Advanced language model fusion method for encoder-decoder model in Japanese speech recognition

Daiki Mori*, Kengo Ohta†, Ryota Nishimura‡, Atsunori Ogawa§, and Norihide Kitaoka*
* Toyohashi University of Technology
† Anan National College of Technology
‡ Tokushima University
§ Nippon Telegraph and Telephone Corporation (NTT)

***Abstract*—We apply an advanced language model fusion method to the encoder-decoder models of a Japanese automatic speech recognition (ASR) system to reduce character error rates (CERs) in cross-domain scenarios. Our method, which uses a Density Ratio approach based on Bayes' theorem, is an extension of the Shallow Fusion method widely used for integrating the scores of speech recognition and language models. Since Japanese has a much larger number of characters than alphabet languages, and since there can be multiple possible meanings and pronunciations of the same kanji character, it is unclear whether the linguistic information inside a character-based end-to-end speech recognition is approximated by the language model.**

We conducted experiments using two types of encoder-decoder models, an RNN model and a Transformer model, and evaluated our results by calculating the cross-domain CERs of the ASR's text output when using the Japanese Academic Presentation Speech (APS) corpus and the Japanese Simulated Presentation Speech (SPS) corpus. When using the RNN model, our proposed method achieved a 1.2% lower CER in comparison to the Shallow Fusion method. When using the Transformer model, our method also outperformed the Shallow Fusion method, achieving a 0.7% lower CER.

I. Introduction

Conventional DNN-HMM ASR systems are very complex [1, 2], consisting of multiple modules such as acoustic, lexical and language models. On the other hand, end-to-end ASR models use a simple neural network structure to directly convert acoustic features into target symbols (e.g., a character or word) using RNNs such as an LSTM, allowing the ASR to perform high-speed recognition. In particular, the use of sequence-to-sequence (Seq2Seq) models with attention [3, 4] has significantly improved performance in natural language processing tasks such as speech recognition and machine translation. When performing speech recognition tasks, with a large enough training dataset these Seq2Seq models can model speech-to-text mapping well, but if a language model trained with large amounts of text data is also used, this rich linguistic information can be leveraged to achieve even better recognition performance. Since the language model can be trained with just text data, it is relatively easy to prepare a large amount of training data for the target domain. The Shallow Fusion method is a standard method used for integrating an ASR model and a language model [5, 6, 7, 8, 9]. It does this by adding the output probability of the ASR model and the output probability of the language model in the logarithmic region at the time of inference. Another proposed integration method, called Deep Fusion [10], inputs the hidden states of the pre-trained ASR model and the pre-trained language model into the neural-network. A modified version of Deep Fusion, called Cold Fusion [11], uses a pre-trained language model to train the ASR model from scratch. There are various other approaches which also utilize both ASR and language models, such as the Density Ratio approach [12]. All of the language model integration methods described above can improve the performance of Seq2Seq models, however there are problems with each of these integration methods. The ASR model contains "implicit linguistic information" since the training parameters of the ASR model are estimated so that the model will output accurate inference results for the training data. Therefore, when using the Shallow Fusion method, the linguistic information contained in the training data is reflected in the ASR model's parameters. In other words, the Shallow Fusion method adds the output probability of the language model for the target task to the output probability of the ASR model, which depends on the linguistic information contained in the training data. The Deep Fusion and Cold Fusion methods require re-training each time a language model is integrated. When using the Deep Fusion method, the DNN is trained using the hidden states of the pre-trained ASR model and the pre-trained language model as input. This means that if you replace the language model, you also need to re-estimate the parameters of the DNN model. For similar reasons, when using the Cold Fusion method, the ASR model is trained from scratch using the pre-trained language model, thus re-training is required when the language model is replaced. Because of these drawbacks, these methods have failed to replace the simple Shallow Fusion method as the go-to method for most of the ASR community. Part of the appeal of Shallow Fusion is that it does not require model retraining. This is purely applicable at the time of decoding.

The Density Ratio approach is an advanced language model fusion method which can be regarded as an extension of

Shallow Fusion using Bayes ' theorem. This approach approximates the ASR model with the linguistic information of the target domain by adding the output probability of the target domain language model to the output probability of the ASR model with its " implicit linguistic information " removed. This approach makes it possible to create an ASR model for domains which have little speech data available for training, by using only the textual data available for that domain.

