC6hmR34bYua?dl=0

unpaired text data for our experiments.[2] The number of sentences in the two unpaired text data is approximately 9 and 250 times the number of utterances in the paired speech data. We tune our models on the 10-hours merged development set and evaluate the *test-clean* and *test-other* set performance. The data structure of the Librispeech corpus and text database after the clean-up is summarized in Table 2.

Table 2: Dataset statistics of in-domain English experiments.

		#Hours	#Utterance	#Uniq. Sent.
LibriSpeech	train-clean-100	100	28,539	28,539
	train-clean-360	363	–	103,973
	train-other-500	497	–	148,627
	dev-clean	5	2,703	–
	dev-other	5	2,864	–
	test-clean	5	2,620	–
	test-other	5	2,934	–
Subset of LibriSpeech-LM	–	–	–	7,134,750

Chinese Cross-domain Speech Recognition: We conduct cross-domain experiments with the AISHELL-1 corpus and a multi-domain Mandarin speech corpus WenetSpeech [66]. WenetSpeech is a $10,000+$ hours multi-domain speech corpus collected from YouTube and Podcast and re-sampled to 16 kHz. It contains ten domains: audiobook, commentary, documentary, drama, interview, news, reading, talking, variety, and others. We set the read speech from AISHELL-1 as the source domain and the radio speech from WenetSpeech audiobook and news domain as the target domains for our cross-domain speech recognition experiments. Besides the domain mismatch, the inherent difference between the audio characteristics of the source domain's read speech and the target domain's radio speech poses another challenge to our cross-domain speech recognition experiments. Specifically, audio samples from the WenetSpeech-audiobook and news domain are characterized by a wide range of original audio bandwidth, large variation in speech style and speaking rate, and inclusion of background music and noises. In contrast, AISHELL-1's audio samples are all read speech collecting in a controlled environment with proper pronunciation, consistent speaking style and rates, and minimal background

[2] The selected unpaired text data for the English in-domain speech recognition experiments can be downloaded at: https://www.dropbox.com/sh/n8zvdk11kj3y8i3/AAAXyP-MifOEyZh1DCDkheqva?dl=0

noise. All ASR models are trained with the paired speech from the AISHELL-1 *training* set, and tuned and tested on the target domain's *dev.* and *test* set.[3]

We randomly select 14,000 utterances from the target domain with label confidence of 1.0 as the initial development set and select another 7,000 utterances as the *test* set. We use the transcriptions of the remaining utterances from the target domain with label confidence greater than 0.95 as the initial unpaired text data. The remaining data are used as the *training* set. We remove any duplicated sentences from the *training* set and the *dev.* set, and make sure that these three sets do not overlap. The target domain data after the clean-up for the cross-domain experiments is summarized in Table 3.

Table 3: Dataset statistics of cross-domain Chinese experiments.

Domain		#Hours	#Utterance	#Uniq. Sent.
Audiobook	Training	–	–	269,516
	Dev.	10	13,040	–
	Test	5	7,000	–
News	Training	–	–	1,239,205
	Dev.	8	12,255	–
	Test	4	7,000	–

5.2 Implementation Details

Basic Settings. We extract 80-dimensional log Mel-filter bank features (FBANK) plus 3-dimensional pitch features as the acoustic features, normalized with cepstral mean and variance normalization (CMVN) calculated from the training set. For English experiments, we apply byte-pair encoding (BPE) on the Librispeech 960-hours speech transcriptions to generate subword units of vocabulary size 5,000 as the text modeling units. For Chinese experiments, we use 5,210 frequently used Chinese characters extracted from AISHELL-2 training text with $\langle unk \rangle$, $\langle sos \rangle$, and $\langle eos \rangle$ tokens added as the output units.

ASR Models. We follow the LibriSpeech[4] and AISHELL-1[5] small Transformer recipes provided by ESPnet for model configuration. The speech-and-text Transformers contain 12 encoder and 6 decoder blocks, with an output dimension of 256 and an inner-layer dimension of 2,048. To maintain fairness in terms of model size, the baseline Transformer utilized in the experiments

[3] The selected speech (*dev.* and *test* set) and unpaired text data for the Chinese cross-domain experiments can be downloaded at: https://www.dropbox.com/sh/eargormy1u15eu5/AAB8V3cYU2s4SoGUg9BcgSySa?dl=0

[4] https://github.com/espnet/espnet/blob/master/egs/librispeech/asr1/RESULTS.md

[5] https://github.com/espnet/espnet/blob/master/egs/aishell/asr1/RESULTS.md

comprises 17 encoder layers instead of the 6 used in the speech-and-text Transformer. Both the speech-and-text Transformer and the baseline Transformer have a model size of 38M. 4 attention heads are used for the multi-head attention modules and the proposed multi-input multi-head attention modules. A stack of two 3×3 CNN subsampling layers with stride 2 is used as the acoustic embedding. The ASR models are optimized with the Adam algorithm with an initial learning rate of 1.0 for all Chinese experiments and 5.0 for all English experiments. The batch size is 32 for Chinese models, and the number of batch bins is 599,200 for English models. We use Noam [56] as the learning rate scheduler. A warm-up learning rate schedule with a warm-up rate of 25,000 is used for all ASR models. We set gradient clipping to 5 and gradient accumulation to 8 for English models and 5 for Chinese models. The dropout rate is set to 0.1 for all ASR models. For baseline Transformers, we train English models for 100 epochs and train Chinese models for 50 epochs. We increase the number of epochs for speech-and-text Transformers to ensure convergence. The weight α of the CTC branch is set to 0.3 during training and set to 0.5 during decoding for all experiments. Parameters from the best 10 epochs on the validation set are averaged as the final ASR model for inference.

