information

Article

A Bidirectional Context Embedding Transformer for Automatic Speech Recognition

Lyuchao Liao [1,2], Francis Afedzie Kwofie [1,2,*], Zhifeng Chen [1,2], Guangjie Han [2,3], Yongqiang Wang [1,2], Yuyuan Lin [1,2] and Dongmei Hu [1,4]

[1] Fujian Key Laboratory of Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou 350118, China; fjachao@gmail.com (L.L.); qiming756@gmail.com (Z.C.); yqwang0374@gmail.com (Y.W.); 2201905137@smail.fjut.edu.cn (Y.L.); 52902379@ncepu.edu.cn (D.H.)
[2] Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology, Fujian University of Technology, Fuzhou 350118, China; hanguangjie@gmail.com
[3] College of Internet of Things Engineering, Hohai University, Changzhou 213022, China
[4] College of Environmental Science and Engineering, North China Electric Power University, Beijing 102206, China
* Correspondence: fakwofie2@gmail.com

Abstract: Transformers have become popular in building end-to-end automatic speech recognition (ASR) systems. However, transformer ASR systems are usually trained to give output sequences in the left-to-right order, disregarding the right-to-left context. Currently, the existing transformer-based ASR systems that employ two decoders for bidirectional decoding are complex in terms of computation and optimization. The existing ASR transformer with a single decoder for bidirectional decoding requires extra methods (such as a self-mask) to resolve the problem of information leakage in the attention mechanism This paper explores different options for the development of a speech transformer that utilizes a single decoder equipped with bidirectional context embedding (BCE) for bidirectional decoding. The decoding direction, which is set up at the input level, enables the model to attend to different directional contexts without extra decoders and also alleviates any information leakage. The effectiveness of this method was verified with a bidirectional beam search method that generates bidirectional output sequences and determines the best hypothesis according to the output score. We achieved a word error rate (WER) of 7.65%/18.97% on the clean/other LibriSpeech test set, outperforming the left-to-right decoding style in our work by 3.17%/3.47%. The results are also close to, or better than, other state-of-the-art end-to-end models.

Keywords: automatic speech recognition (ASR); speech transformer; bidirectional decoder; bidirectional embedding; end-to-end model; attention; bidirectional beam search

Citation: Liao, L.; Afedzie Kwofie, F.; Chen, Z.; Han, G.; Wang, Y.; Lin, Y.; Hu, D. A Bidirectional Context Embedding Transformer for Automatic Speech Recognition. *Information* **2022**, *13*, 69. https://doi.org/10.3390/info13020069

Academic Editor: Kostas Stefanidis

Received: 16 December 2021
Accepted: 23 January 2022
Published: 29 January 2022

1. Introduction

Automatic speech recognition (ASR) is the process whereby an algorithm is used to generate a sequence of words from a given speech signal. Traditional ASR systems usually consist of independent parts, such as an acoustic model, a pronunciation model, and a language model. These parts are trained separately and then combined for model inference. Over the years, several end-to-end models [1–6] have emerged to address this undesirable way of training by combining all the different parts into one network, thereby simplifying the training process.

Recently, the transformer [7], an end-to-end model that mainly relies on self-attention, has achieved superior results in natural language processing (NLP) tasks such as neural machine translation (NMT) and language modeling. In ASR, transformers have also achieved outstanding results in different modeling paradigms such as sequence-to-sequence [8], neural transducer [9], and connectionist temporal classification (CTC) systems [10,11]. A transformer network has a stack of encoder and decoder layers, and each layer performs

computations for the whole sequence in parallel, which greatly improves efficiency. The multi-head self-attention, which is a major component of the transformer, learns to directly connect related positions in the entire sequence. This allows the network to exploit long-range dependencies regardless of distance. This attention-based network has been found to be more parallelizable and can be trained faster than other end-to-end models, which are mostly based on recurrent neural networks (RNN).

However, transformer models built for ASR tasks are such that they are trained to decode output sequences in a left-to-right order without considering the right-to-left context. Extra components are needed to attend to the right-to-left context. Thus far, very few research works have been carried out in response to this issue. In the current literature, there are some works [12,13] that employ two decoders to tackle this issue. One decoder is used for attending to the left-to-tight context and another decoder is used for attending to the right-to-left context. This approach significantly boosts the prediction performance. However, utilizing two decoders for such a task presents an increased cost in both computation and optimization. There also exists works, such as that in [14], which utilize a single decoder for bidirectional decoding. Due to the bidirectional decoding implementation method at the architecture level, extra methods such as self-mask are required to resolve the problem of information leakage in the self-attention during model training.

This paper focuses on exploring different options to develop an improved speech transformer that utilizes a single decoder equipped with bidirectional context embedding (BCE) for bidirectional decoding. In particular, the decoding direction of each input sequence is set up at the input level of the decoder. This is in contrast to [14] where the decoding direction was implemented at the architecture level. Therefore, while this approach enables the model to generate bidirectional output sequences in a more efficient way without relying on any extra decoders, it also alleviates the possible leakage of information in the attention mechanism which was encountered by [14]. The left-to-right and right-to-left decoding subtasks are jointly optimized with shared weights. Unlike [12], utilizing only one decoder for such a task is desirable as it helps minimize the computational complexity. The effectiveness of this method was verified with a bidirectional beam search (BBS) method that generates bidirectional output sequences (i.e., left-to-right and right-to-left) and keeps the hypothesis with the best score as the final output. On the LibriSpeech dataset [15], the model achieves a word error rate (WER) of 7.65%/18.97% on the clean/other test sets, which is 3.17%/3.47% better than the left-to-right decoding style in our work. With fewer model parameters, the results are also close to or better than some state-of-the-art models. No language model or data augmentation techniques were employed in this work. Our main contributions in this paper are the following:

1. We propose to explore different options and implement an improved speech transformer that relies on a single decoder equipped with BCE for bidirectional decoding. This significantly minimizes computation complexity compared to methods that employ two decoders.
2. We trained the BCE, end to end, with unique sentence start tokens for each decoding direction, allowing the model with its single decoder to directly generate a right-to-left output without first generating a left-to-right output. This method alleviates the possible information leakage in the attention mechanism, which was encountered by other works.
3. A BBS method that generates bidirectional output sequences was implemented in the decoding stage, and used to perform extensive analysis to show the effectiveness of the model with different beam sizes. We also analyzed the performance of the model on different sequence lengths.