The Japanese language is said to be a difficult or complex language system compared to Western alphabet languages, which are composed of about 52 uppercase and lowercase letters. Japanese has a much larger number of characters than the alphabet languages, numbering into the thousands, and there can be multiple homonyms and pronunciation variations for some kanji characters. In fact, there are three sets of Japanese characters: kanji (logograms), hiragana (a syllabary), and katakana (a syllabary used for non-Japanese words). In addition, Japanese contains complicated prefixes and suffixes, and has lax grammatical constraints (e.g., the subject of a sentence is often omitted), especially in daily conversation. In other words, Japanese does not conform to Western SVO grammar (subject + verb + object), making the recognition of spontaneous speech difficult. It has been reported that the linguistic information in speech recognition models in English, Spanish and Italian can be approximated by external language models, but it is still unclear whether the linguistic information trained in character- based end-to-end Japanese speech recognition can be approximated by a language model.

We conducted experiments using two types of Japanese encoder-decoder models: an RNN model and a Transformer model [13]. We calculated cross-domain CERs for recognition of the Japanese Academic Presentation Speech (APS) corpus and the Japanese Simulated Presentation Speech (SPS) corpus.

II. Background and related work

A. Language model integration methods

The standard method used to integrate ASR and language models is called Shallow Fusion. It adds the output probabilities of an ASR model and a language model in a logarithmic domain, which can be expressed using the following formulation:

$$y = \underset{y}{\operatorname{argmax}}\{\log P_{ASR}(y|x) + \lambda \log P_{LM}(y)\}, \quad (1)$$

where $\log P_{ASR}(y|x)$ is the output probability of the ASR model, which is the probability of inferring symbol label y when acoustic feature x is given, and where $\log P_{LM}(y)$ is the prior of the language model. When using the Shallow Fusion method, the language model is only used during inference, and the language model and ASR models are trained independently.

Another method of integrating the ASR and language models is called Deep Fusion, which can be expressed using the following equations:

$$g_t = \sigma(v^\top s_t^{LM} + b), \quad (2a)$$

$$s_t^{DF} = [s_t; g_t s_t^{LM}], \quad (2b)$$

$$y_t = \mathrm{softmax}(\mathrm{DNN}(s_t^{DF})), \quad (2c)$$

where $[s_t; g_t s_t^{LM}]$ is the concatenation of vector s_t and vector $g_t s_t^{LM}$. s_t, s_t^{LM}, and s_t^{DF} represent the hidden states of the pre-trained ASR model, pre-trained language model, and Deep Fusion model, respectively. The scalar g_t is a gate value trained using s_t and weight parameters v and b. In Equation (2c), the DNN is a deep neural network which can have any number of layers. The Deep Fusion method uses a pre-trained ASR model and a pre-trained language model, which are first trained independently. The ASR model and the language model are integrated by training the DNN, to which the hidden state information of each pre-trained model are fed.

In addition, a modified version of the Deep Fusion method called Cold Fusion has also been proposed. The ASR model is trained using linguistic information from a pre-trained language model. The Cold Fusion method can be expressed as follows:

$$h_t^{LM} = \mathrm{DNN}(l_t^{LM}), \quad (3a)$$

$$g_t = \sigma(W[s_t; h_t^{LM}] + b), \quad (3b)$$

$$s_t^{CF} = [s_t; g_t \circ h_t^{LM}], \quad (3c)$$

$$y_t = \mathrm{softmax}(\mathrm{DNN}(s_t^{CF})). \quad (3d)$$

Here, l_t^{LM} is the logit output of the language model, and s_t is the state of the ASR model. Gate value g_t is trained using state h_t^{LM}, state s_t of the ASR model, and weight parameters W and b. s_t^{CF} is a concatenation of the vectors obtained by the Hadamard product of s_t, g_t and l_t^{LM}. Therefore, the state of the ASR model (s_t) and the DNN output of language model (l_t^{LM}) are concatenated to integrate their information. In addition, it has been reported that the performance of Cold Fusion is improved by using the fine-grained (FG) gating mechanism as the gate algorithm [14].

As mentioned previously, several methods for integrating the ASR model and the language model have been proposed, and this has been reported to improve speech recognition performance.

B. A Density Ratio approach to language model fusion

A Density Ratio approach [12] is similar to Shallow Fusion but with an important difference, which is that this method also considers "implicit linguistic information". This is a key point which we would like to emphasize. During inference in ASR tasks, we try to infer the sequence $\hat{y}$ as follows:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\{\log P_{souce}(y|x)\}, \quad (4)$$

where $\log P_{source}(y|x)$ is the probability of output sequence y obtained from the ASR modeling "source task" when the input sequence is x. The "source task" refers to the task in which the speech data used for training was recorded. During this process, input sequence x and output sequence

y represent the input acoustic feature sequence and the output symbol label sequence, respectively. The log output probability $\log P_{souce}(y|x)$ from the ASR model can be expanded using Bayes' rule as follows:

$$\begin{aligned}\log P_{source}(y|x) &= \log P_{source}(x|y) \\ &\quad + \log P_{source}(y) - \log P_{source}(x) \\ &\propto \log P_{source}(x|y) + \log P_{source}(y).\end{aligned} \tag{5}$$