LMs. We train long short-term memory (LSTM) based LMs as the external LMs used for shallow fusion. All LMs are one-layer LSTM models with 512 LSTM units. The LMs are trained with stochastic gradient descent (SGD) algorithm for 20 epochs. The batch size is set to 64, and the learning rate is set to 1.0. We select the best epoch on the validation set as the final LM used for shallow fusion. The perplexities of the trained LMs on the LibriSpeech testing sets are reported in Table 4. To study the influences of domain discrepancy on LM performance, we report the perplexities of the trained LMs on AISHELL-1, WenetSpeech-Audiobook and WenetSpeech-News testing sets in separate blocks of Table 5. The LM trained using transcriptions of AISHELL-1 has a relatively higher perplexity of 272 on WenetSpeech-Audiobook *test* set and a lower perplexity of 99 on WenetSpeech-News *test* set. It is understood that WenetSpeech-News is more similar than WenetSpeech-Audiobook to the AISHELL-1 *training* set.

Table 4: Perplexity of LMs on the LibriSpeech testing sets.

LM	#Sent.	PPL	
		test-clean	test-other
Trans. of Librispeech	281,231	82	81
Subset of LibriSpeech-LM	7,134,750	78	77

Evaluations. We evaluate the performance of English ASR models using Word Error Rate (WER) and evaluate the performance of Chinese ASR models

Table 5: Perplexity of LMs on dev and test set of AISHELL-1, WenetSpeech-Audiobook, and WenetSpeech-News.

	LM	#Sent.	PPL	
			Dev.	Test
AISHELL-1	Trans. of AISHELL-1	120,098	61	58
	Trans. of AISHELL-2	591,291	57	53
WenetSpeech-Audiobook	Trans. of AISHELL-1	120,098	276	272
	Trans. of Audiobook	269,516	60	59
WenetSpeech-News	Trans. of AISHELL-1	120,098	100	99
	Trans. of News	1,239,205	44	43

using Character Error Rate (CER). Perplexity (PPL) is used as the evaluation metric for LMs. We also evaluate the language modeling ability of ASR models with speech-and-text decoder by calculating the perplexity of the inner-LM part of it on testing data.

6 Results

6.1 In-domain Evaluation on AISHELL-1

The in-domain evaluations are conducted on the AISHELL-1 (Chinese) *dev.* and *test* set and reported in Table 6.

Table 6: AISHELL-1: CERs on the *dev.* and *test* set. Upper section: fusion methods discussed in Section 4. Middle section: recent semi-supervised techniques. Lower section: speech-and-text Transformer framework.

Method	#Param.	Unpaired Speech	Unpaired Text	CERs (%)	
				Dev.	Test
LAS [51]	–	–	–	–	10.6
Cold Fusion [51]	–	–	AISHELL-2	–	10.1
Component Fusion [51]	–	–	AISHELL-2	–	9.0
Transformer + SP + SF [31]	30M	–	–	6.0	6.7
Pre-training [16]	–	AISHELL-2	AISHELL-2	–	7.3
Teacher-Student Learning + SA [4]	68M	-	Subset of CLMAD	–	6.4
Baseline Transformer	38M	–	–	7.1	7.9
Baseline Transformer + SF	38M	–	AISHELL-2	6.2	6.7
Speech-and-Text Transformer	38M	–	AISHELL-2	**6.0**	**6.9**
Speech-and-Text Transformer + SF	38M	–	AISHELL-2	5.6	6.1
Speech-and-Text Transformer + SP	38M	–	AISHELL-2	5.7	6.4
Speech-and-Text Transformer + SA	38M	–	AISHELL-2	**5.2**	**5.8**

Note: SP stands for speed perturbation; SF stands for shallow fusion; SA stands for SpecAugment.

We first compare with cold fusion and component fusion discussed in Section IV under the RNN-based encoder-decoder framework. With the transcriptions of AISHELL-2 as the unpaired text in ASR training, these methods obtain 5.0% and 15.1% relative CER reduction over the vanilla LAS model, as summarized in the upper section of Table 6.

We then compare with the recent semi-supervised techniques that utilize unpaired speech and text data for training Transformer-based ASR models. Among all these previous studies, the teacher-student learning [4] could be the closest to our experiment setups, which uses a subset of CLMAD [2] dataset as the external text to train an external LM for knowledge distillation. They use about 30 times the number of text sentences than speech utterances and employ the SpecAugment [44] strategy in ASR training, to reach a CER of 6.4% on the AISHELL-1 *test* set.