The rest of the paper is organized as follows. Section 2 provides a summary of other works related to this paper. Section 3 gives the details of the proposed method. Section 4 presents the details of the experimental work. Section 5 presents and discusses the obtained results. Section 6 provides concluding remarks and suggestions for future works.

2. Related Works

2.1. Existing Works on Transformers for ASR

The first transformer-based model for automatic speech recognition was proposed by the authors of [16] in 2018. In contrast to [7], they replaced the embedding layer of the encoder with convolution layers and reported promising results on the Wall Street Journal corpus [17]. The authors of [10] extended the work by introducing CTC into the transformer framework for a joint training and decoding. This approach made the training faster than with RNNs and also assisted in language model integration. Hang et al. [18] proposed a transformer-based dual-decoder system. The architecture was trained to perform both speech recognition and speech translation. Works such as those in [19–21] have investigated and proposed methods to improve the attention mechanism of the transformer for ASR. In [22], the authors conducted series of experiments to on RNNs against transformers for a number of ASR tasks and reported superior results with transformers. The authors of [23] also reported robust effects on the LibriSpeech benchmark with transformers within the hybrid system. Several works [24–26], as well as transformer transducers [9,27], have demonstrated the effectiveness of transformer for online speech recognition. The authors of [28] also employed an unsupervised pre-training method to enhance the transformer for ASR. In [29], the authors reported that transformers perform even better when trained on very large datasets.

2.2. Existing Works on Transformers with Bidirectional Decoders for ASR

In the current literature, very few works have been carried out on bidirectional transformers for speech recognition. In an attempt to utilize the right-to-left context in the transformer and improve the speech recognition performance, Chen et al. [12] proposed a novel ASR transformer which had a bidirectional decoder structure. Specifically, two decoders were employed in the structure, and each decoder exploits a specific direction (i.e., left-to-right or right-to-left). Wu et al. [13] proposed U2++, a novel bidirectional transformer, which can perform both online and offline speech recognition. Similarly, this was achieved by employing two decoders for the task. More recently, Zhang et al. [14] developed a novel non-autoregressive transformer that utilizes only one decoder for bidirectional decoding. The implementation exposed the model to information leakage, which resulted from the self-attention mechanism. Therefore, a new attention mask called "self-mask" was also proposed to address the issue.

In this work, a different approach (i.e., BCE) was used to achieve bidirectional decoding with the transformer. In contrast to [12], this method minimizes computation complexity as the model utilizes only a single decoder for the task. This method is also desirable as it alleviates the information leakage problem in [14].

3. Materials and Methods

This section provides a detailed explanation of the proposed transformer, which relies on BCE for bidirectional decoding. We start by giving a brief introduction of the conventional transformer used in previous works [12,16,20] for speech recognition in Section 3.1. We then proceed with the details of our network structure, the BCE method, and the adopted masking method in Section 3.2.

3.1. Overview of Transformers for ASR

The first transformer for speech recognition [16] is composed of an encoder and a decoder. The encoder and decoder contain the multi-head self-attention and position-wise feedforward network (FFN) as their major components. The encoder learns to map an input sequence, which is represented as $X = (x_1, \ldots, x_n)$ for a given sequence represented as $= (z_1, \ldots, z_n)$. Therefore, given Z, the decoder is able to generate, element by element, an output sequence, which is represented as $Y = (y_1, \ldots, y_m)$. The slight difference in architecture between the original transformer [7] for NMT and the speech transformer [16] lies in the encoder's input, where the embedding layers are commonly replaced with

convolutional neural network (CNN) layers to facilitate the extraction of rich acoustic features and also to perform sub-sampling on the input acoustic sequence.

3.1.1. Dot-Product Self-Attention

The transformer's self-attention usually consists of multiple attention heads. Given that the multi-head self-attention has h heads, the transformer computes the scaled dot-product attention h times, and their output is concatenated. A linear projection layer is built upon the scaled dot-product attention, which produces the final result from the concatenated outputs. Given $X \in \mathbb{R}^{Txd}$ as an input sequence, with T as the length of the sequence and d as the hidden size of the self-attention layer, each scaled dot-product attention head, as described by [20], can be computed as

$$Attention\ (Q_i, K_i, V_i) = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{1}$$

where the query $Q_i = XW^{Qi}$, key $K_i = XW^{Ki}$, and value $V_i = XW^{vi}$. We also have W^{Qi}, W^{Ki}, $W^{Vi} \in \mathbb{R}^{dxd_k}$ to denote the learnable projection parameter matrices for the i-th head, and $d_k = d/h$ represents the dimension of the feature vector for each head. The multi-head self-attention is defined as

$$Multihead(Q, K, V) = Concat(U_1, \ldots\ldots U_h)W^o \tag{2}$$

and

$$U_i = Attention\left(XW^{Qi}, XW^{Ki}, XW^{Vi}\right) \tag{3}$$

$W^o \in \mathbb{R}^{dxd}$ is the weight matrix of the linear projection layer.

3.1.2. Position-Wise FFN

Additionally, each layer of encoder and decoder of the transformer is usually equipped with a fully connected FFN. The FFN is made up of two linear layers with a ReLU activation employed between them. From [12], the output of the FFN is defined as:

$$F(x) = max(0; xW_1 + b_1)\ W_2 + b_2 \tag{4}$$

where $W_1 \in \mathbb{R}^{d_m \times d_f}$, $W_2 \in \mathbb{R}^{d_f \times d_m}$, $b_1 \in \mathbb{R}^{d_f}$, and $b_2 \in \mathbb{R}^{d_m}$. d_f denotes the feature dimension of the inner layer, and d_m denotes the feature dimension of the final outputs. The FNN is applied separately to each position in the sequence. Across different positions, the linear transformations are the same. However, there is a change in parameter across different layers. The FNN is an important component because it facilitates a richer representation by projecting the outputs of the attention.

