Sequence-to-Sequence Learning via Attention Transfer for Incremental Speech Recognition

Sashi Novitasari[1], Andros Tjandra[1,2] Sakriani Sakti[1,2], and Satoshi Nakamura[1,2]

[1]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan
{sashi.novitasari.si3, andros.tjandra.ai6, ssakti, s-nakamura}@is.naist.jp

Abstract

Attention-based sequence-to-sequence automatic speech recognition (ASR) requires a significant delay to recognize long utterances because the output is generated after receiving entire input sequences. Although several studies recently proposed sequence mechanisms for incremental speech recognition (ISR), using different frameworks and learning algorithms is more complicated than the standard ASR model. One main reason is because the model needs to decide the incremental steps and learn the transcription that aligns with the current short speech segment. In this work, we investigate whether it is possible to employ the original architecture of attention-based ASR for ISR tasks by treating a full-utterance ASR as the teacher model and the ISR as the student model. We design an alternative student network that, instead of using a thinner or a shallower model, keeps the original architecture of the teacher model but with shorter sequences (few encoder and decoder states). Using attention transfer, the student network learns to mimic the same alignment between the current input short speech segments and the transcription. Our experiments show that by delaying the starting time of recognition process with about 1.7 sec, we can achieve comparable performance to one that needs to wait until the end.

Index Terms: incremental speech recognition, attention-based sequence-to-sequence model, attention transfer

1. Introduction

The demand for speech translation systems at meetings and lectures continues to increase. Since the length of complete sentences in such talks can be long and complicated, simultaneous speech translation is required to mimic human interpreters and translate the incoming speech stream from a source language to target language in real time. One challenge for achieving simultaneous speech translation is the development of incremental ASR.

Researchers have been working on speech recognition technology for decades. A number of techniques of real-time ASR exist, especially in the context of statistical ASR with a hidden Markov model (HMM) [1, 2, 3]. However, many current state-of-the-art ASR systems rely on attention-based sequence-to-sequence deep learning frameworks [4, 5]. Today's attentional mechanisms are based on a global attention property that requires the computation of a weighted summarization of the entire input sequence generated by the encoder states. This means that the system can only generate text output after receiving the entire input speech sequence. Consequently, utilizing it in situations that require immediate recognition is difficult.

Several studies proposed local attention mechanisms [6, 7] that limit the area explored by the attention by largely reducing the total training complexity without reducing the latency. For

work that enables incremental recognition of speech, Hwang and Sung employed a unidirectional RNN with a CTC acoustic model and a unidirectional RNN language model [8]. To avoid continuous output revision, they also proposed depth-pruning in the beam-search during the output generation. Jaitly et al. proposed a neural transducer framework [9] that incrementally recognizes the input speech waveforms. The formulation required inferring alignments during training, and they utilized a dynamic programming algorithm to compute "approximate" best alignments in each speech segments. Their model is strongly related to a sequence transducer that used connectionist temporal classification (CTC) [10, 11]. The improved version of a neural transducer, which has also been discussed [12], allows the attention mechanism to look back at many previous chunks without introducing additional latency.

However, most ISR models utilize different frameworks and learning algorithms that are more complicated than the standard ASR model. One main reason is because such models need to decide incremental steps and learn the transcription that is aligned with the current short speech segment. In this work, we propose attention-transfer ISR (AT-ISR) by the following:

1. employing the original architecture of an attention-based ASR for ISR tasks but with shorter sequences by treating the full-utterance ASR as the teacher model and the ISR as the student model.

2. utilizing attention transfer so that the student ISR model mimics the speech-text alignment produced by the standard ASR model.

3. investigating the impact of the input size to achieve the shortest speech chunks that can still produce reliable text output.

2. Related Works

A knowledge distillation method trains a student model, which is a simplification of a more complex model that acts as a teacher [13, 14]. A student network is commonly constructed as a compression version that is shallower or thinner that then trains the network to mimic the original teacher network by minimizing the loss (typically L2-norm or cross-entropy) between the student and teacher output.

