Interspeech 2022
18-22 September 2022, Incheon, Korea

Internal Language Model Estimation Through Explicit Context Vector Learning for Attention-based Encoder-decoder ASR

Yufei Liu[1,2], *Rao Ma*[1], *Haihua Xu*[1], *Yi He*[1], *Zejun Ma*[1], *Weibin Zhang*[2]

[1] ByteDance AI LAB
[2]South China University of Technology, Guangzhou, China

Abstract

An end-to-end (E2E) ASR model implicitly learns a prior Internal Language Model (ILM) from the training transcripts. To fuse an external LM using Bayes posterior theory, the log-likelihood produced by the ILM has to be accurately estimated and subtracted. In this paper we propose two novel approaches to estimate the ILM based on the Listen-Attend-Spell (LAS) framework. The first method is to replace the context vector of the LAS decoder at every time step with a vector that is learned with the training transcripts. The second approach is to use a lightweight feed-forward network to directly map query vectors to context vectors in a dynamic sense. Since the context vectors are learned by minimizing the perplexities on the training transcripts, and their estimation is independent of the encoder output, hence the ILMs are accurately learned for both methods. Experiments show that the ILMs achieve the lowest perplexity, indicating the efficacy of the proposed methods. In addition, they also significantly outperform the shallow fusion method, as well as two previously proposed ILM Estimation (ILME) approaches on several datasets.

Index Terms: speech recognition, language model, attention-based, encoder-decoder, internal language model estimation

1. Introduction

End-to-end (E2E) Automatic Speech Recognition (ASR) models [1–4] are becoming more and more popular due to 1) its success in achieving state-of-the-art results; and 2) its compactness in that both the acoustic and language models are jointly learned with a single network. One of the most popular E2E models is the Listen-Attend-Spell (LAS) model [5], which is also called the attention-based encoder-decoder (AED) model.

Though compact and effective in modeling, E2E models are inherently limited in taking full advantage of an external language model (LM) that is trained on a much larger text-only data. This can be explained by the Bayes probabilistic theory that governs speech recognition models. Given an acoustic feature sequence X and the corresponding word/sub-word sequence W, an E2E model directly learns the posterior $P(W|X)$. $P(W|X)$ can be decomposed into an acoustic model $P(X|W)$ and a language model $P_{\texttt{prior}}(W)$. The language model $P_{\texttt{prior}}(W)$ is trained by using only the training scripts and thus is suboptimal. When more external text data is available, one can train a much more robust external language model $P_{\texttt{ext}}(W)$. To fuse $P(W|X)$ with the external LM $P_{\texttt{ext}}(W)$, the effect of the internal LM $P_{\texttt{prior}}(W)$ has to be removed first.

However, the difficulty lies in that we cannot easily estimate $P_{\texttt{prior}}(W)$ in an E2E model. Many simplified methods just ignore the effect of $P_{\texttt{prior}}(W)$, such as component fusion [6], cold fusion [7],and shallow fusion [8]. As the internal language model $P_{\texttt{prior}}(W)$ is implicitly retained in $P(W|X)$, the fusion process could be biased, yielding sub-optimal results in both intra- and inter-domain ASR scenarios.

Recently, Hybrid Autoregressive Transducer (HAT) [9] and Density Ratio approaches [10–12] were proposed as an extension of shallow fusion [13–17]. The Density Ratio method uses a separate model trained with training transcripts to approximate $P_{\texttt{prior}}(W)$. By contrast, HAT estimates the ILM by removing the effect of the encoder from the RNN-T network. Both methods have been shown to outperform shallow fusion, especially in cross-domain tasks. Inspired by HAT, Meng *et al*. proposed an Internal Language Model Estimation (ILME) method [18] to estimate the prior for both the RNN-T and the AED models.