Finally, in the lower section of Table 6, we summarize the experiment results of the baseline and our various speech-and-text models. With the number of sentences in the unpaired text being approximately 5 times that of speech transcripts in the paired speech-text data, the proposed speech-and-text Transformer obtains 12.7% relative CER reduction on the AISHELL-1 *test* set over the baseline Transformer. The CER is further reduced to 6.4% and 5.8% with speed perturbation [34] and SpecAugment applied to the speech-and-text Transformer during training. These results imply that the speech-and-text Transformer is compatible with other data augmentation techniques such as SpecAugment and speed perturbation. When decoding with an external RNNLM trained on the AISHELL-2 transcriptions, the speech-and-text Transformer attains a relative CER reduction of 9.0% compared to the baseline Transformer with shallow fusion applied. This implies that the speech-and-text Transformer can also be used with an external LM during the decoding stage to boost its performance. With a small number of model parameters and fewer unpaired text data, our method outperformed the baseline model and previous methods for the AISHELL-1 in-domain speech recognition task without introducing additional decoding complexity.

6.2 In-Domain Evaluation on LibriSpeech-100 hours

We then conduct in-domain experiments using the LibriSpeech (English) *train-clean-100* set as the paired speech data and evaluate on the *test-clean* and *test-other* sets. The results are reported in Table 7.

The upper section of Table 7 shows the results of previous semi-supervised methods trained with 100 hours of paired speech-text data and varied amounts of unpaired text data. Among these studies, the best result comes from the ASR system trained using the multi-task training method [58] with the unpaired text data from the whole LibriSpeech LM dataset, which achieves WERs of 10.1% and 30.4% on the LibriSpeech *test-clean* and *test-other*.

Table 7: Librispeech-100h: WERs on the *test-clean* and *test-other* set. Upper section: recent techniques exploiting text data. Lower section: speech-and-text Transformer framework.

Method	Unpaired Text	WERs (%)	
		test-clean	test-other
Adversarial Training [38]	860 hrs	18.7	–
Pre-Training [18]	860 hrs	11.2	30.5
Multitask Training [58]	LibriSpeech-LM	10.1	30.4
Baseline Transformer	-	12.7	31.8
Baseline Transformer + SF	860 hrs	10.5	27.8
Speech-and-Text Transformer	860 hrs	11.0	29.0
Speech-and-Text Transformer + SF	860 hrs	9.9	26.3
Baseline Transformer + SF	Subset of LibriSpeech-LM	10.1	27.2
Speech-and-Text Transformer	Subset of LibriSpeech-LM	**10.2**	**28.4**
Speech-and-Text Transformer + SF	Subset of LibriSpeech-LM	**9.0**	**25.1**

Note: SF stands for shallow fusion.

The lower section of Table 7 summarizes the experiment results of our baseline model and our proposed method for exploiting unpaired text-only data for E2E ASR. We find that our method achieves the best recognition performance compared to the previous techniques [18, 38] when using a small amount of unpaired text, i.e., the transcription of the 860-hours LibriSpeech training data. Specifically, our method yields WERs of 11.0% on *test-clean* and 29.0% on *test-other*. Furthermore, when applying shallow fusion during decoding, the WERs of our model on *test-clean* and *test-other* are further reduced to 9.9% and 26.3%, respectively. When using a large amount of unpaired text, i.e., the subset of the LibriSpeech-LM dataset, our method achieves competitive results compared to the multi-task learning model [58], which is trained with 5 times more text data than ours. Moreover, when applying shallow fusion for post-processing, our model attains relative WER reductions of 10.9% and 7.7% on *test-clean* and *test-other*, respectively, compared to the baseline Transformer.

We further study the impact of varying the amount of unpaired text data used in ASR training to better understand how the text-only data affects the performance of the speech-and-text Transformer. We report the perplexity of the inner-LM of the speech-and-text Transformer as well as the WER of the model. The results are summarized in Table 8. We fix the LM weight to 0.5 and the text ratio to 5 for all experiments in this table to eliminate other interference factors. As the amount of text data increases from zero to 250 times, the perplexity of our model drops consistently from 177 and 179 to 59 and 59 on *test-clean* and *test-other*. Similarly, the WERs of our model also decrease from 13.2% and 31.0% to 10.2% and 28.2% on *test-clean* and *test-other*. These results suggest that the inner-LM branch of the speech-and-text Transformer acts as a stand-alone LM and benefits from a large amount

Table 8: Librispeech-100h: Contributions of Unpaired Text data.

Unpaired Text	#Sentences	test PPL		test WERs (%)	
		clean	other	clean	other
–	0	177	179	13.2	31.0
10 times	285,390	110	110	12.6	30.3
50 times	1,426,950	82	79	11.1	29.0
100 times	2,853,900	77	75	10.6	28.8
200 times	5,707,800	67	66	10.3	28.5
250 times	7,134,750	59	59	**10.2**	**28.4**

Note: “times” means that the number of sentences in the unpaired text is N times large than that of the paired speech-text training data.

of text data to improve its language modeling ability. Besides, the inner-LM highly correlates to the ASR model, which implies that a stronger inner-LM means a speech-and-text Transformer with better linguistic knowledge. These findings demonstrate the ability of the speech-and-text Transformer to learn from external text data and utilize the learned linguistic knowledge for speech recognition. Moreover, the speech-and-text Transformer is able to maintain low perplexity on the LM task while improving accuracy on the ASR task, indicating our method effectively mitigates catastrophic forgetting.