3.1.3. Positional Encoding

With the transformer network, the output sequence does not depend on the input sequence order permutation. As the multi-head attention contains no recurrences or convolution layers, the transformer is unable to model the order of the input acoustic sequence. For this reason, "positional encodings" are used to learn the positional information in the input sequence. The positional encodings, as described by [7], are defined as

$$PE(pos; 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{5}$$

$$PE(pos; 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{6}$$

pos represents the position of the current frame or token in the current sequence, and i represents the dimension. The encoding values at each "even" and "odd" position are obtained by Equations (5) and (6), respectively. As different values are added to different dimensions of each input element, the position information is inserted into the attention process, allowing the model to learn the dependencies between different elements.

3.2. Bidirectional Context Embedding Transformer (Bi-CET)

To attend to both the left-to-right and the right-to-left contexts and improve the prediction accuracy of the transformer for speech recognition, we propose a speech transformer that utilizes a single decoder equipped with BCE for bidirectional decoding in one forward pass. In short, the architecture is denoted as Bi-CET.

3.2.1. Structure of Bi-CET

The Bi-CET architecture is such that the encoder is comprised of a stack of $M = 8$ layers. Each of the stacked encoder layer has the multi-head self-attention and the position-wise FFN as sublayers. The FNN has an inner-layer dimension of $d_{\text{f}} = 2048$, and the multi-head self-attention has vector size of $d_{\text{model}} = 512$. The decoder also contains a stack of $N = 4$ layers, and each stacked layer has three sublayers: the masked multi-head self-attention sublayer, the encoder-decoder attention sublayer, and the position-wise FFN sublayer. The mask employed around the multi-head self-attention sublayer masks the present positions from attending to the future positions during training. There is also layer normalization, which is added to each sublayer in the encoder and decoder. All the sublayers in both the encoder and decoder, as well as the linear and embedding layer, produce outputs with a dimension of $d_{\text{model}} = 512$. Before the input features were passed to the encoder, we employed the first two layers of a VGG [30] convolution block with layer normalization functions to extract the contextual information. VGG is a state-of-the-art CNN network that was originally implemented for large-scale image recognition tasks. The first and second layers of the VGG convolution have output channels of 64 and 128, respectively. Both layers have a kernel size, stride, and padding of 3, 1, and 1, respectively. We also used a 1D-CNN layer before the decoder to extract local features replacing the position embedding, which is used in most works. The ID-CNN layer has an output channel, kernel size, and stride of 512, 3, and 1, respectively. The softmax function was used to generate next-token probabilities from the decoder output. The structure of Bi-CET is shown in Figure 1.

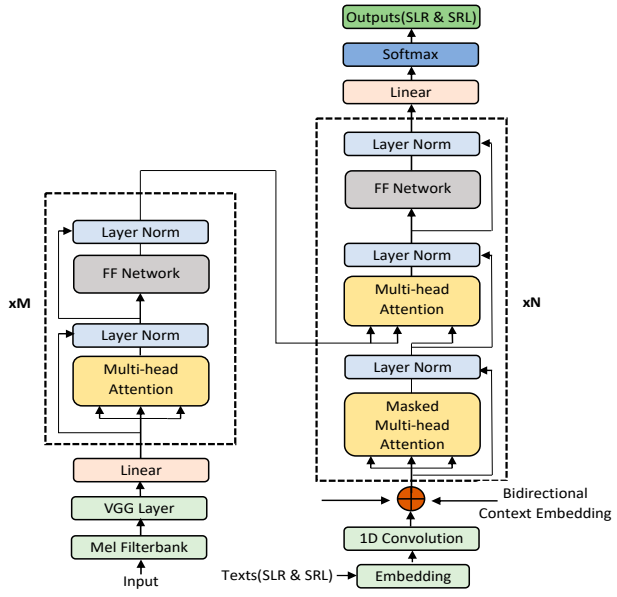

Figure 1. The structure of the bidirectional context embedding transformer (Bi-CET) for bidirectional decoding.

3.2.2. Setup of the BCE

Similar to the implementation in [31], the decoding direction of the output sequence is considered as two subtasks (i.e., left-to-right and right-to-left). In addition to the character embedding layer of the transformer, an additional 512-dimensional vector is also initialized during training. This allows for a two-time decoding to be performed on the input sequence in every batch during training: once from left-to-right and then from right-to-left. The ground truth transcription of the left-to-right decoded character sequence is reversed to serve as the ground truth transcription for the right-to-left character sequence. The method in [31] was initially proposed for a scene text recognition (STR) task where the maximum sequence length was 24. Moreover, the method only depended on the order of the input target sequences to set the decoding direction. This is because the two input targets to the decoder had a common start of sentence $\langle SOS \rangle$ token. Therefore, if there is only one input target, then only a left-to-right output sequence will be generated. If there are two input targets, then a left-to-right output sequence will be generated, followed by a right-to-left output sequence. We took the implementation further by looking at the following issues:

1. Avoiding the idea of always waiting for a second input to switch the decoding direction. It is possible to change the decoding direction with just one input.
2. Minimizing the decoding time in cases where only a right-to-left output is needed.

We therefore discarded the common $\langle SOS \rangle$ token and introduced $\langle SLR \rangle$ (i.e., sequence left-to-right) and $\langle SRL \rangle$ (sequence right-to-left) to serve as unique start tokens for the two directional contexts. We trained the BCE end to end with the rest of the model to learn and distinguish between the two directional targets and their corresponding decoding directions. The training process enhances the model's decision making on whether or not to maintain or change its current decoding direction. In this way, we do not necessarily have to provide two input targets before obtaining a right-to-left output sequence. We can directly obtain only a right-to-left output by only setting an $\langle SRL \rangle$ token at the input time, which saves a significant amount of time. We can also set two $\langle SRL \rangle$ tokens at the input time and obtain two right-to-left outputs in a row. The workflow of the BCE is shown in Figure 2. Additionally, as shown in Figure 1, the BCE is added on top of the 1D-CNN layer to provide extra contextual information for the model.