Though yielding performance gains in [18], zeroing out the encoder's output may potentially lead to mismatch during inference. To deal with the mismatch issue, Meng *et al*. have proposed an ILM training method in [19] to update the model parameters engaged in predicting the ILM score more recently. Moreover, to relax the constraint of zeroing out the encoder's output, Zeineldeen *et al*. proposed a series of improved ILM estimation methods [20].

In this paper[1], we propose two novel ILME methods to estimate the ILM under AED framework. One of the simpler methods is to explicitly treat the decoder as an LM that is responsible for the ILM score estimation. Different from previous works, we propose to learn such an ILM through optimizing the context vector with the training transcripts after the normal ASR training. The advantage of the proposed method is the simplicity to interpret and yet can be applied to different AED framework. Alternatively, as the learned context vector is fixed during inference, we propose another method that allows for different context vector at each decoding step. Specifically, we propose to use a lightweight feed-forward network to map the query vector to the context vector.

2. Related work

Though the proposed work is inspired by diversified E2E ASR-based LM fusion works [6, 7, 21], as well as shallow fusion works [13–17], the most related ones are from [18], [19] and [20] respectively.

Our first proposed ILME method is equivalent to the combination of works [18] and [19] for AED models. We freeze all the parameters of AED network, and employ the training transcripts to minimize the perplexity of the decoder output, yielding a learned context vector from scratch. With the assistance of such a context vector, the decoder is more like a LM, namely an ILM. Here we don't have the zeroing out operation, nor other unnecessary assumptions.

Recently, Zeineldeen *et al*. proposed a series of improved

[1]This work was done when Yufei Liu was an intern in ByteDance.

Copyright © 2022 ISCA

10.21437/Interspeech.2022-606

ILME methods in [20], of which, the most effective method employs a "Mini-LSTM" network to estimate the decoding-synchronous context vector by taking decoding output as input. By a shallow contrast, our second ILME is very close to such a "Mini-LSTM" method. However, the difference is decisive. To estimate the context vector in this paper, we take the query vectors in the decoder, *i.e.*, hidden state vectors, as input instead. Not only do the query vectors contain context information, and hence more "internal", but they also let us sufficiently take advantage of what has been learned by the AED decoder, and therefore only a lightweight feed-forward network is required to learn the mapping. By comparison, since the "Mini-LSTM" takes the decoder output as input as mentioned, it is decoupled with the existing decoding network, and more related to the prior density ratio method [10].

3. Attention-based encoder-decoder ASR

The objective of the AED-based E2E model is to predict the posterior $P(\mathbf{W}|\mathbf{X};\theta^{\text{AED}})$ of a word sequence $\mathbf{W}$, given the input feature sequence $\mathbf{X}$. In an AED model [5], the encoder learns to map the feature representation $\mathbf{X}$ to a higher level representation $\mathbf{H}^{\text{enc}} \triangleq \{\mathbf{h}_\mathbf{1}^{\text{enc}}...\mathbf{h}_\mathbf{t}^{\text{enc}}...\mathbf{h}_\mathbf{T}^{\text{enc}}\}$, where the dimension and sequence length between $\mathbf{X}$ and $\mathbf{H}^{\text{enc}}$ are normally different due to the down-sampling operation. The attention network determines which subset of the sequence $\mathbf{H}^{\text{enc}}$ are to be attended, given the decoder's hidden state representation $\mathbf{H}^{\text{dec}} \triangleq \{\mathbf{h}_\mathbf{1}^{\text{dec}}...\mathbf{h}_\mathbf{i}^{\text{dec}}...\mathbf{h}_\mathbf{I}^{\text{dec}}\}$, where $\mathbf{h_i}^{\text{dec}}$ acts as a query vector. That is

$$\mathbf{a_i} = \text{AttentionNet}(\mathbf{H}^{\text{enc}}, \mathbf{h}_\mathbf{i}^{\text{dec}}) \tag{1}$$

$$\mathbf{c_i} = \sum_{t=1}^{T} a_{i,t}\mathbf{h}_\mathbf{t}^{\text{enc}} \tag{2}$$