Another approach is attention transfer, which was recently proposed by Zagoruyko and Komodakis [15] for image processing. Its basic idea is to ensure the spatial distribution of the student and teacher activations that are similar at selected layers in the network. Each layer in the student network is trained to focus on the same things as in the teacher network. Various tasks have also utilized attention transfer [16, 17], but not yet in ASR.

In this work, we applied attention transfer in speech recognition task by treating the full-utterance ASR as the teacher
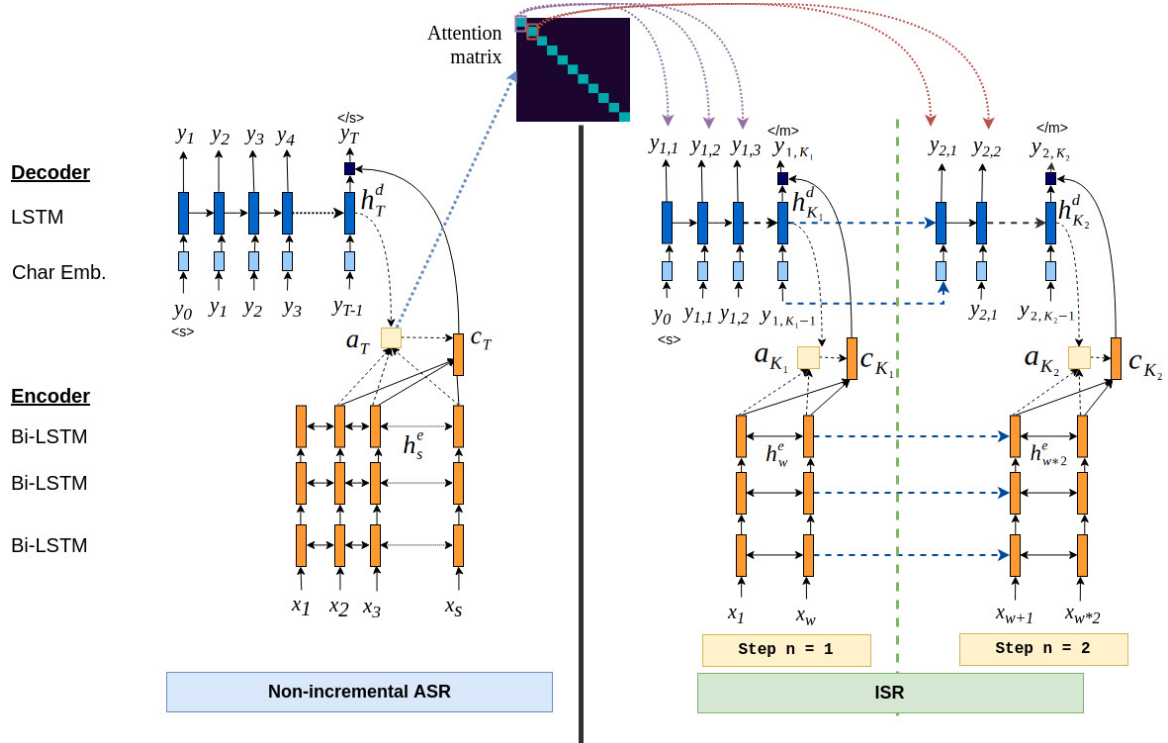
Figure 1: *Training method of proposed AT-ISR: Each recognition step in ISR takes an input segment. For each step, ISR refers the attention alignment from non-incremental ASR model. If current true output is aligned to the current segment, ISR performs a decoding step for it. If previous true output is the last character that should be aligned to the current segment, ISR decodes for the end-of-block </m> and moves to recognize the next segment.*

model and the ISR as the student model. Instead of using a thinner or shallower model, however, we design an alternative student network that retains the original architecture of the teacher model but with shorter sequences (only a few encoder and decoder states). In this way, no new redesign is needed for the ISR, and some hyperparameters can be used without changing them. With attention transfer, the student network learns to mimic the same alignment between the current input short speech segments and the transcription.