On the right side of Equation (5), we can see that the ASR model includes the acoustic information term $\log P_{source}(x|y)$ and the linguistic information term $\log P_{source}(y)$. This linguistic information, contained in the ASR model, is the "implicit linguistic information", the statistics of language obtained from the speech data which was used for training. The Shallow Fusion method does not take this "implicit linguistic information" into consideration. Seq2Seq end-to-end speech recognition utilizes this "implicit linguistic information", which improves decoding when the test task is the same as the training task. However if the test task is quite different from the training task, it can result in a degradation in decoding performance.

Based on Bayes' rule, this method removes the "implicit linguistic information" contained in the ASR model using a probabilistic approach. Let us assume the "implicit linguistic information" included in the ASR model from the source task can be approximated by the language model trained using text data from the source task. The "implicit linguistic information" can be removed by subtracting the output probability of the language model (trained using the text data from the source task) from the output probability of the ASR model for the source task, as follows:

$$\begin{aligned}&\log P_{source}(y|x) - \lambda_{sub} \log \tilde{P}_{source}(y) \\ &\quad \propto \log P_{source}(x|y) + \log P_{source}(y) - \lambda_{sub} \log \tilde{P}_{source}(y) \\ &\quad \approx \log P_{source}(x|y),\end{aligned} \tag{6}$$

where $\log \tilde{P}_{source}(y)$ is the probability of the language model for the source task, and λ_{sub} is a subtraction weight which balances the acoustic and linguistic information. This may compensate for the estimation error of P_{source}, which is the difference between P_{source} and $\tilde{P}_{source}$. Equation (6) can also be thought of as the log output probability of a pure acoustic model. It is then possible to replace the linguistic information by adding the output probabilities of the language model trained for the target task to Equation (6), as follows:

$$\begin{aligned}\log P_{source}(y|x) - \lambda_{sub} \log \tilde{P}_{source}(y) + \lambda_{add} \log \tilde{P}_{target}(y) \\ \approx \log P_{source}(x|y) + \lambda_{add} \log \tilde{P}_{target}(y) \\ \propto \log P_{(source,target)}(y|x),\end{aligned} \tag{7}$$

where $\log \tilde{P}_{target}(y)$ is the probability of the language model for the target task, and λ_{add} is an addition weight.

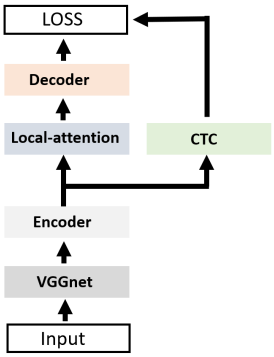

Fig. 1: Hybrid CTC/Attention Architecture.

$P_{(source,target)}(y|x)$ indicates that the acoustic information is for the source task, and the linguistic information is for the target task. It can also be said that $P_{(source,target)}(y|x)$ is the ASR model, where the acoustic information is the source task and the linguistic information is the target task.

Using Equations (4) and (7), our method successfully replaces only the linguistic information of the ASR model. Thus, it is possible to create an ASR model for any target task simply by preparing text data for that task.

Additionally, since this method integrates the language model only at the time of inference, like Shallow Fusion, retraining is not necessary, unlike the Cold Fusion and Deep Fusion approaches.

C. Espnet

ESPnet is an open-source speech processing toolkit for creating end-to-end models [15]. In our experiment, we used the RNN ASR and Transformer models provided by ESPnet. The RNN model uses a Hybrid CTC/Attention Architecture [16, 17, 18], while the Transformer model is a Joint CTC Attention Transformer. We also used the RNN language model [19] provided by ESPnet.

1) Hybrid CTC/Attention Architecture: Figure 1 shows a diagram of Hybrid CTC/Attention Architecture model. First, the input acoustic features are formatted using VGG-net, and are then converted into intermediate representation H by the six BLSTM (Bidirectional LSTM) layers, which are used as the encoder. The Seq2Seq decoder consists of one LSTM layer and one Linear layer. An additional Linear layer is used as the CTC decoder.