where $\mathbf{a_i}$ is the attention weighting vector at each step i and $\mathbf{c_i}$ is the context vector that will be employed by the decoder to predict the next token. With the obtained context vector, the decoder proceeds as follows:

$$\mathbf{h}_\mathbf{i}^{\text{dec}} = \text{DecoderRNN}(\mathbf{h}_\mathbf{i-1}^{\text{dec}}, \text{Concat}(\mathbf{e}_\mathbf{i-1}^{\text{dec}}, \mathbf{c_{i-1}})) \tag{3}$$

$$\mathbf{z_i} = \mathbf{W_z}\mathbf{h}_\mathbf{i}^{\text{dec}} + \mathbf{b_z} \tag{4}$$

$$P(w_i|\mathbf{X}, \mathbf{W_{<i}}; \theta^{\text{AED}}) = \text{Softmax}(\mathbf{z_i}) \tag{5}$$

where $\mathbf{e}_\mathbf{i-1}^{\text{dec}}$ is the embedding vector corresponding to the output token w_{i-1}. w_{i-1} is normally a word-piece in practice. As a result, the word/token sequence posterior $P(\mathbf{W}|\mathbf{X};\theta^{\text{AED}})$ is obtained as the product of equation (5).

4. Proposed method

4.1. One-time context learning (OTCL) for ILME

From what is formularized in Section 3, if each context vector $\mathbf{c_i}$ in Eq. (2) is not estimated from the encoder's output, the decoder itself can be seen as an LM. This motivates us to learn a context vector by only using training transcripts. Specifically, during inference Eq. (2) is rewritten as

$$\mathbf{c_i} = \mathbf{c}\ , \forall\ \mathbf{i} \tag{6}$$

where $\mathbf{c}$ is a learned vector, and once it is learned it is kept fixed during inference. Thus, we call it one-time-context-learning (OTCL) based ILME. Notably, during the training of $\mathbf{c}$, all the other parameters of the AED model are kept fixed. The learned context vector, together with the previously learned decoder, forms the estimated ILM.

4.2. Label-synchronous context learning (LSCL) for ILME

In Section 4.1 a static context vector that is kept fixed during inference at each time step is learned. This may not be optimal. From Eq. (3), we can see that the context vector varies along with the decoding state $\mathbf{h}_\mathbf{i}^{\text{dec}}$ at every decoding step. The actual internal LM may benefit from this variation. Thus, we propose another ILME method, namely label-synchronous context learning (LSCL) method, where $\mathbf{c_{i-1}}$ is allowed to change at each decoding step as it is shown in Eq. (3). However, as a pure LM, $\mathbf{c_{i-1}}$ is not allowed to depend on the output of the encoder.

To achieve the objective as mentioned, we propose to generate the context vector by using only the current decoder state $\mathbf{h}_\mathbf{i}^{\text{dec}}$. Specifically, we use a nonlinear function $\mathbf{f}$ to do the mapping, *i.e.*, $\mathbf{c_i} = \mathbf{f}(\mathbf{h}_\mathbf{i}^{\text{dec}})$. As a result, our objective is to learn such a nonlinear mapping function. The learned mapping function, together with the decoder, forms the ILM. To make the learning simpler, we propose to use a lightweight Feed-Forward Neural Network (FFNN) for $\mathbf{f}(\cdot)$, *i.e.*

$$\mathbf{c_i} = \text{FFNN}(\mathbf{h}_\mathbf{i}^{\text{dec}}) \tag{7}$$

Training of the FFNN is similar to what is described in Section 4.1. Once the training of the entire AED model is done. We continue to train the FFNN to minimize the perplexity on the training transcripts. The parameters for the trained AED model are kept fixed during updating of the FFNN.