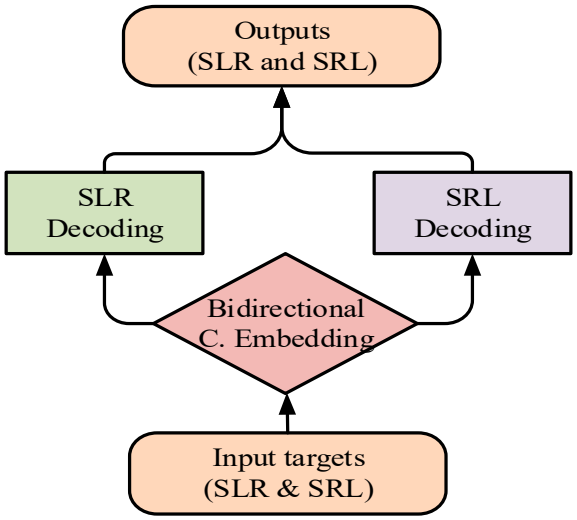

Figure 2. The workflow of bidirectional context embedding (BCE) during training.

3.2.3. Masking Method

We adopted the semantic masking method proposed by the authors of [32], which utilizes alignment information to perform token-wise masking. Therefore, we refer to [32] for the details of the implementation. During training, a percentage of the tokens is selected, and masking is applied to the corresponding speech sequence in each iteration. In contrast

to other methods [33] that purposely add noise to the input sequence by randomly masking the spectrum, this approach rather aims to train the model to learn a better language model. The authors explain that, in this way, a model may become less prone to over-fitting because the generation of words will not only depend on their corresponding speech features but also on other useful features, thus making training more effective even with noisy input signals

3.2.4. Character Decoding

To further improve model prediction accuracy, we implemented a BBS method to generate bidirectional output sequences using different beam sizes. To obtain a single output sequence, we can input only $\langle SLR \rangle$ or $\langle SRL \rangle$ as the start token. To perform bidirectional decoding, we input both of the sentence start tokens to provide more contextual information for the decoder. The specific method, based on that in [34], is illustrated in Algorithm 1.

Algorithm 1: Bidirectional Beam Search Method

Data: source(x), targets (SLR, SRL), beam size (β), max length (L_{max}), score (s1, s2)

Initialize: $SLR_0 \leftarrow \{\langle 0, \text{SLR}\rangle\}$, $SRL_0 \leftarrow \{\langle 0, \text{SRL}\rangle\}$
while not converged **do**
  for t in range $(1, \ldots\ldots\ldots, L_{max-1})$ **do** *//update left-to-right beam*
    SLR ← empty
    for $(\langle$s1, **u**$\rangle) \in SLR_{t-1}$ **do** *//*u *is the current state*
      if **u**.last() == EOS **then**
        SLR.add($\langle$s1, **u**$\rangle$) *//*s1 is the score for **u**
        continue
      for $n \in N$ **do** *//N contains neighbors of* **u**
        s1 = score(**x**, **u** ∘ n)
        SLR.add($\langle$s1, **u** ∘ n $\rangle$)
    end
    SLR_t ← SLR.top(β)
  end
  SLR.path ← SLR.max()
  for t in range $(L_{max-1}, \ldots\ldots\ldots, 1)$ **do** *//update right-to-left beam*
    SRL ← empty
    for $(\langle$s2, **v**$\rangle) \in SRL_{t-1}$ **do** *//***v** *is the current state*
      if **v**.last() == EOS **then**
        SRL.add($\langle$s2, **v**$\rangle$) *//*s2 is the score for **v**
        continue
      for $v \in V$ **do** *//V contains neighbors of* **v**
        s2 = score(**x**, **v** ∘ v)
        SRL.add($\langle$s2, **v** ∘ v $\rangle$)
    end
    SRL_t = SRL.top(β)
  end
  SRL.path ← SRL.max()
OUTPUT ← best score from (SLR.path, SRL.path)

The goal is to search for the best and most likely transcription based on the source sequence. The BBS algorithm works by selecting possible choices for an input sequence at each step. At each step, t, only the best, β (i.e., beam size), hypothesis is maintained to generate the output sequence of the next step. The decoding process continues, and we only pick the output with the best score for both the left-to-right and right-to-left outputs at the end. On completion, the scores of the two generated outputs are compared; the best one is chosen as the final output, and the remaining one is discarded. If the chosen output is an SRL sequence, the SRL will be reversed, and the word error rate will be calculated.

This method can generate bidirectional output sequences with equal or different sequence lengths. The Pytorch deep learning framework was used to implement this method.

4. Experiment

4.1. Dataset

All experiments were performed with the LibriSpeech dataset [15]. LibriSpeech is an English read speech corpus obtained from audiobooks, and it is publicly available. A total of 2484 speakers with accents close to those of the US were involved in the recording activities. A total of 1283 males and 1201 females participated, making the corpus more gender balanced. It has a 960 h (i.e.,100 h, 360 h, and 500 h) set with corresponding transcriptions for training. The 100 h and the 360 h sets are clean audio, while the 500 h set is noisier. The corpus also has additional "Dev" and "Test" sets, which are split into "clean" and "other" subsets for model evaluation or testing. The "clean" subsets are made up of clean audio with their corresponding transcriptions, while the "other" subsets have noisy and more challenging audio with their corresponding transcriptions. Each subset of the "Dev" and "Test" sets contains approximately 5 h of audio data. All of the audio data were sampled at 16 kHz. Due to memory limitations at the time of experiment, we only used the 100 h and 360 h sets for training and the "Dev" and "Test" sets for evaluation and testing.