However, we also observed that the performance improvement of the model with additional unpaired text from 100 to 250 times is relatively small. We attribute this to the proposed model’s ability to leverage linguistic knowledge being constrained by the quality of the inner-LM, as evaluated through perplexity. Specifically, as we increase the amount of unpaired text from 100 to 250 times, the perplexity reduction on the test sets becomes less significant, resulting in a smaller improvement in WERs. In the future, we plan to investigate the training objective and architecture of the inner-LM to further enhance its language modeling ability, which would also help the proposed model to leverage unpaired text better and improve performance.

6.3 Cross-domain Evaluation on WenetSpeech

We evaluate the speech-and-text Transformer trained on out-of-domain AISHELL-1 speech data and in-domain text data on the WenetSpeech-Audiobook and News. The results are shown in Table 9.

For the audiobook domain, the speech-and-text Transformer is trained with unpaired text data approximately 2 times the amount of speech data in terms of the number of sentences. It obtains CERs of 43.2% and 43.7% on the WenetSpeech-Audiobook *dev.* and *test* set, amounting to 13.6% and 13.6% relative CER reduction over the baseline Transformer. When applying shallow

Table 9: WenetSpeech: CERs on the *dev.* and *test* set of audiobook and news domain.

Domain	Method	Unpaired Text	CERs (%) Dev.	CERs (%) Test
Audiobook	*Transf. + SP	No	48.7	49.4
	Baseline Transf.	No	50.0	50.6
	Baseline Transf. + SF	Yes	42.0	42.5
	Speech-and-Text Transf.	Yes	**43.2**	**43.7**
	Speech-and-Text Transf. + SF	Yes	**36.4**	**36.5**
News	*Transf. + SP	No	47.0	49.2
	Baseline Transf.	No	49.3	52.1
	Baseline Transf. + SF	Yes	42.9	45.4
	Speech-and-Text Transf.	Yes	**42.2**	**42.9**
	Speech-and-Text Transf. + SF	Yes	**37.8**	**38.3**

Note: Transf. is an abbreviation for Transformer. * are evaluated on the Transformer model downloaded from the ESPnet official repository. https://drive.google.com/open?id=1BIQBpLRRy3XSMT5IRxnLcgLMirGzu8dg.

fusion to the speech-and-text decoder with an external RNNLM trained on the same amount of unpaired text, the CERs are further reduced to 36.4% and 36.5%.

For the news domain, the speech-and-text Transformer is trained with unpaired text data approximately 10 times the amount of speech data in terms of the number of sentences. It outperforms the baseline model with shallow fusion, achieving CERs of 42.2% and 42.9% on the WenetSpeech-Audiobook *dev.* and *test* set, which corresponds to 14.4% and 17.6% relative CER reduction over the baseline Transformer. With additional RNNLM used for shallow fusion, the CERs of our model are further reduced to 37.8% and 38.3%.

By summarizing the results from these two domain adaptation experiments, we can see that the proposed speech-and-text Transformer consistently improves speech recognition performance in scenarios where the unpaired text data originates from a distinct domain than the paired speech training data. This indicates that our proposed model successfully learns the target domain's linguistic knowledge from the unpaired text and effectively mitigates the domain mismatch between the training and testing speech data, even in the presence of the audio characteristic difference between the source and target domain.

7 Analysis

In this section, we conduct ablation studies on the speech-and-text Transformer and investigate the impacts of its hyper-parameters text ratio and LM weight on the model's performance. We also examine some decoded examples of our method and baseline method to investigate the contribution of external text to ASR performance.

7.1 Ablation Study

We conduct ablation studies on the speech-and-text Transformer to examine the contributions of each of its components to the overall system under the in-domain Chinese speech recognition experiment setting. We use the AISHELL-1 training set as the paired speech data and transcriptions of the AISHELL-2 training set as the unpaired text data. To isolate the impact of other tunable parameters on model performance, we set the text ratio τ to 20 and LM weight β to 0.7 for all experiments in this sub-section. We exclude one component each time and compare the perplexity and CER changes on the AISHELL-1 *dev.* and *test* set. The experiment results are reported in Table 10.

Table 10: AISHELL-1: Ablation study on speech-and-text Transformer by excluding one component each time.

	#Param.	PPL		CERs (%)	
		Dev.	Test	Dev.	Test
Speech-and-Text Transformer	38M	39	34	6.0	6.9
w/o Parameter Sharing	48M	40	35	6.4	7.3
w/o Inner-LM Branch	38M	–	–	6.5	7.4
w/o Deep Acoustic Branch	38M	39	35	6.7	7.6
w/o Text Gradient Accumulation	38M	43	38	10.7	12.0

First, we investigate the impact of parameter sharing between the inner-LM branch and the speech decoding branch of the speech-and-text Transformer. Erasing this parameter sharing leads to a 10M increase in model parameters, a slight degradation in perplexity, and a considerable drop in speech recognition performance. These results confirm that parameter sharing between the inner-LM branch and the speech decoding branch reduces the model parameter and helps the ASR model utilize the linguistic knowledge learned from the external text.