As mentioned in Section 3, Zeineldeen *et al.* recently proposed a similar context learning method, called "Mini-LSTM" in [20]. While the commonality is that both methods have introduced an extra network to estimate the context vectors for ILME, namely, "Mini-LSTM" versus "FFNN", the difference between two approaches are remarkable. In [20],the decoder's output $w_i \in \mathbf{W}$ is used as input, while we use decoder's state vector $\mathbf{h}_\mathbf{i}^{\text{dec}}$ instead. Since $\mathbf{h}_\mathbf{i}^{\text{dec}}$ has already embedded historical information, it is sufficient for us to model the mapping by a lightweight FFNN, reducing the risk of overfitting. Moreover, as we use hidden state vector, the FFNN is more coupled with the existing decoder, and hence it is more "internal". In contrast, the "Mini-LSTM" in [20] is actually decoupled with the decoder, and it is more close to a density ratio method. More importantly, our experiments in what follows show that the proposed method can yield better results than the Mini-LSTM-based ILME method.

4.3. ILME-based Language model fusion

For ILME-based LM fusion, we need three scores to proceed during inference. The three scores are, the output from the AED ASR decoder $P(\mathbf{W}|\mathbf{X};\theta^{\text{AED}})$, the score produced by the estimated ILM $P(\mathbf{W};\theta^{\text{AED}})$, and finally the score calculated by an external LM $P(\mathbf{W};\theta_{\text{ext}}^{\text{LM}})$. As a result, the final inference results are obtained with following equation:

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{\text{argmax}}[\log P(\mathbf{W}|\mathbf{X};\theta^{\text{AED}}) - \lambda_{\text{ILM}} \log P(\mathbf{W};\theta^{\text{AED}}) + \lambda_{\text{LM}} \log P(\mathbf{W};\theta_{\text{LM}}^{\text{ext}})] \tag{8}$$

where λ_{ILM} and λ_{LM} are the estimated ILM and external LM weighting factors respectively.

5. Experiments

5.1. Data

To verify the efficacy of the proposed methods for both intra- and inter-domain LM fusion, we conduct experiments on 5 training data sets over three languages, *i.e.*, English, Japanese, and Mandarin respectively. Table 1 reports the details of the training data distributions. There are three test sets. For English, we use Librispeech [22] `test-other` for both cross- and intra-domain tests. For Japanese, we use our in-house test data that contains 1538 utterances for intra-domain test. While for Mandarin, we have an in-house medical domain test set with 1092 utterances for cross-domain test.

Table 1: *Training data description: transcribed data for AED-based E2E ASR modeling and text data building LM for both intra- and cross-domain-based LM fusions."Giga" refers to Gigaspeech, "Libri" refers to Librispeech, while "house" is for "in-house".*

Language	E2E-ASR data		LM data		Domain
	Source	Hours	Source	Words	
English	house	18k	Libri.	853M	cross
	Giga.	10k			cross
	Libri.	960			intra
Japanese	house	100	house	913M	intra
Mandarin		100k		4183M	cross

5.2. Models and experimental setups

Three different LAS models with different encoders, *i.e.*, BLSTM, Transformer [23, 24], and Conformer [25] are trained for comparison. The BLSTM encoder is configured the same as in [5]. Together with the decoder, they form a conventional LAS model [5]. The Transformer's main parameters {layer, dim, head} are {18, 512, 8}, and the intermediate GLU layer size is 2048 with 0.1 dropout. The Conformer's parameters {layer, dim, head} are {12, 512, 8} and the intermediate SWISH layer size is also 2048 with 0.1 dropout. The convolution kernel size for the Conformer is 32. As for the decoder, we use the same architecture for different LAS models. The decoder is a 4 layers LSTM with 1024 hidden units. For LAS models trained on the English dataset, we use the Byte Pair Encoding (BPE) subword units with a vocabulary of 7000. For Japanese, the BPE is 8516. For Chinese, 8046 Chinese characters are used as the modeling units. The model used for external LMs is a 3-layer LSTM with 4096 units per layer. The feed-forward neural network in LSCL-ILME mapping is a fully connected 3 layers network with 512 units per layer. RELU is applied to the first two layers as the activation function. Both the learnable vector and the feed-forward network are trained with the initial learning rate of 0.001 and decay to the final learning rate of 0.0001 over 10000 steps. All the LM fusion parameters λ_{LM} and λ_{ILM} were tuned using grid search. Overall experiments are conducted on the Lingvo [26] platform.