3. Overview of Sequence-to-Sequence ASR Architecture

Our works are based on the standard non-incremental character-level sequence-to-sequence ASR [18, 19]. It consists of encoder, decoder, and attention modules that can directly model the conditional probability of $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X}$ is a sequence of the framed speech features with length S and $\mathbf{Y}$ is a sequence of the characters with length T.

In this network, the encoder transforms the input speech sequence $\mathbf{X}$ to hidden representative information $\mathbf{h}^e = [h_1^e, ..., h_S^e]$ that will later be processed by the decoder. If down-sampling is applied in the encoder, each hidden state represents several number of input frames.

The decoder attempts to predict the target sequence probability p_{y_t}, given the previous output $\mathbf{Y}_{<t}$, the current context information c_t and the current decoder hidden state h_t^d. The context information c_t is produced by attention modules [20] at time t based on encoder and decoder hidden states with following formula:

$$c_t = \sum_{s=1}^{S} a_t(s) * h_s^e \qquad (1)$$

$$a_t(s) = \frac{exp(Score(h_s^e, h_t^d))}{\sum\limits_{s=1}^{S} exp(Score(h_s^e, h_t^d))} \qquad (2)$$

The scoring for the context can be done using one of the following scoring functions [21]:

$$Score(h_s^e, h_t^d) = \begin{cases} \langle h_s^e, h_t^d \rangle, & \text{dot product} \\ h_s^{e\mathsf{T}} W_s h_t^d, & \text{bilinear} \\ V_s^{\mathsf{T}} \tanh(W_s[h_s^e, h_t^d]) & \text{MLP,} \end{cases} \qquad (3)$$

and Score is $(\mathbb{R}^M \times \mathbb{R}^N) \to \mathbb{R}$, where M is the number of encoder hidden units and N is the number of decoder hidden units. The model loss function is formulated as:

$$\text{Ł}_{ASR}(\mathbf{Y}, \mathbf{p}_Y) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{c=1}^{C} \mathbb{1}(y_t = c) \log p_{y_t}[c], \qquad (4)$$

where C is the number of output classes.

4. Proposed Sequence-to-Sequence ISR

The proposed AT-ISR retains the original ASR architecture and performs several recognition steps incrementally with shorter sequences as shown in Fig. 1. In the character-based full-utterance ASR, a sequence of speech frames $\mathbf{X}$ with length S

transcribed as a character sequence $\mathbf{Y} = [y_1, y_2, ..., y_T]$ with length T. In the AT-ISR, for each recognition step, the model takes a segment of $\mathbf{X}$ with length W where $(W < S)$ and outputs an aligned segment of $\mathbf{Y} = [y_{n,1}, y_{n,2}, ..., y_{n,K_n}]$ with maximum length K where $(K_n \leq K < T)$ and n is the step index. Here, the window size W is equal for all steps.

In the AT-ISR training, we treat the full-utterance ASR as the teacher model and the ISR as the student model. With attention transfer, the student network will learn to mimic the same alignment between the input segments and the transcription as the teacher network. Therefore, inferring alignments during training is not necessary anymore.

During the training, for each incremental step n, AT-ISR learns to decode the characters that scored the highest monotonic alignment score to the current input segment or blocks from the distilled teacher's attention matrix. Attention matrix that generated by teacher consists of alignment scores of each character to each frame block. Therefore, the AT-ISR input segment size W will be equals to 1 frame block or multiple of it.

In training, if the decoding in AT-ISR reaches the last character that aligned to the current input segment, AT-ISR will learn to output an end-of-block symbol $</m>$ then move to the next segment. In the actual recognition with AT-ISR, the decoding in each step n will be done until the $</m>$ symbol is predicted or the output length in step n reaches K.