Second, we analyze the effect of removing the inner-LM branch on the speech recognition ability of our proposed model. We find that removing the inner-LM branch results in a significant increase in CERs, which underscores the critical role that the inner-LM branch plays in enabling the model to learn from unpaired text data and improving its linguistic knowledge. Moreover, this finding is consistent with previous results in Table 8, which highlights the importance of incorporating external text data in the proposed speech-and-text

Transformer. By leveraging a large amount of unpaired text, the proposed models can enhance their language modeling capabilities and achieve better performance in recognizing speech.

Third, we examine the role of the deep acoustic branch in the speech-and-text decoder by removing it and feeding the same encoder representation to every decoder block, similar to the vanilla Transformer decoder. To ensure a fair comparison of model size, we increase the number of encoder layers in the speech-and-text Transformer from 12 to 18, resulting in a model size of 38M after removing the deep acoustic branch. The results show that the model achieves similar perplexity on test sets compared to the full speech-and-text Transformer, indicating that the removed deep acoustic branch does not significantly impact the model's language modeling capability. However, we observe a noticeable drop in speech recognition performance when the deep acoustic branch is removed, suggesting its effectiveness in extracting high-level acoustic representations and reducing the mismatches between speech and text representations in different decoder layers.

Lastly, we assess the effect of *text gradient accumulation* on our model's performance. Specifically, we update the parameters of the inner-LM branch and the other part of our model separately for each text or speech batch. We find that stopping the accumulation of gradients from text batches does not have a substantial impact on the model's language modeling ability, as evidenced by the slight increase in perplexity. However, we observe a severe deterioration in ASR performance, indicating that the model struggles to retain its proficiency in the ASR task while simultaneously learning the LM task. These results underscore the crucial role played by the *text gradient accumulation* mechanism in stabilizing our model during training and mitigating catastrophic forgetting.

7.2 Text Ratio and LM Weight

We conduct experiments with the in-domain Chinese speech recognition experiment settings to study the effect of text ratio τ and the LM training weight β on the performance of the speech-and-text Transformer.

We set the LM weight β to 0.7 for the text ratio experiments where we vary the text ratio from 1 to 50 and report the results on the AISHELL-1 *dev.* set. The experiment results are reported in Figure 5. As the text ratio increases, the model's perplexity improves and saturates at 1:20, where we observe the lowest CER. We notice that the CER increases for a higher text ratio. This suggests that a proper text ratio and the training strategy play a role.

The LM weight β is the training weight that regulates the contribution of unpaired text data to the inner-LM. We further conduct experiments with the in-domain Chinese speech recognition experiment settings to study its impact by setting the text ratio to 20 for all experiments and varying the LM weight

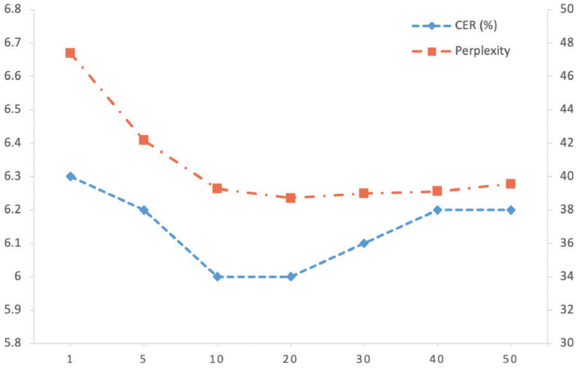

Figure 5: CERs and perplexity on the AISHELL-1 *dev.* set under various text ratios. The x-axis denotes the text ratio. The primary y-axis (left) is the axis for CER (%), and the secondary y-axis (right) is the axis for perplexity.

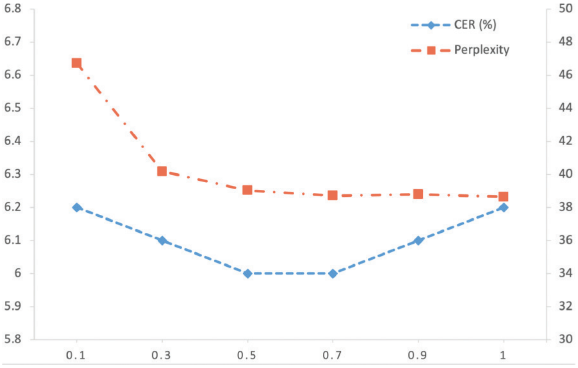

Figure 6: CERs and perplexity on the AISHELL-1 *dev.* set under various LM weights. The x-axis denotes the LM weight. The primary y-axis (left) is the axis for CER (%), and the secondary y-axis (right) is the axis for perplexity.

from 0.1 to 1.0. The experiment results are summarized in Figure 6. The CER reaches the lowest when $\beta = 0.7$.

8 Conclusion

In this work, we introduce the speech-and-text Transformer, an end-to-end speech recognition architecture that can directly learn from unpaired text data to gain linguistic competence with minimum decoding computational overheads. In-domain speech recognition experiments and perplexity studies on Chinese (AISHELL-1) and English (LibriSpeech) corpora demonstrated the effectiveness of our proposed architecture and training scheme in leveraging small and large text corpus to enhance the linguistic capability of the E2E ASR model. Besides, the cross-domain experiments on the WenetSpeech corpus confirmed that our model has the capacity to acquire the linguistic knowledge of the target domain from unpaired text, thereby minimizing the domain mismatch between the speech data used for training and testing. The ablation studies on the speech-and-text Transformer reveal that all branches and training strategies are crucial to the performance of our model. We found that the text ratio and LM weight selection significantly influence the model performance. Therefore, we would like to study ways for the model to auto-adjust the hyperparameters to achieve good performance in the future. We will also consider applying our model to under-resourced languages and multi-lingual scenarios.