2) Joint CTC Attention Transformer: Figure 2 shows a diagram of the Joint CTC Attention Transformer model. The encoder consists of a stack of N = 18 identical layers. Each layer has two sub-layers, one of which is a multi-head self-attention mechanism while the other is a simple, locally fully connected feed-forward network. This method adopts a residual connection around each of the two sub-layers, followed by layer normalization. That is, the output of each sub-layer is Layer-Norm $(x + Sub - layer(x))$. The decoder also consists of a stack of Q = 6 identical layers. While

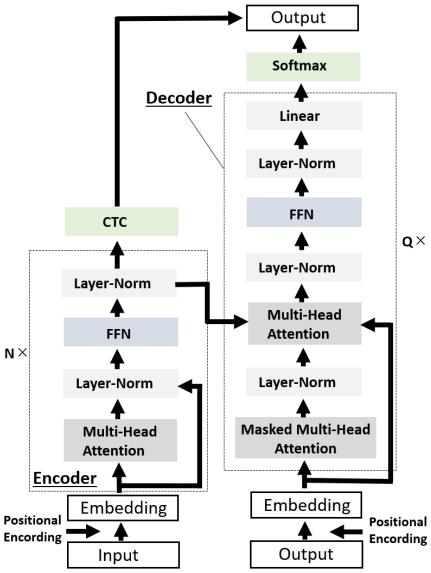

Fig. 2: Joint CTC Attention Transformer.

each encoder layer contains two sub-layers, each decoder layer contains three sub-layers in order to perform multi-head attention on the output of the encoder stack. Similar to the encoder, it uses a residual connection around each sub-layer, followed by layer normalization. It also modifies the self-attention sub-layer of the decoder stack so that positions do not join subsequent positions. Further, in order to supplement the acoustic information, a CTC composed of one Linear layer is used.

3) RNN language mdel: ESPnet can also be used to train the language model. We used the character-based RNN language model [19] in our experiment. This model consists of an Embedding layer, two LSTM layers and a Linear layer. During decoding, ESPnet allows the use Shallow Fusion with this language model.

D. Approaches similar to this research

A study by McDermott et al. [12], in which an RNN-T [20] was used for the ASR model, was a powerful motivation for our research. However, in contrast to that approach, we use integrated RNN and Transformer models for our ASR model, which are the models most commonly used for speech recognition. In [21], the authors also formulated their encoder-decoder model using Bayes ' rule, but they did not take scaling/normalization into account. Investigators in [22] showed that when the method proposed by McDermott et al. is applied to the encoder-decoder model, it is effective for ASR with English, Spanish and Italian. In this study, we wanted to verify that a similar approach is effective with Japanese, which is considered to be a relatively difficult language for speech recognition, especially in terms of language modeling. These other approaches, especially that used in [12], are more general, while in this paper we test this approach using

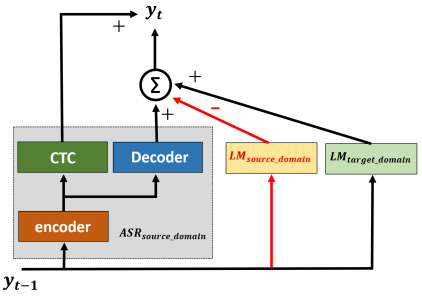

Fig. 3: The Density Ratio approach for encoder-decoder model.

an integrated LSTM-based and Transformer-based encoder-decoder model. From a theoretical point of view, this method should also be effective with other end-to-end encoder-decoder models, such as Conformer models [23].

III. The Density Ratio approach for encoder-decoder model

In this study, we apply the Density Ratio approach to encoder-decoder model. McDermott et al. [12] conducted experiments using an RNN-T decoder as an ASR model. However, when using an RNN-T in this manner, past recognition results and current acoustic features are directly converted into output symbol labels. In our study, in order to verify whether inference using the beam search algorithm is possible, experiments were conducted using a RNN model and a Transformer model.

The diagram in Figure 3 shows how the Density Ratio approach is applied to the encoder-decoder model. We applied this approach to the Seq2Seq decoder only. The output of the decoder is constructed using an RNN layer such as an LSTM, which tries to predict the current state from a previous state. In this manner, the decoder is trained to utilize linguistic information. On the other hand, when using CTC, the current state does not depend on the previous state, so CTC was used as an additional method, utilizing acoustic information.

We first subtract the output probability of the language model trained in the source domain, from the output probability of the decoder of the encoder-decoder model trained in the source domain. The subtraction layer of this language model is designed to cut out the linguistic information contained in decoder. The linguistic information can then be tuned to the target domain, by adding the output probabilities of the language model of the target domain.

IV. Experiments

A. Setup

We tested the Density Ratio approach experimentally using an ASR task, and compared the results to those obtained when performing the same task using the Shallow Fusion method. The results were evaluated using the character error rates (CER) for the test sets.

We prepared four types of ASR models for our experiment. The first ASR model was the Hybrid CTC/Attention Architecture trained using the Japanese Academic Presentation Speech

TABLE I: Details of datasets used in our experiments.