4.2. Setup

We represented the input signals as a sequence of 80 dim log Mel filter banks, which were extracted with a frame size of 25 ms and a frame shift of 10 ms. The global cepstral mean and variance normalization (CMVN) were applied to normalize the features. We implemented two speech transformer models, Bi-CET$_{\text{small}}$ and Bi-CET$_{\text{big}}$, for the experiment. The smaller speech transformer model, Bi-CET$_{\text{small}}$, was set up with eight encoder layers, four decoder layers, and four attention heads. The attention vector size and feedforward dimensions were set as 256 and 1024, respectively. The bigger speech transformer model, Bi-CET$_{\text{big}}$, was set up with eight encoder layers, four decoder layers, and eight attention heads. We set the attention vector size as 512 and the feedforward dimension as 2048. Bi-CET$_{\text{small}}$ and Bi-CET$_{\text{big}}$ have total model parameters of 14 M and 46 M, respectively. The AdamW optimizer [35] was used with a warm-up strategy, and we set $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. β_1, β_2, and ϵ values are popular values that are widely used by state-of-the-art works, such as in [7,12], in setting up the optimizer. Since experimenting to test other values is time prohibitive, we adopted these settings in our work. The learning rate is defined as

$$lr = k \cdot \min\left(step^{-0.5}, step \cdot warmup\_steps^{-1.5}\right) \tag{7}$$

where we set k as 1.0, and *warmup_steps* was set as 16,000 and 25,000 for Bi-CET$_{\text{small}}$ and Bi-CET$_{\text{big}}$, respectively. The feedforward dropout was set as 0.2 to prevent or minimize any overfitting. For the masking strategy, we followed that implemented in [32]. The cross-entropy loss with label smoothing of 0.1 was applied. The batch size was set as 16. Using an NVIDIA 3080 GPU, Bi-CET$_{\text{small}}$ was trained on the 100 h set for 156 epochs, and Bi-CET$_{\text{big}}$ was trained on the combined 100 h and 360 h set for 58 epochs. In the decoding stage, we set the beam size as 2 for both unidirectional and bidirectional decoding. No language model or data augmentation technique was employed in this work. The entire setup was implemented with the Pytorch deep learning framework.

5. Results and Discussion

5.1. Unidirectional vs. Bidirectional

Using the BBS method, comprehensive experiments were conducted on the "Dev" and "Test" sets of the LibriSpeech dataset to verify the effectiveness of the proposed method. The "clean" and "other" subsets of "Dev", as well as the "clean" and "other" subsets of "Test", were all considered in this stage. We first carried out unidirectional decoding. To obtain

only left-to-right output sequences, we set $\langle SLR \rangle$ as the start token for the decoder input. We also set $\langle SRL \rangle$ as the start token to obtain only right-to-left output sequences. Next, we carried out bidirectional decoding where both $\langle SLR \rangle$ and $\langle SRL \rangle$ were fed to the decoder to generate bidirectional output sequences. Unidirectional and bidirectional decoding were conducted with both Bi-CET$_{\text{small}}$ and Bi-CET$_{\text{big}}$ models, and the performance was recorded in terms of WER. WER is the general criteria for assessing the performance of speech recognition models. The results of with/without bidirectional decoding are shown in Tables 1 and 2. The "Direction" column heading refers to the specific decoding direction chosen during decoding.

Table 1. Word error rate (WER) of Bi-CET$_{\text{small}}$ trained on LibriSpeech (100 h). We compare WER in percentage (%) using different decoding directions.

Direction	Test Clean	Dev Clean	Test Other	Dev Other
left-to-right	21.77	20.64	35.58	34.79
right-to-left	20.20	21.07	36.39	35.98
bidirectional	16.83	17.67	29.18	29.93

Table 2. WER (%) of Bi-CET$_{\text{big}}$ trained on LibriSpeech (460 h). We compare WER using different decoding directions.

Direction	Test Clean	Dev Clean	Test Other	Dev Other
left-to-right	10.82	10.76	22.44	22.97
right-to-left	9.98	10.64	23.12	23.26
bidirectional	7.65	7.85	18.97	19.33

We also visualized the training and validation losses of both the Bi-CET$_{\text{small}}$ and Bi-CET$_{\text{big}}$ models. This is shown in Figures 3 and 4.

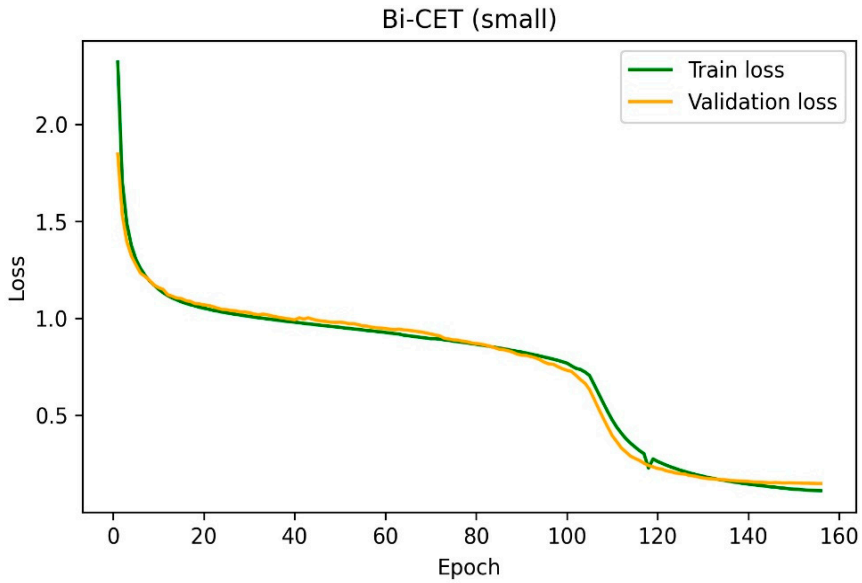

Figure 3. Visualization of training and validation loss of Bi-CET$_{\text{small}}$ trained on LibriSpeech (100 h).