To accommodate the connection between each recognition step, we slightly modify the encoder and decoder parts:

- Encoder
  The modification in the encoder is done on the input side. AT-ISR encodes W frames in each step and the attention transfer during training will force the outputs to be aligned into W frames. As a character might actually be aligned to frames longer than W, we investigated two input settings of encoder: allow the model to look-back or look-ahead the main input block and not. If these additional context frames are included, the outputs of a step are the characters that aligned to the main input block.

- Decoder
  We used $<s>$ and $</s>$ symbols to define a sentence's beginning and end. In incremental recognition, we used $</m>$ symbols in the transcription to define the end of the block. In the decoding of the first input segment, AT-ISR takes $<s>$ symbol as the decoder's initial input. For the decoding of subsequent segments, we experimented on two types of initial input: beginning-of-block $<m>$ and the last output from the previous incremental step. Similar to the mechanism in the neural transducer, we allowed the model to produce no output from the current input segment.

5. Experimental Set-up

5.1. Model Configuration

The same attention-based encoder-decoder architecture is used for the non-incremental ASR and AT-ISR systems. The encoder part consists of an FNN layer and three bidirectional LSTM layers with a downsampling rate of two states in each of the LSTM layers. The first layer in the encoder takes a sequence of framed speech and outputs 512 features, and each LSTM layer outputs 256 features. Further details of our implementation of attention-based encoder-decoder architecture can be found here [22].

The encoder resulted in downsampled states with downsampling rate of eight. Therefore, in this experiment, a basic block of speech consists of eight frames (137.5 ms). The attention matrix from the non-incremental ASR aligned each character to each encoder state that represents eight frames.

The decoder side consists of an embedding layer, an LSTM layer with an attention mechanism, and a softmax layer. Here, we also applied an attention mechanism with an MLP-scoring function that utilized previously proposed multi-scale alignment and contextual history [23]. This mechanism maintains the history of the location and the contextual information of the previous time-steps, which improved the scoring function more than the standard attention. The standard attention applies the multi-scale convolution of the previous attention vector to the current decoder state.

5.2. Dataset and Features

Due to resource constraints, we first used the LJ Speech corpus [24] to find the best configuration of our ISR model. Then we applied the well-performed configurations on the Wall Street Journal (WSJ) corpus [25].

The LJ Speech dataset consists of 13.100 English speech utterances spoken by a single speaker (24 hours of speech). We simply divided the LJ Speech data into 12.314, 393, and 393 utterances as training, development, and test sets. The complete data of the WSJ corpus are contained in an SI284 (SI84+SI200) dataset. We followed the Kaldi [26] recipe for the training (*si84* and *si284*), development (*dev93*), and test (*eval92*) sets. SI84 consists of 7138 utterances (16 hours of speech) spoken by 83 speakers and SI200 consists of 30,180 utterances (66 hours of speech) spoken by 200 speakers without any overlap with the speakers of SI84.

All of the utterances in both datasets have 16-kHz sampling rates. From the speech utterances, we extracted the 80-Mel spectrogram for each frame with a window length of 50 ms that shifted by 12.5 ms from the previous frame.

6. Experiment Results

Our topline is the standard non-incremental ASR that conducted a greedy search for the decoding. We used an ISR without attention transfer learning as our baseline. The baseline ISR was trained identically as the non-incremental ASR and then tested incrementally by adding a zero vector at the end of the input segment without input overlap. Here, the decoding of each segment finished when the $</s>$ or $<blank>$ symbol (included in the vocabulary) predicted.

First, we compared the performances of topline ASR, baseline ISR, and AT-ISR as shown in Table 1. Here we also investigated the performance of the AT-ISRs with or without overlapping input as well as the different symbols for decoding initial input. One input segment in all ISRs in Table 1 consist of one main input block. In the models that allow input overlap, the input segment also includes one block ahead of the main block, resulting in 16 input frames. All of the models were evaluated based on the character error rate (CER).

The results in Table 1 show that the AT-ISR models significantly outperformed the baseline and achieved CER below 10%, similar to the topline. Among the proposed models, keeping the model states and transferring the learning greatly improved the performance. This configuration then applied in the further experiments. The lowest CER was achieved by allowing the model to overlap the input and by feeding the last character from the previous step as the decoder's initial input.