Corpus	Splits	Speakers	Utterances	Characters
Academic	Train	929	144,268	5,355,547
Presentation	Dev1	39	4,000	155,519
Speech	Dev2	10	1,272	45,169
(APS)	Test	10	1,292	44,915
Simulated	Train	1,654	232,886	6,026,797
Public	Dev1	38	4,018	129,738
Speech	Dev2	10	1,385	43,480
(SPS)	Test	13	1,336	29,610

TABLE II: Test set perplexities for character-based RNN language models trained using APS and SPS speech corpora.

Model	Perplexity of test set	
	APS	SPS
APS LM	17.19	34.62
SPS LM	33.29	18.87

(APS) corpus, which includes presentations on the subjects of engineering, humanities and sociology. The second ASR model was the Hybrid CTC/Attention Architecture trained using the Japanese Simulated Public Speech (SPS) corpus, which does not contain academic terminology. Both corpora are included in the Corpus of Spontaneous Japanese (CSJ) dataset. The third ASR model was the Joint CTC Attention Transformer trained with the APS dataset, and the last model was the Joint CTC Attention Transformer trained with SPS dataset. We used the Hybrid CTC/Attention Architecture and Joint CTC Attention Transformer models provided by ESPnet, as outlined in Section Ⅲ-A.

The utterances in each corpus are randomly divided into four sets: train, dev1, dev2 and test. The "train" set contains the training data for the ASR model, the "dev1" set is the development set for ASR model training, the "dev2" set is the development set for tuning the parameters of the Density Ratio approach or Shallow Fusion methods. The "test" set is the data to be used during testing. All of the datasets are open to each other. Table I shows the details of the datasets used in the experiments. The vocabulary size (number of character types) used in this experiment is 3031 in APS ASR model and 2755 in the SPS ASR model. The vocabulary of the language model was matched to the vocabulary of the ASR model in each experiment.

We used the character-based RNN language model outlined in Section Ⅲ-A as the language model. Table II shows the test set perplexities of the language model when trained with the training sets of the APS and SPS corpora. These results confirm that the language information contained in the two corpora are different. Note that similar results were obtained when using the vocabulary of APS ASR model and when using the vocabulary of SPS ASR model.

B. ASR Model Training

The following ASR models were used in this study.

1) Hybrid CTC/Attention Architecture: The acoustic features of the input speech for the ASR models were CMV-normalized using an 80 mel-scale filter bank. We expanded the training sets using speaking speed perturbation, adding voice data converted at factors of 0.9 and 1.1 to the original voice data. We used the Hybrid CTC/Attention Architecture outlined in Section Ⅲ-A as our RNN ASR models. We trained the entire models end-to-end using Ada-delta with a batch size of 24. We used early stopping with the dev1 set to prevent overfitting, and a patience value of 3 [24].

2) Joint CTC Attention Transformer: The training data is the same as in Section Ⅳ-B-1. In addition, per the original ESPnet model, spec augmentation (time warping, along with masking blocks of frequency channels and time steps) was conducted [25]. The time warp parameter was set to 5, and the ranges for frequency masking and time masking were randomly determined, between [0 to 30] and [0 to 40], respectively. Masking was performed twice in each case. We used the Joint CTC Attention Transformer outlined in Section Ⅲ-A for our Transformer ASR models. The number of heads of Multi-Head Attention in the encoder and decoder was set to 8. We trained the entire models end-to-end using Adam.

3) RNN language model: As outlined in Section Ⅲ-A, the character-based RNN language model consisted of an Embedding layer, two LSTM layers and a Linear layer. The input to the Embedding layer was a character one-hot vector. The batch size was 256, and SGD was used as the optimization function.

C. Experimental Results

We evaluated the performance of the Shallow Fusion method and the Density Ratio approach when used as the encoder-decoder methods, with each of the four ASR models described in Section Ⅳ-A. We then evaluated the cross-domain CERs of these integrated models using the APS and SPS test sets, respectively. We used the dev2 sets to tune addition weight λ_{add} of the Shallow Fusion method, and subtraction and addition weights λ_{sub} and λ_{add}, respectively, of the Density Ratio approach method.

Experimental results for the Hybrid CTC/Attention Architecture and Joint CTC Attention Transformer are shown in Tables Ⅲ and Ⅳ, respectively. For reference, we also show the CERs when the APS ASR model is evaluated using the APS test set, and when the SPS ASR model is evaluated using the SPS test set (i.e., CERs for matched cases).