Acknowledgement

The work is funded by National Natural Science Foundation of China (Grant No. 62271432), Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen (Grant No. B10120210117-KP02), and the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project No. A18A2b0046).

Biographies

Qinyi Wang received the B.Eng. (First-Class) degree in Electrical Engineering from York University, ON, Canada, in 2018. She is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering of National University of Singapore (NUS), Singapore, under the supervision of Professor Haizhou Li. Since April 2019, she has been working as a Research Engineer at the Human Language Technology (HLT) Lab at the Department of Electrical and Computer Engineering of NUS, Singapore. Her research interests include automatic speech recognition and natural language processing.

Xinyuan Zhou received his B.Eng. degree from the Department of Electronic Engineering at Chengdu University of Information Technology, Chengdu, China, in 2018, and the M.Sc. degree in Communication and Information Systems from Shanghai Normal University, Shanghai, China, in 2021. He was an exchange student with the Human Language Technology (HLT) Lab at National University of Singapore (NUS), Singapore, from 2019 to 2021. He is currently a Researcher with the Research and Development Group at Iflytek Corporation, Hefei, China. His research interests include speech processing and natural language processing, particularly speech recognition and translation. machine learning.

Haizhou Li received the B.Sc, M.Sc, and Ph.D degrees in electrical and electronic engineering from South China University of Technology, Guangzhou, China in 1984, 1987, and 1990 respectively. He is now a Presidential Chair Professor and Executive Dean at the School of Data Science, The Chinese University of Hong Kong (Shenzhen). Dr. Li is also with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore. Dr. Li's research interests include automatic speech recognition, natural language processing and information retrieval. Dr. Li has worked on speech and language technology in academia and industry since 1988. He has taught in the University of Hong Kong (1988-1990) and South China University of Technology (1990-1994). He was a Visiting Professor at CRIN in France (1994-1995), Research Manager at the Apple-ISS Research Centre (1996-1998), Research Director in Lernout & Hauspie Asia Pacific (1999-2001), Vice President in InfoTalk Corp. Ltd. (2001-2003), and the Principal Scientist and Department Head of Human Language Technology in the Institute for Infocomm Research, Singapore (2003-2016). Dr. Li served as the Editor-in-Chief of IEEE/ACM Transactions on Audio, Speech, and Language Processing (2015-2018), a Member of the Editorial Board of Computer Speech and Language (2012-2018). He was an elected Member of IEEE Speech and Language Processing Technical Committee (2013-2015), the President of the International Speech Communication Association (2015-2017), the President of Asia Pacific Signal and Information Processing Association (2015-2016), and the President of Asian Federation of Natural Language Processing (2017-2018). He was the General Chair of ACL 2012, INTERSPEECH 2014 and ASRU 2019. Dr. Li is a Fellow of the IEEE and the ISCA. He was the recipient of the National Infocomm Award 2002 and the President's Technology Award 2013 in Singapore. He was named one of the two Nokia Visiting Professors in 2009 by the Nokia Foundation, and U Bremen Excellence Chair Professor in 2019.

References

[1] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-End Attention-based Large Vocabulary Speech Recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, 4945–9.

[2] Y. Bai, J. Tao, J. Yi, Z. Wen, and C. Fan, "CLMAD: A Chinese Language Model Adaptation Dataset," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, IEEE, 2018, 275–9.

[3] Y. Bai, J. Yi, J. Tao, Z. Tian, and Z. Wen, "Learn Spelling from Teachers: Transferring Knowledge from Language Models to Sequence-to-Sequence Speech Recognition," in *Proc. Interspeech 2019*, 2019, 3795–9, DOI: 10.21437/Interspeech.2019-1554.

[4] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, "Fast End-to-end Speech Recognition via Non-autoregressive Models and Cross-modal Knowledge Transferring from Bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021, 1897–911.

[5] Y. Bai, J. Yi, J. Tao, Z. Wen, Z. Tian, and S. Zhang, "Integrating Knowledge Into End-to-End Speech Recognition From External Text-Only Data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021, 1340–51.

[6] A. Bapna, Y.-a. Chung, N. Wu, A. Gulati, Y. Jia, J. H. Clark, M. Johnson, J. Riesa, A. Conneau, and Y. Zhang, "SLAM: A Unified Encoder for Speech and Language Modeling via Speechtext Joint Pre-training," *arXiv preprint arXiv:2110.10329*, 2021.

[7] M. K. Baskar, L. Burget, S. Watanabe, and R. F. Astudillo, "EAT: Enhanced ASR-TTS for Self-supervised Speech Recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, 6753–7.