We further investigated the impact of the size of the additional blocks in the AT-ISR. Our results in Table 2 reveal that

Table 1: *CER (%) comparisons among topline ASR, baseline ISR, and AT-ISR with different approaches on LJ Speech*

Enc-Inp	Dec-Inp	Delay (sec)	Dev.	Test
Topline ASR		6.54 (avg)	2.84	2.78
Baseline ISR		0.14	79.63	80.34
AT-ISR - reset state				
No overlap	$<m>$	0.14	32.51	32.35
No overlap	last prev. char	0.14	26.15	26.52
Overlap	$<m>$	0.24	23.74	23.40
Overlap	last prev. char	0.24	23.40	14.22
AT-ISR - keep state				
No overlap	$<m>$	0.14	24.35	24.44
No overlap	last prev. char	0.14	22.69	23.04
Overlap	$<m>$	0.24	8.83	8.16
Overlap	last prev. char	0.24	**8.82**	**8.39**

Table 2: *AT-ISRs CER (%) with different number of additional blocks in each step on LJ Speech (1 main block=137.5ms)*

Look-back	Look-ahead	Delay (sec)	Dev.	Test
Non-incremental ASR			2.84	2.78
ISR				
0 block	0 block	0.14	22.69	23.04
0 block	1 block	0.24	8.82	8.39
0 block	2 blocks	0.34	5.68	5.32
0 block	4 blocks	0.54	**4.45**	**4.33**
1 block	1 block	0.34	7.60	6.86
2 blocks	1 block	0.44	7.03	6.80
4 blocks	1 block	0.64	7.09	6.89

looking ahead from the main block resulted in a better performance than looking back. The AT-ISR maintains the information from the previous steps, thus adding the previous block to the main block is not necessary. On the contrary, the ahead blocks able to support a better understanding of the main block by providing new information. By looking four blocks ahead, the AT-ISR achieved a performance with a smallest difference to the non-incremental ASR.

We also investigated the impact of the size of the main block in each recognition steps. Fig. 2 illustrates the performances of the proposed model where a single recognition step accommodates several blocks. In this figure, each model allows looking ahead to another block. The results show a clearer trade-off between time and performance. Although significant improvement happened until delay of 1.7 seconds, it did not happen similarly on subsequent sizes. Here, we investigated on how to reduce the delay without causing a significant decrease in ASR performance. Our experiments show that by delaying the starting time of recognition with about 1.7 sec, we can achieve a comparable performance to one that needs to wait until the end. This indicates that the student model performance with attention transfer approached that of the teacher model.

From our experiments on LJ Speech, we learned that the optimum performance in reasonable latency achieved the following: (1) included ahead blocks, (2) set the last character of the previous step as the decoder input, (3) kept the recurrent states across the steps, and (4) utilized the distilled knowledge of the attention matrix in the training. With this configuration,
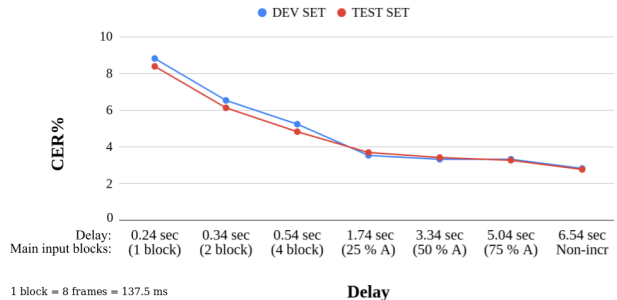

Figure 2: *AT-ISRs performance on LJ Speech with different main block size, 0 look-back, and 1 look-ahead block in each step. (A = average block number of an utterance in LJ Speech)*

we trained the ISR models with the WSJ dataset with 1 block as the main block for short latency. The results on *eval92* can be seen in Table 3. Here, we also compared our results with several published models of non-incremental ASR. Our results demonstrate that the AT-ISR can still performed with comparable performance with other published models. In order to balance the performance and latency, our experiments show that AT-ISR delay 0.54 sec or input segment with one main input block and four look-ahead blocks in each recognition step are sufficient.