5.3. Results

5.3.1. Cross-domain LM fusion

Table 2 reports the Word Error Rates (WERs) of the proposed methods for cross-domain LM fusion. The Librispeech `test-other` test data was used as the target domain, while the source ASR models were trained using the 18k-hours in-house English data. As can be seen from Table 2, the proposed methods, particularly the LSCL-ILME method, achieve consistently the lowest WER with different encoders. Though the proposed OTCL-ILME method is very simple, it achieves comparable results with the Mini-LSTM method. The Zero-out method works slightly better than the shallow fusion method. Finally, compared with the shallow fusion method, the proposed LSCL-ILME can achieve up to 22% relative WER reduction.

Table 2: *WER(%) on Librispeech `test-other`. Three AED models were trained with the 18k-hours in-house English data. Different fusion methods, including no fusion (None), Shallow Fusion (SF), Zero-out context (Zero), Mini-LSTM (LSTM), the OTCL-ILME (OTCL), and the LSCL-ILME (LSCL) are compared.*

Encoder type		None	SF	Zero	LSTM	OTCL	LSCL
BLSTM	WER	10.35	8.75	7.97	7.68	7.56	**6.88**
	λ_{LM}	0.0	0.15	0.25	0.3	0.35	0.35
	λ_{ILM}	0.0	0.0	0.1	0.15	0.15	0.25
Transformer	WER	8.94	7.44	7.11	6.35	6.29	**5.99**
	λ_{LM}	0.0	0.15	0.25	0.3	0.4	0.4
	λ_{ILM}	0.0	0.0	0.05	0.15	0.2	0.25
Conformer	WER	8.96	7.61	7.61	6.85	6.98	**6.41**
	λ_{LM}	0.0	0.1	0.1	0.15	0.25	0.25
	λ_{ILM}	0.0	0.0	0.0	0.15	0.1	0.15

Table 3 reports the Character Error Rates (CERs) on our in-house medical data set. The LAS model was trained on the 100k-hours Mandarin data. Due to space limitation, we only report the fusion results with the LAS model using Transformer encoder. From Table 3, the proposed LSCL-ILME fusion method achieves the best performance, making 28.57% relative CER reduction over what is obtained without LM fusion, while making 19.05% relative CER reduction over the shallow fusion result. More interestingly, its CER is 12.72% better than the Mini-LSTM's. We can notice that Zero-out method [18] achieves no CER improvement over shallow fusion method under the Transformer-encoder-based LAS ASR framework. More details about this will be described in Section 5.4.

Table 3: *CERs(%) on the in-house Chinese medical test data.*

Encoder type		None	SF	Zero	LSTM	OTCL	LSCL
Transformer	WER	6.72	5.93	5.93	5.50	5.50	**4.80**
	λ_{LM}	0.0	0.15	0.15	0.35	0.35	0.45
	λ_{ILM}	0.0	0.0	0.0	0.25	0.25	0.4

5.3.2. Intra-domain LM fusion

Table 4 reports CERs of our Japanese ASR system trained with 100 hours of data. Like Table 3, we only report the results of the Transformer-encoder-based LAS model. From Table 4, the proposed LSCL-ILME method again produces the best result of the overall fusion methods. The improvement is smaller compared with what is achieved in Table 3. This is understandable since the training data is rather small, with only 100 hours, as a result, the influence of ILM should be minor. However, we can notice though the training data is rather small ILME-based LM fusion methods are consistently effective, yielding improved results over the conventional shallow fusion result in Table 4.