[8] M. K. Baskar, S. Watanabe, R. Astudillo, T. Hori, L. Burget, and J. Černocký, "Semi-Supervised Sequence-to-Sequence ASR Using Unpaired Speech and Text," in *Proc. Interspeech 2019*, 2019, 3790–4, DOI: 10.21437/Interspeech.2019-3167, http://dx.doi.org/10.21437/Interspeech.2019-3167.

[9] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An Open-source Mandarin Speech Corpus and a Speech Recognition Baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, IEEE, 2017, 1–5.

[10] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, 4960–4.

[11] K. Choi and H.-M. Park, "Distilling a Pretrained Language Model to a Multilingual ASR Model," *arXiv preprint arXiv:2206.12638*, 2022.

[12] J. Chorowski and N. Jaitly, "Towards Better Decoding and Language Model Integration in Sequence to Sequence Models," in *Proc. Interspeech 2017*, 2017, 523–7, DOI: 10.21437/Interspeech.2017-343.

[13] K. Deng, S. Cao, Y. Zhang, and L. Ma, "Improving Hybrid CTC/Attention End-to-End Speech Recognition with Pretrained Acoustic and Language Models," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, 76–82.

[14] L. Dong, S. Xu, and B. Xu, “Speech-Transformer: A No-recurrence Sequence-to-sequence Model for Speech Recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, 5884–8.

[15] J. Du, X. Na, X. Liu, and H. Bu, “Aishell-2: Transforming Mandarin ASR Research into Industrial Scale,” *arXiv preprint arXiv:1808.10583*, 2018.

[16] Z. Fan, S. Zhou, and B. Xu, “Unsupervised Pre-training for Sequence to Sequence Speech Recognition,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017, 383–91, DOI: 10.18653/v1/D17-1039.

[17] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, “Distilling the Knowledge of BERT for Sequence-to-Sequence ASR,” in *Proc. Interspeech 2020*, 2020, 3635–9, DOI: 10.21437/Interspeech.2020-1179.

[18] C. Gao, G. Cheng, R. Yang, H. Zhu, P. Zhang, and Y. Yan, “Pre-Training Transformer Decoder for End-to-End ASR Model with Unpaired Text Data,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, 6543–7.

[19] A. Graves, “Sequence Transduction with Recurrent Neural Networks,” *arXiv preprint arXiv:1211.3711*, 2012.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence data with recurrent neural networks,” in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, 369–76.

[21] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, Ieee, 2013, 6645–9.

[22] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proc. Interspeech 2020*, 2020, 5036–40, DOI: 10.21437/Interspeech.2020-3015.

[23] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, “On Using Monolingual Corpora in Neural Machine Translation,” *arXiv e-prints*, 2015, arXiv–1503.

[24] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, and A. Coates, “Deep Speech: Scaling up End-to-end Speech Recognition,” *arXiv preprint arXiv:1412.5567*, 2014.

[25] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, and A. Coates, “Deep Speech: Scaling up End-to-end Speech Recognition,” *arXiv preprint arXiv:1412.5567*, 2014.

[26] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style Data Augmentation for End-to-end ASR," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, 426–33.

[27] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, *et al.*, "Streaming End-to-end Speech Recognition for Mobile Devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 6381–5.

[28] X. Huang and L. Deng, "An Overview of Modern Speech Recognition," *Handbook of Natural Language Processing*, 2, 2010, 339–66.

[29] M. Ihori, R. Masumura, N. Makishima, T. Tanaka, A. Takashima, and S. Orihashi, "Memory Attentive Fusion: External Language Model Integration for Transformer-based Sequence-to-Sequence Model," in *Proceedings of the 13th International Conference on Natural Language Generation*, 2020, 1–6.

[30] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proceedings of the IEEE*, 64(4), 1976, 532–56.

[31] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A Comparative Study on Transformer vs RNN in Speech Applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, 449–56, DOI: 10.1109/ASRU46091.2019.9003750.

[32] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised End-to-end Speech Recognition Using Text-to-speech and Autoencoders," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 6166–70.

[33] S. Karita, S. Watanabe, T. Iwata, A. Ogawa, and M. Delcroix, "Semi-Supervised End-to-End Speech Recognition," in *Proc. Interspeech 2018*, 2018, 2–6, DOI: 10.21437/Interspeech.2018-1746.

[34] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for Speech Recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[35] Y. Kubo, S. Karita, and M. Bacchiani, "Knowledge Transfer from Large-scale Pretrained Language Models to End-to-end Speech Recognizers," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 8512–6.

[36] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not Need More Data: Improving End-to-end Speech Recognition by Text-to-speech Data Augmentation," in *2020*

13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE, 2020, 439–44.

[37] J. Li, R. Gadde, B. Ginsburg, and V. Lavrukhin, "Training Neural Speech Recognition Systems with Synthetic Speech Augmentation," *arXiv preprint arXiv:1811.00707*, 2018.

[38] A. H. Liu, H.-y. Lee, and L.-s. Lee, "Adversarial Training of End-to-end Speech Recognition using a Criticizing Language Model," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 6176–80.

[39] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR Systems for LibriSpeech: Hybrid vs Attention," in *Proc. Interspeech 2019*, 2019, 231–5, DOI: 10.21437/Interspeech.2019-1780.