When the APS ASR model (Hybrid CTC/Attention Architecture) and SPS test set were used, the CER for the Density Ratio Approach (DRA) method was 14.0%, outperforming the Shallow Fusion method, which achieved a CER of 15.2%. When the SPS ASR model (Hybrid CTC/Attention Architecture) and the APS test set were used, the CER when using the DRA method was 1.6% lower than when using the Shallow Fusion method. Using the APS ASR model (Joint CTC Attention Transformer) and the SPS test set, the DRA obtained a CER of 9.6%, which was lower than the Shallow Fusion method, which achieved a CER of 10.3%. When using the SPS ASR model (Joint CTC Attention Transformer) and the APS test set, the CER for the DRA method was 2.5% lower than that obtained when using the Shallow Fusion

TABLE III: Speech recognition results of Hybrid CTC/Attention Architecture for Shallow Fusion (SF) and Density Ratio approach (DRA).

Method	ASR model	Test set	Language model integration	λ_{sub}	λ_{add}	CER
Baseline	APS	SPS	—	—	—	16.2
SF			$ASR_{APS} + \lambda_{add} LM_{SPS}$	—	0.3	15.2
DRA			$ASR_{APS} - \lambda_{sub} LM_{APS} + \lambda_{add} LM_{SPS}$	0.9	0.9	**14.0**
Baseline	SPS	APS	—	—	—	16.2
SF			$ASR_{SPS} + \lambda_{add} LM_{APS}$	—	0.3	14.6
DRA			$ASR_{SPS} - \lambda_{sub} LM_{SPS} + \lambda_{add} LM_{APS}$	0.9	0.9	**13.0**
Matched	APS	APS	—	—	—	9.8
	SPS	SPS	—	—	—	8.0

TABLE IV: Speech recognition results of Joint CTC Attention Transformer for Shallow Fusion (SF) and Density Ratio approach (DRA).

Method	ASR model	Test set	Language model integration	λ_{sub}	λ_{add}	CER
Baseline	APS	SPS	—	—	—	10.6
SF			$ASR_{APS} + \lambda_{add} LM_{SPS}$	—	0.3	10.3
DRA			$ASR_{APS} - \lambda_{sub} LM_{APS} + \lambda_{add} LM_{SPS}$	0.9	0.9	**9.6**
Baseline	SPS	APS	—	—	—	10.7
SF			$ASR_{SPS} + \lambda_{add} LM_{APS}$	—	0.3	11.1
DRA			$ASR_{SPS} - \lambda_{sub} LM_{SPS} + \lambda_{add} LM_{APS}$	0.9	0.9	**8.6**
Matched	APS	APS	—	—	—	6.1
	SPS	SPS	—	—	—	7.7

method. Therefore, the Density Ratio approach improved (i.e., lowered) the CERs for ASR in the Japanese encoder-decode model, compared to Shallow Fusion. No significant difference was observed in the relative error rate between English and Japanese when using the Shallow Fusion and Density Ratio approaches.

Note that the addition and subtraction weights of the language models are important parameters for optimizing the performance of the proposed method. Figure 4 shows the details of the language model weights and the resulting CERs when using the Shallow Fusion method and the Density Ratio approaches in this experiment. The vertical axes of these figures show the subtraction weights of the language model, while the horizonal axes show the addition weights. Figures 4 (a) to (d) show the language model weights and associated CERs for the experiments using the Hybrid CTC/Attention Architecture, while Figures 4 (e) to (h) show language model weights and associated CERs when the Joint CTC Attention Transformer is used. The rows with subtraction weights of 0.0 (the boxes shown in blue) correspond to the CERs achieved when using various addition weights with Shallow Fusion, while the boxes marked in red show the CERs achieved when using various language model weights with the Density Ratio approach.

As can be seen in Figure 4, the CERs achieved when using language model weights optimized with the dev2 dataset, and the CERs achieved when using the language model weights optimized with the test dataset are not so different. Our results show that the proposed encoder-decoder model for Japanese works best when the subtraction and addition weights of the language model are set to about 1.0 when using the Density Ratio approach. McDermott et al. [12] reported that the optimal subtraction and addition weights of the language model were both about 0.5. It is unclear whether the difference between these results was caused by differences in the ASR model architectures used or by differences between the languages being recognized, thus further investigation is necessary.

V. Conclusions

We applied the Density Ratio approach, which is an extension of Shallow Fusion using Bayes' theorem, to the encoder-decoder of Japanese speech recognition models. In comparison to Western languages, Japanese is considered to be a more difficult language to model.

In our experiments, we evaluated two types of encoder-decoder models, an RNN model and a Transformer model. Cross-domain CERs were calculated after ASR when using the Japanese-language Academic Presentation Speech (APS) and Simulation Presentation Speech (SPS) corpora. When using a Density Ratio approach, our proposed RNN modeling method achieved a CER 1.2% lower than the Shallow Fusion method. When using the Transformer model, we achieved a CER 0.7% lower than the Shallow Fusion method. We observed that when using the Density Ratio approach, the encoder-decoder model worked best with Japanese when the subtraction and addition weights of the language models were set to around 1.0. In [12], the optimal subtraction and addition weights for the language model were both about 0.5. It is unclear whether this difference is due to the architecture of the ASR model or due to the use of different languages, so further investigation will be necessary.