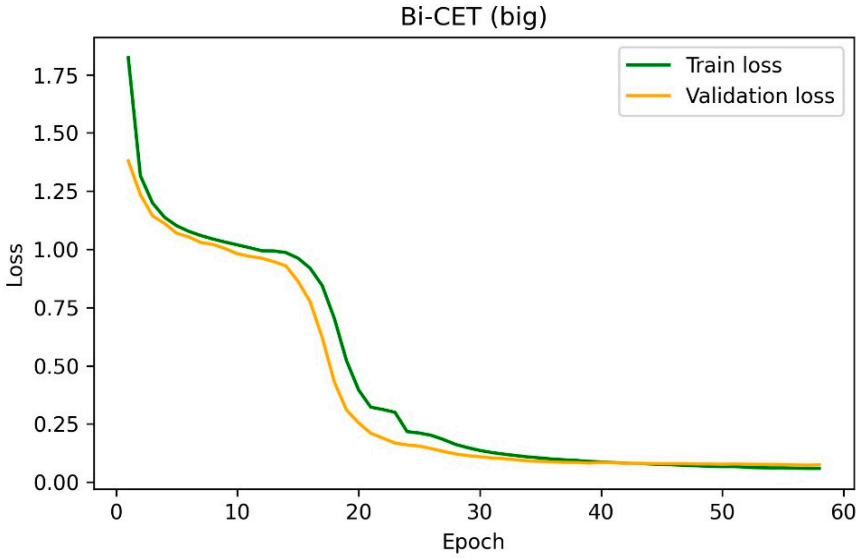

Figure 4. Visualization of training and validation loss of Bi-CET$_{big}$ trained on LibriSpeech (460 h).

In Table 3, we compare the model performance with other existing end-to-end models that were also trained on the LibriSpeech dataset.

Table 3. Comparison of WER (%) with other end-to-end models on the LibriSpeech benchmark.

Hours	Model	Train Type	Test		Dev		Network
			Clean	Other	Clean	Other	
100	Hsu et al. [36]	Supervised	14.85	39.95	14.00	37.02	Seq2Seq/TDS
	Kahn et al. [37]	Supervised	14.90	40.00	14.00	37.00	TDS/Attention
	Lüsher et al. [38]	Supervised	14.70	40.80	14.70	38.5	E2E/Attention
	Bi-CET$_{small}$	Supervised	16.83	29.18	17.67	29.93	Transformer
460	Ling et al. [39]	Semi-supervised	7.11	24.31	-	-	BLSTM/CTC
	Hsu et al. [36]	Supervised	7.99	26.59	7.20	25.32	Seq2Seq/TDS
	Bi-CET$_{big}$	Supervised	7.65	18.97	7.85	19.33	Transformer

All the models used for comparison in Table 3, including our work (i.e., Bi-CET$_{small}$ and Bi-CET$_{big}$), report their numbers without any language model. From the given results, it is shown that Bi-CET produces results that are somewhat close to or less than those of the other models on the "Clean" sets. It is slightly better than [36] on the 460 h set. On the "Other" subset of both the "Test" and "Dev" sets, Bi-CET significantly outperforms all other models on both the 100 h and 460 h sets. This confirms the impact of the semantic mask [32] method, which makes training more effective even with noisy input signals.

5.2. Discussion

During the bidirectional decoding process, it was observed that some sequences were better decoded in the left-to-right direction, while s others were better decoded in the right-to-left direction. The two generated output sequences were sometimes equal, close, or totally different in both length and score. If the output sequence with the best score is right-to-left, then the sequence will be reversed to calculate its WER against the ground truth transcription. Therefore, selecting the best result among the two decoding directions resulted in a massive improvement for the bidirectional decoding. With Bi-CET$_{big}$, 54% and 61% of the final output sequences were decoded in the right-to-left direction during the

bidirectional decoding on the "test" and "development" clean sets, respectively. Compared with the unidirectional decoding style, there is also a significant increase in decoding time with bidirectional decoding. This varies with different sequence lengths. From the results obtained, it can be seen that the BCE is very effective for bidirectional decoding in transformers and will have more advantages on large-scale datasets.

5.3. Further Analysis

5.3.1. Effect of Beam Size

Using Bi-CET$_{big}$, further tests were carried out on the "test clean" set to observe the effect of beam size when using the BBS method for bidirectional decoding. As shown in Figure 5, the size of the beam significantly affects the performance of the model. When the beam size was first increased from 1 to 2, a significant improvement in WER was observed. However, further increments in beam size only resulted in a slight improvement in WER, and then it remained constant or even decreased afterwards. It was also observed that the decoding speed or time increased with increase in beam size. Compared with both the left-to-right and the right-to-left methods, the bidirectional decoding method performed better.

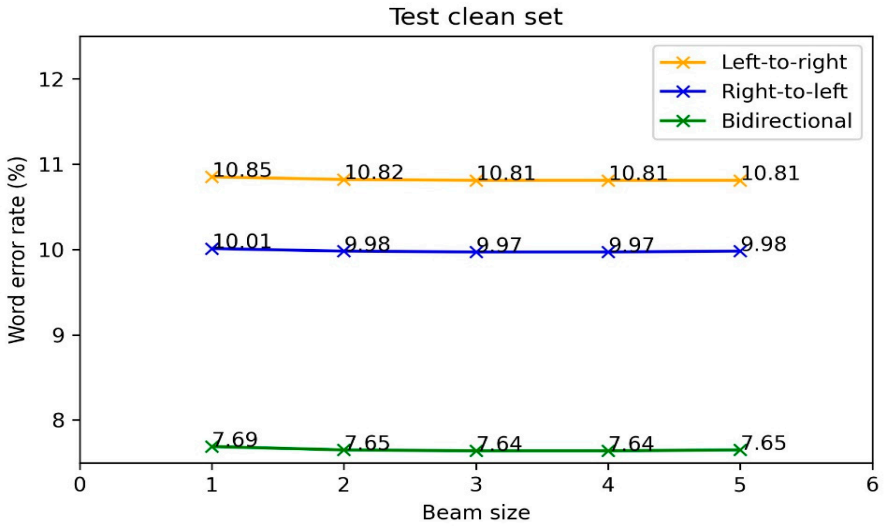

Figure 5. Word error rate with different beam sizes on Bi-CET$_{big}$.