[40] E. McDermott, H. Sak, and E. Variani, "A Density Ratio Approach to Language Model Fusion in End-to-end Automatic Speech Recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2019, 434–41.

[41] Z. Meng, N. Kanda, S. Parthasarathy, Y. Gaur, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, "Internal Language Model Training for Domain-adaptive End-to-end Speech Recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, 7338–42.

[42] Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong, "Internal Language Model Estimation for Domain-adaptive End-to-end Speech Recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, 243–50.

[43] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An Asr Corpus based on Public Domain Audio Books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2015, 5206–10.

[44] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, 2613–7, DOI: 10.21437/Interspeech.2019-2680.

[45] J. Pylkkönen, A. Ukkonen, J. Kilpikoski, S. Tamminen, and H. Heikinheimo, "Fast Text-only Domain Adaptation of RNN-transducer Prediction Network," *arXiv preprint arXiv:2104.11127*, 2021.

[46] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-Modal Data Augmentation for End-to-end ASR," *Proc. Interspeech 2018*, 2018, 2394–8.

[47] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, "Speech Recognition with Augmented Synthesized Speech," in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, IEEE, 2019, 996–1002.

[48] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating Synthetic Audio Data for Attention-based Speech Recognition Systems," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 7069–73.

[49] T. N. Sainath, R. Pang, R. J. Weiss, Y. He, C. Chiu, and T. Strohman, "An Attention-based Joint Acoustic and Text On-device End-to-end Model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 7039–43.

[50] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, 3465–9, DOI: 10.21437/Interspeech.2019-1873.

[51] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Component Fusion: Learning Replaceable Language Model Component for End-to-end Speech Recognition System," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 5361–635.

[52] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold Fusion: Training Seq2Seq Models Together with Language Models," in *Proc. Interspeech 2018*, 2018, 387–91, DOI: 10.21437/Interspeech.2018-1392.

[53] Y. Tang, J. Pino, C. Wang, X. Ma, and D. Genzel, "A General Multi-task Learning Framework to Leverage Text Data for Speech to Text Tasks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, 6209–13.

[54] S. Thomas, B. Kingsbury, G. Saon, and H.-K. J. Kuo, "Integrating Text Inputs For Training and Adapting RNN Transducer ASR Models," *arXiv preprint arXiv:2202.13155*, 2022.

[55] A. Tjandra, S. Sakti, and S. Nakamura, "Listening While Speaking: Speech Chain by Deep Learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2017, 301–8.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems*, 30, 2017.

[57] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving Speech Recognition using Consistent Predictions on Synthesized Speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 7029–33.

[58] P. Wang, T. N. Sainath, and R. J. Weiss, “Multitask Training with Text Data for End-to-end Speech Recognition,” *arXiv preprint arXiv:2010.14318*, 2020.

[59] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, and N. Chen, “ESPnet: End-to-End Speech Processing Toolkit,” *Proc. Interspeech 2018*, 2018, 2207–11.

[60] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, “Hybrid CTC/Attention Architecture for End-to-end Speech Recognition,” *IEEE Journal of Selected Topics in Signal Processing*, 11(8), 2017, 1240–53.

[61] M. Wiesner, A. Renduchintala, S. Watanabe, C. Liu, N. Dehak, and S. Khudanpur, “Pretraining by Backtranslation for End-to-End ASR in Low-Resource Settings,” *Proc. Interspeech 2019*, 2019, 4375–9.

[62] Y. Yang, H. Xu, H. Huang, E. S. Chng, and S. Li, “Speech-text Based Multi-modal Training with Bidirectional Attention for Improved Speech Recognition,” *arXiv preprint arXiv:2211.00325*, 2022.

[63] J. W. Yoon, H. Lee, H. Y. Kim, W. I. Cho, and N. S. Kim, “TutorNet: Towards Flexible Knowledge Distillation for End-to-end Speech Recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021, 1626–38.

[64] S. Young, “A Review of Large-vocabulary Continuous-Speech,” *IEEE Signal Processing Magazine*, 13(5), 1996, 45.

[65] B. Yusuf, A. Gandhe, and A. Sokolov, “USTED: Improving ASR with a Unified Speech and Text Encoder-Decoder,” *arXiv preprint arXiv:2202.06045*, 2022.

[66] B. Zhang, H. Lv, P. Guo, Q. Shao, C. Yang, L. Xie, X. Xu, H. Bu, X. Chen, and C. Zeng, “WenetSpeech: A 10000+ Hours Multi-domain Mandarin Corpus for Speech Recognition,” *arXiv preprint arXiv:2110.03370*, 2021.

[67] Z. Zhang, L. Zhou, J. Ao, S. Liu, L. Dai, J. Li, and F. Wei, “Speechut: Bridging Speech and Text with Hidden unit for Encoder-Decoder based Speech-text Pre-training,” *arXiv preprint arXiv:2210.03730*, 2022.

[68] X. Zhou, G. Lee, E. Ylmaz, Y. Long, J. Liang, and H. Li, “Self-and-Mixed Attention Decoder with Deep Acoustic Structure for Transformer-Based LVCSR,” in *Proc. Interspeech 2020*, 2020, 5016–20, DOI: 10.21437/Interspeech.2020-2556.