Acknowledgment

This work was supported in part by JSPS KAKENHI Grant Numbers JP18K18170, 19H01125, 19K04311, and 21K13641.

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	19.1	18.3	**17.8**	18.1	19.3	21.8	25.3
0.1		15.6	15.1	15.0	15.7	16.7	18.2
0.3		15.9	14.9	14.4	14.7	15.5	16.8
0.5		16.7	14.9	14.2	14.0	14.6	15.5
0.7		18.2	15.6	14.2	13.6	13.8	14.6
0.9		21.7	17.1	14.9	13.8	**13.5**	14.0
1.1		30.5	20.6	16.6	14.5	13.8	13.7

(a) Language model weights on dev2 datasets in Hybrid CTC / Attention Architecture in SPS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	16.2	15.3	**14.6**	14.8	15.7	18.0	22.2
0.1		15.7	14.9	15.1	15.7	16.8	18.6
0.3		15.9	14.7	14.5	14.9	15.6	16.9
0.5		16.5	14.8	14.2	14.2	14.8	15.7
0.7		18.0	15.4	14.2	14.1	14.2	14.8
0.9		21.0	16.7	14.9	14.1	**14.0**	14.4
1.1		28.2	19.8	16.6	15.0	14.2	14.3

(b) Language model weights on test datasets in Hybrid CTC / Attention Architecture in SPS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	19.1	18.3	**17.8**	18.1	19.3	21.8	25.8
0.1		18.4	17.6	17.7	18.5	20.4	23.7
0.3		18.6	17.3	16.9	17.5	18.7	20.9
0.5		19.5	17.5	16.6	16.6	17.3	18.8
0.7		21.4	18.3	16.8	16.3	16.5	17.7
0.9		25.1	19.9	17.5	16.5	**16.2**	16.8
1.1		34.9	23.3	19.2	17.0	16.3	16.6

(c) Language model weights on dev2 datasets in Hybrid CTC / Attention Architecture in APS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	16.2	15.3	**14.6**	14.8	15.7	18.0	22.2
0.1		15.2	14.4	14.4	14.9	16.7	19.8
0.3		15.6	14.1	13.7	14.0	14.7	16.9
0.5		16.3	14.3	13.5	13.4	13.9	15.1
0.7		17.8	15.0	13.5	13.1	13.4	14.1
0.9		20.7	16.3	14.3	13.2	**13.0**	13.6
1.1		27.9	18.7	15.4	13.8	13.2	13.2

(d) Language model weights on test datasets in Hybrid CTC / Attention Architecture in APS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	10.2	10.4	**10.3**	10.4	10.7	11.4	12.5
0.1		10.4	10.1	10.2	10.5	11.1	12.0
0.3		10.3	10.0	9.9	10.1	10.5	11.1
0.5		10.6	10.0	9.8	9.8	9.9	10.5
0.7		12.1	10.2	9.7	9.6	9.7	10.0
0.9		26.9	10.9	9.8	**9.5**	**9.5**	9.7
1.1		47.8	23.0	10.5	9.7	**9.5**	9.7

(e) Language model weights on dev2 datasets in Joint CTC Attention Transformer in SPS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	10.6	10.4	**10.3**	**10.3**	10.8	11.3	12.1
0.1		10.4	10.2	10.2	10.6	11.0	11.7
0.3		10.3	10.0	9.9	10.2	10.6	11.1
0.5		10.4	9.9	9.7	9.8	10.1	10.5
0.7		12.3	10.0	9.6	9.6	9.8	10.1
0.9		26.3	10.9	9.7	**9.5**	9.6	9.9
1.1		43.0	22.7	10.4	9.6	9.7	9.8

(f) Language model weights on test datasets in Joint CTC Attention Transformer in SPS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	**12.4**	**12.4**	12.7	13.0	13.8	15.0	16.7
0.1		12.0	11.4	11.2	11.3	11.7	12.4
0.3		12.0	11.3	11.1	10.9	11.2	11.7
0.5		12.2	11.3	10.9	10.6	10.7	11.1
0.7		13.9	11.4	10.8	10.4	10.4	10.7
0.9		27.6	11.9	10.7	10.4	**10.2**	10.4
1.1		43.4	23.0	11.4	10.5	**10.2**	10.3

(g) Language model weights on dev2 datasets in Joint CTC Attention Transformer in APS domain