5.3.2. Effect of Sequence Length

An additional analysis was carried out to investigate the effect of the BCE on different sequence lengths. Using Bi-CET$_{big}$ on the "test clean" set, the sequences were grouped according to their corresponding transcription lengths. The shortest lengths were in the range of 0–50, and the longest lengths were those greater than 250. The corresponding average WER of each group of sequences is shown in Figure 6.

It can be seen that the average WER first decreased as sequence length increased, and later began to rise with much longer sequences. It should also be noted that each group of sequences had different amounts of utterances. Sequences with lengths greater than 250 were the least in number.

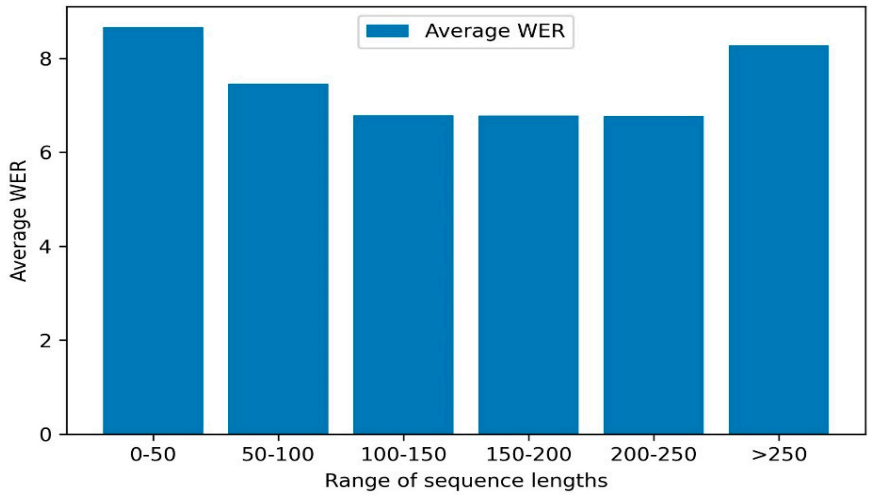

Figure 6. Average word error rate (WER) per sequence length on the "test clean" set.

6. Conclusions

In this paper, we explored different options and implemented Bi-CET, an improved speech transformer that relies on a single decoder equipped with BCE for bidirectional decoding. Bi-CET is such that the decoding direction of each target input is implemented at the decoder input level by adding extra contextual target information to the input. This is in contrast to methods such as that in [14] where decoding direction is conditioned at the architecture level. We showed that with a more efficient approach as compared to methods, Bi-CET significantly outperforms the traditional left-to-right or unidirectional style of decoding. The performance is also close to, or better than, other state-of-the-art end-to-end models that are trained on a similar size of datasets. Bi-CET utilizes a single decoder to exploit the same attention head for the two different directional contexts, which makes the computations less complex than in other approaches that employ two decoders [12]. It is evident that larger datasets will be necessary to achieve state-of-the-art results. Therefore, future works will include expanding the model and dataset sizes, and exploring other fascinating techniques to prepare the model for streaming tasks.

Author Contributions: Conceptualization, L.L., F.A.K., Z.C., G.H. and D.H.; methodology, F.A.K., L.L. and Z.C.; software, Y.W. and Y.L.; validation, F.A.K., D.H. and Y.L.; formal analysis, L.L., F.A.K., G.H. and D.H.; investigation, F.A.K., Z.C. and Y.L.; resources, L.L. and G.H.; data curation, Z.C., Y.W. and Y.L.; writing—original draft preparation, F.A.K., Z.C. and Y.W.; writing—review and editing, L.L., F.A.K. and G.H.; visualization, Z.C., Y.W. and Y.L.; supervision, L.L. and G.H.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the projects of the National Natural Science Foundation of China (41971340,41471333), the projects of Fujian Provincial Department of Science and Technology (2021Y4019,2020D002, 2020L3014, 2019I0019), and the support of the Foundation of Fujian Key Laboratory of Automotive Electronics and Electric Drive (Fujian University of Technology) (KF-X18002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found at: https://www.openslr.org/12, accessed on 6 July 2021.