Table 4: *CERs(%) on the in-house Japanese test data.*

Encoder type		None	SF	Zero	LSTM	OTCL	LSCL
Transformer	WER	24.64	23.83	23.66	22.84	23.37	**22.77**
	λ_{LM}	0.0	0.10	0.10	0.10	0.10	0.15
	λ_{ILM}	0.0	0.0	0.05	0.20	0.05	0.15

Table 5 reports the WERs of various intra-domain LM fusion methods. The source E2E models were trained on the 960-hours Librispeech data set and evaluated on the `test-other` test set. From Table 5, both the proposed methods win an obvious margin over the Zero-out and the shallow fusion methods. Meanwhile, the proposed LSCL achieves better results over the Mini-LSTM method.

Table 5: *WERs(%) on the Librispeech `test-other` test set with different intra-domain LM fusion methods.*

Encoder type		None	SF	Zero	LSTM	OTCL	LSCL
BLSTM	WER	7.13	6.22	5.44	5.17	5.18	**5.16**
	λ_{LM}	0.0	0.15	0.35	0.55	0.55	0.55
	λ_{ILM}	0.0	0.0	0.2	0.4	0.4	0.4
Transformer	WER	7.6	7.06	6.98	**6.51**	6.71	6.63
	λ_{LM}	0.0	0.1	0.25	0.25	0.2	0.2
	λ_{ILM}	0.0	0.0	0.15	0.15	0.15	0.1
Conformer	WER	6.3	5.8	5.48	5.19	5.19	**5.11**
	λ_{LM}	0.0	0.1	0.2	0.3	0.35	0.35
	λ_{ILM}	0.0	0.0	0.05	0.25	0.3	0.3

5.4. Ablation

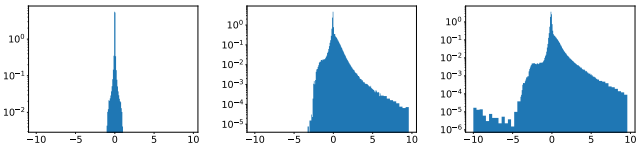
5.4.1. Perplexity of internal LM

One way to evaluate the efficacy of different ILME methods is to calculate the perplexity of the estimated ILM on the training text. To this end, we train the ILMs on Librispeech training transcript data. Specifically, we divide the entire transcript into 2 parts, 90% of utterances for training and 10% utterances for validation. Table 6 presents the perplexity of LAS models with different encoders and different ILME methods.

Table 6: *Perplexity of different LAS models and different internal language modeling methods evaluated on the held-out training transcript. ILM is trained on Librispeech data.*

Encoder Type	BLSTM	Transformer	Conformer
Zero-out	387	6247	3271
Mini-LSTM	240	460	477
OTCL-ILM	266	528	563
LSCL-ILM	**235**	**428**	**463**

From Table 6, we observe that the proposed LSCL-ILME method yields consistently the lowest perplexity among all methods. The performance is closely followed by the Mini-LSTM method. However, the perplexity of the Zero-out method [18] is very large, especially when it comes to Transformer or Conformer encoders. These abnormal perplexities explain why Zero-out method leads to rather poorer LM fusion results in the case of using Transformer and Conformer decoders. From Table 2 to Table 5. Directly zeroing-out context vector as an ILME method is probably not sensible. Therefore, we are curious about what context vector looks like for the three encoders. Figure 1 plots the distributions of context vectors from three different encoders.



(a) BLSTM (b) Conformer (c) Transformer

Figure 1: *The numeric distribution of context vector from three different encoders.*

We can see that the distribution of context vector from the BLSTM's encoder is symmetric. In addition. all values fall in the range of [-1,1]. Therefore, a zero-context vector is a reasonable assumption if the BLSTM is used as the encoder. On the other hand, the distribution of context vector for both the Transformer and the Conformer encoders are rather "messy". The dynamic range is also relatively large, leading to an "unpredictable" context vector. Thus zeroing-out the context vector leads to very large perplexities due to assumption mismatch.