λ_{add}

λ_{sub}	0.0	0.1	0.3	0.5	0.7	0.9	1.1
0.0	**10.7**	10.8	11.1	11.6	12.3	13.4	15.4
0.1		10.3	9.6	9.4	9.4	9.8	10.4
0.3		10.3	9.5	9.2	9.2	9.2	9.8
0.5		10.4	9.5	9.1	8.9	9.0	9.3
0.7		12.0	9.7	9.1	8.8	8.8	9.0
0.9		24.8	10.2	9.2	8.8	**8.6**	8.7
1.1		39.7	20.2	9.5	8.8	**8.6**	8.6

(h) Language model weights on test datasets in Joint CTC Attention Transformer in APS domain

Fig. 4: Language model addition and subtraction weights and CERs (character error rate).
Blue areas correspond to the Shallow Fusion, whereas the red areas correspond to the Density Ratio approach.

REFERENCES

[1] Lawrence Rabiner and Bing-Hwang Juan, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[2] Dong Yu and Li Deng, Automatic Speech Recognition, Springer, 2015.

[3] Dzmitry Bahdanau, KyungHyun Cho and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in ICLR, 2015.

[4] Jan K. Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho and Yoshua Bengio, "Attention-based models for speech recognition,"in Advances in Neural Information Processing Systems, pp. 577–585, 2015.

[5] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in INTERSPEECH, 2010.

[6] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in Proc. Interspeech 2017, 2017, pp.523–527.

[7] Takaaki Hori, Shinji Watanabe, and John R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in Proc.ASRU, 2017.

[8] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, TaraSainath, Zhifeng Chen, and Rohit Prabhavalkar, "Ananalysis of incorporating an external language model into a sequence-to-sequence model," in Proc. IEEE ICASSP, 2018.

[9] Shubham Toshniwal, Anjuli Kannan, Chung-ChenChiu, Yonghui Wu, Tara Sainath, and Karen Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," CoRR,2018.

[10] Caglar Gulcehre, Orhan Firat, Kelvin Xu, KyunghyunCho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "On using monolingual corpora in neural machine translation," CoRR, vol.abs/1503.03535, 2015.

[11] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, andAdam Coates, "Cold fusion: Training seq2seq modelstogether with language models," in Proc. Interspeech.2018, ISCA.

[12] Erik McDermott, Hasim Sak, and Ehsan Variani, "A Density Ratio Approach to Language Model Fusion in End-to-End Automatic Speech Recognition," ASRU2019, pp. 434-441, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," 2017.

[14] Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W. Cohen and Ruslan Salakhutdinov, "Words or characters? Fine-grained gating for reading comprehension," in arXiv preprint arXiv:1611. 01724, 2016.

[15] Shinji Watanabe, Florian Boyer, Xuankai Chang, Pengcheng Guo, Tomoki Hayashi, Yosuke Higuchi, Takaaki Hori, Wen-Chin Huang, Hirofumi Inaguma, Naoyuki Kamo, Shigeki Karita, Chenda Li, Jing Shi, Aswin Shanmugam Subramanian, and Wangyou Zhang, "The 2020 ESPnet update: new features, broadened applications, performance improvements, and future plans, " in arXiv preprint arXiv:2012. 13006v1, 2020.

[16] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM, " in INTERSPEECH 2017.

[17] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey and Tomoki Hayashi. , "Hybrid CTC/Attention Architecture for End- to-End Speech Recognition,"in IEEE Journal of Selected Topics in Signal Processing, 2017.

[18] Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in Proceedings of the 23rd International Conference on Machine Learning, pp. 369-376, 2006.

[19] Tomas Mikolov, Statistical Language Models Based on Neural Networks. PhD thesis, Brno University of Technology, 2012.

[20] Alex Graves, "Sequence transduction with recurrent neural networks," CoRR, vol. abs/1211.3711, 2012.

[21] Zhuo Gong, Nobuaki Minematsu, and Daisuke Saito, "Language Model Augmentation in End-to-End ASR Systems Based on Noisy Channel Model," in Acoustical Society of Japan Spring Annual Meetings, (in Japanese), 2021.

[22] George Saon, Zoltan Tuske, Daniel Bolanos, and Brian Kingsbury, "ADVANCING RNN TRANSDUCER TECHNOLOGY FOR SPEECH RECOGNITION," ICASSP, 2021.

[23] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition,"in INTERSPEECH, 2020.

[24] Tsubasa Kobayashi, and Masashi Sugiyama, "Early stopping heuristics in pool-based incremental active learning for least-squares probabilistic classifier," in IEICE Transactions on Information and Systems E95.D(8):2065–2073.

[25] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," arXiv preprintarXiv:1904.08779, 2019.