Acknowledgments: We would like to express our profound gratitude to the authors of the dataset used in this work for releasing the corpus to the general research community.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gonzalez-Dominguez, J.; Eustis, D.; Lopez-Moreno, I.; Senior, A.; Beaufays, F.; Moreno, P.J. A real-time end-to-end multilingual speech recognition architecture. *IEEE J. Sel. Top. Signal Processing* **2014**, *9*, 749–759. [CrossRef]
2. Bosch, L.T.; Boves, L.; Ernestus, M. Towards an end-to-end computational model of speech comprehension: Simulating a lexical decision task. In Proceedings of the INTERSPEECH, Lyon, France, 25–29 August 2013.
3. Chorowski, J.; Bahdanau, D.; Cho, K.; Bengio, Y. End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. *arXiv* **2014**, arXiv:1412.1602.
4. Chan, W.; Jaitly, N.; Le, Q.V.; Vinyals, O. Listen, Attend and Spell. *arXiv* **2015**, arXiv:1508.01211.
5. Emiru, E.D.; Xiong, S.; Li, Y.; Fesseha, A.; Diallo, M. Improving Amharic Speech Recognition System Using Connectionist Temporal Classification with Attention Model and Phoneme-Based Byte-Pair-Encodings. *Information* **2021**, *12*, 62. [CrossRef]
6. Wang, X.; Zhao, C. A 2D Convolutional Gating Mechanism for Mandarin Streaming Speech Recognition. *Information* **2021**, *12*, 165. [CrossRef]
7. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *arXiv* **2017**, arXiv:1706.03762.
8. Zhou, S.; Dong, L.; Xu, S.; Xu, B. Syllable-Based Sequence-to-Sequence Speech Recognition with the Transformer in Mandarin Chinese. *arXiv* **2018**, arXiv:1804.10752.
9. Zhang, Q.; Lu, H.; Sak, H.; Tripathi, A.; McDermott, E.; Koo, S.; Kumar, S. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7829–7833.
10. Karita, S.; Yalta, N.; Watanabe, S.; Delcroix, M.; Ogawa, A.; Nakatani, T. Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019.
11. Miao, H.; Cheng, G.; Gao, C.; Zhang, P.; Yan, Y. Transformer-based online CTC/attention end-to-end speech recognition architecture. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6084–6088.
12. Chen, X.; Zhang, S.; Song, D.; Ouyang, P.; Yin, S. Transformer with Bidirectional Decoder for Speech Recognition. In Proceedings of the INTERSPEECH, Shanghai, China, 25–29 October 2020.
13. Wu, D.; Zhang, B.; Yang, C.; Peng, Z.; Xia, W.; Chen, X.; Lei, X. U2++: Unified Two-pass Bidirectional End-to-end Model for Speech Recognition. *arXiv* **2021**, arXiv:2106.05642.
14. Zhang, C.-F.; Liu, Y.; Zhang, T.-H.; Chen, S.-L.; Chen, F.; Yin, X.-C. Non-autoregressive Transformer with Unified Bidirectional Decoder for Automatic Speech Recognition. *arXiv* **2021**, arXiv:2109.06684.
15. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5206–5210.
16. Dong, L.; Xu, S.; Xu, B. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5884–5888.
17. Paul, D.B.; Baker, J.M. The Design for the Wall Street Journal-based CSR Corpus. In Proceedings of the HLT, Harriman, NY, USA, 23–26 February 1992.
18. Le, H.; Pino, J.; Wang, C.; Gu, J.; Schwab, D.; Besacier, L. Dual-decoder Transformer for Joint Automatic Speech Recognition and Multilingual Speech Translation. *arXiv preprint* **2020**, arXiv:2011.00747.
19. Shi, Y.; Wang, Y.; Wu, C.; Fuegen, C.; Zhang, F.; Le, D.; Yeh, C.-F.; Seltzer, M.L. Weak-Attention Suppression For Transformer Based Speech Recognition. *arXiv preprint* **2020**, arXiv:2005.09137.
20. Xu, M.; Li, S.; Zhang, X.-L. Transformer-based end-to-end speech recognition with local dense synthesizer attention. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 5899–5903.
21. Luo, H.; Zhang, S.; Lei, M.; Xie, L. Simplified self-attention for transformer-based end-to-end speech recognition. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 75–81.
22. Karita, S.; Chen, N.; Hayashi, T.; Hori, T.; Inaguma, H.; Jiang, Z.; Someki, M.; Soplin, N.E.Y.; Yamamoto, R.; Wang, X. A comparative study on transformer vs. rnn in speech applications. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Sentosa, Singapore, 14–18 December 2019; pp. 449–456.
23. Wang, Y.; Mohamed, A.; Le, D.; Liu, C.; Xiao, A.; Mahadeokar, J.; Huang, H.; Tjandra, A.; Zhang, X.; Zhang, F. Transformer-based acoustic modeling for hybrid speech recognition. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6874–6878.
24. Tsunoo, E.; Kashiwagi, Y.; Kumakura, T.; Watanabe, S. Transformer ASR with contextual block processing. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Sentosa, Singapore, 14–18 December 2019; pp. 427–433.
25. Wu, C.; Wang, Y.; Shi, Y.; Yeh, C.-F.; Zhang, F. Streaming transformer-based acoustic models using self-attention with augmented memory. *arXiv preprint* **2020**, arXiv:2005.08042.

26. Li, M.; Zorila, C.; Doddipatla, R. Transformer-Based Online Speech Recognition with Decoder-end Adaptive Computation Steps. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 1–7.
27. Huang, W.; Hu, W.; Yeung, Y.T.; Chen, X. Conv-Transformer Transducer: Low Latency, Low Frame Rate, Streamable End-to-End Speech Recognition. *arXiv* **2020**, arXiv:2008.05750.
28. Jiang, D.; Lei, X.; Li, W.; Luo, N.; Hu, Y.; Zou, W.; Li, X. Improving Transformer-based Speech Recognition Using Unsupervised Pre-training. *arXiv* **2019**, arXiv:1910.09932.
29. Lu, L.; Liu, C.; Li, J.; Gong, Y. Exploring transformers for large-scale speech recognition. *arXiv preprint* **2020**, arXiv:2005.09684.
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint* **2014**, arXiv:1409.1556.
31. Bleeker, M.; de Rijke, M. Bidirectional Scene Text Recognition with a Single Decoder. *arXiv* **2020**, arXiv:1912.03656.
32. Wang, C.; Wu, Y.; Du, Y.; Li, J.; Liu, S.; Lu, L.; Ren, S.; Ye, G.; Zhao, S.; Zhou, M. Semantic Mask for Transformer based End-to-End Speech Recognition. *arXiv* **2020**, arXiv:1912.03010.
33. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019.
34. Meister, C.; Cotterell, R.; Vieira, T. Best-First Beam Search. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 795–809. [CrossRef]
35. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. *arXiv* **2017**, arXiv:1711.05101.
36. Hsu, W.-N.; Lee, A.; Synnaeve, G.; Hannun, A.Y. Semi-Supervised Speech Recognition via Local Prior Matching. *arXiv* **2020**, arXiv:2002.10336.
37. Kahn, J.; Lee, A.; Hannun, A.Y. Self-Training for End-to-End Speech Recognition. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7084–7088.
38. Lüscher, C.; Beck, E.; Irie, K.; Kitza, M.; Michel, W.; Zeyer, A.; Schlüter, R.; Ney, H. RWTH ASR Systems for LibriSpeech: Hybrid vs. Attention—w/o Data Augmentation. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019.
39. Ling, S.; Liu, Y.; Salazar, J.; Kirchhoff, K. Deep Contextualized Acoustic Representations for Semi-Supervised Speech Recognition. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6429–6433.