Table 3: *CER (%) on eval92 set from topline ASR and AT-ISR models trained on WSJ-SI84 and WSJ-SI284.*

Models		WSJ CER (%)	
Non-Incremental ASR (Topline)		**SI84**	**SI284**
CTC [4]		20.34	8.97
Att Enc-Dec Content [4]		20.06	11.08
Att Enc-Dec Location [4]		17.01	8.17
Joint CTC+Att (MTL) [4]		14.53	7.36
Att Enc-Dec (ours)		17.05	6.80
AT-ISR			
Look-back	**Look-ahead**	**SI84**	**SI284**
0 block	1 block	30.81	19.78
0 block	4 blocks	**18.05**	**9.06**
Same delay configuration as Table 2			

7. Conclusions

We constructed a character-level AT-ISR framework that was trained with the original architecture of the attention-based sequence-to-sequence ASR model. The main difference is that it consists of shorter sequences than the standard architecture. No new redesign was needed for the ISR, and some hyperparameters can be used without any changes. Transfer learning treats the non-incremental ASR model as the teacher and the ISR as the student model. Student ISR learns the same attention alignment as the teacher model's, allowing a simple mechanism in the incremental recognition. Various types of models have been explored. The optimum performance was achieved by including a few ahead blocks, setting the last character of the last set as the decoder input, keeping the recurrent states across the steps, and utilizing the attention transfer.

8. Acknowledgements

9. References

[1] L. G. R. R. Paul Bamberg, Yen-lu Chow and D. Sturtevant, "The Dragon continuous speech recognition system: A real-time implementation," in *Proceedings of the Workshop on Speech and Natural Language*, Pennsylvania, USA, 1990.

[2] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "POCKETSPHINX: A free, real-time continuous speech recognition system," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006, pp. 185–188.

[3] O. Platek and F. Jurccek, "Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices," in *Proceedings the Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 108–112.

[4] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multitask learning," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017.

[5] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018.

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 4945–4949.

[7] A. Tjandra, S. Sakti, and S. Nakamura, "Local monotonic attention mechanism for end-to-end speech and language processing," in *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, Taipei, Taiwan, 2017, pp. 431–440.

[8] K. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 5335–5339.

[9] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain, 2016, pp. 5067–5075.

[10] A. Graves, "Sequence transduction with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning (ICML): Representation Learning Workshop*, 2012.

[11] ——, "Speech recognition with deep recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 6645–6649.

[12] T. N. Sainath, C.-C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, "Improving the performance of online neural transducer models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 5864–5868.

[13] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, USA, 2006, pp. 535–541.

[14] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[15] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[16] Q. Z. M. K. Junnan Li, Yongkang Wong, "Attention transfer from web images for video recognition," in *Proceedings of the ACM on Multimedia Conference (MM)*, Mountain View, USA, 2017.

[17] J. Yu, L. Marujo, J. Jiang, P. Karuturi, and W. Brendel, "Improving multi-label emotion classification via sentiment classification with dual attention transfer network," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, 2018, pp. 1097–1102.

[18] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 4960–4964.

[19] ——, "Attention-based models for speech recognition," in *Proceedings of the Advances in neural information processing systems (NIPS)*, Montreal, Canada, 2015, pp. 577–585.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[21] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[22] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Okinawa, Japan, 2017, pp. 301–308.

[23] ——, "Multi-scale alignment and contextual history for attention mechanism in sequence-to-sequence model," in *Proceedings of IEEE Workshop on Spoken Language Technology Workshop (SLT)*, Athens, Greece, 2018, pp. 648–655.

[24] K. Ito, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[25] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the Workshop on Speech and Natural Language (HLT)*, 1992, pp. 357–362.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Hawaii, USA, 2011.