5.4.2. Transformer Decoder

So far we conduct all experiments using LAS ASR framework. A good ILME method should be free of specific decoder architecture. Table 7 reports our ILME performance on Librispeech `test-other` test set, where the experiments are performed using Espnet [27] instead[2], and the model has a Conformer encoder and a Transformer decoder [28, 29]. The intra-domain ASR is trained with 960 hours of Librispeech data, while the cross-domain ASR is trained with 10k hours of Gigaspeech [30] data.

Table 7: *WERs(%) of the Librispeech `test-other` with Transformer decoder using proposed ILME methods.*

Encoder type		None	SF	OTCL	LSCL
Intra domain	WER	6.0	4.7	4.37	**4.35**
	λ_{LM}	0.0	0.7	0.9	0.9
	λ_{ILM}	0.0	0.0	0.6	0.6
Cross Domain	WER	7.9	5.56	4.89	**4.87**
	λ_{LM}	0.0	0.5	0.7	0.9
	λ_{ILM}	0.0	0.0	0.9	0.9

From Table 7, the proposed method has achieved significant performance improvement over the shallow fusion method in either case.

6. Conclusion

In this paper, we proposed two novel ILME methods by learning a static context vector or a mapping between the query vector and the context vector. Experiments on multiple datasets demonstrate the effectiveness of the proposed methods. Compared with shallow fusion and other previously proposed ILME methods, the methods proposed in this paper significantly reduce the error rate of the system. In the future, we would like to extend these methods to the recurrent neural network transducer-based ASR framework.

[2]Source code: https://github.com/victor45664/espnet/tree/ilme

7. References

[1] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[2] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[3] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[4] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*. PMLR, 2016, pp. 173–182.

[5] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[6] 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 *Proc. ICASSP*. IEEE, 2019, pp. 5361–5635.

[7] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," *arXiv preprint arXiv:1708.06426*, 2017.

[8] F. Stahlberg, J. Cross, and V. Stoyanov, "Simple fusion: Return of the language model," *arXiv preprint arXiv:1809.00125*, 2018.

[9] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (hat)," in *Proc. ICASSP*. IEEE, 2020, pp. 6139–6143.

[10] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Proc. ASRU*. IEEE, 2019, pp. 434–441.

[11] A. Zeyer, A. Merboldt, W. Michel, R. Schlüter, and H. Ney, "Librispeech transducer model with internal language model prior correction," *arXiv preprint arXiv:2104.03006*, 2021.

[12] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[13] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Proc. Interspeech*, vol. 2. Makuhari, 2010, pp. 1045–1048.

[14] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[15] T. Hori, S. Watanabe, and J. R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *Proc. ASRU*. IEEE, 2017, pp. 287–293.

[16] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *Proc. ICASSP*. IEEE, 2018, pp. 1–5828.

[17] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *Proc. SLT*. IEEE, 2018, pp. 369–375.

[18] 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 *Proc. SLT*. IEEE, 2021, pp. 243–250.

[19] Z. Meng, N. Kanda, Y. Gaur, S. Parthasarathy, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, "Internal language model training for domain-adaptive end-to-end speech recognition," in *Proc. ICASSP*. IEEE, 2021, pp. 7338–7342.

[20] M. Zeineldeen, A. Glushko, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "Investigating methods to improve language model integration for attention-based encoder-decoder asr models," *arXiv e-prints*, pp. arXiv–2104, 2021.

[21] 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 preprint arXiv:1503.03535*, 2015.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[24] S. Li, D. Raj, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "Improving transformer-based speech recognition systems with compressed structure and speech attributes augmentation." in *Proc. INTERSPEECH*, 2019, pp. 4400–4404.

[25] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[26] J. Shen, P. Nguyen, Y. Wu, Z. Chen, M. X. Chen, Y. Jia, A. Kannan, T. Sainath, Y. Cao, C.-C. Chiu *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *arXiv preprint arXiv:1902.08295*, 2019.

[27] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[28] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[29] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," *arXiv preprint arXiv:1804.10752*, 2018.

[30] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.