Mori *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2024) 2024:40
https://doi.org/10.1186/s13636-024-00360-8

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

Recognition of target domain Japanese speech using language model replacement

Daiki Mori[1], Kengo Ohta[2], Ryota Nishimura[3], Atsunori Ogawa[4] and Norihide Kitaoka[1*]

Abstract

End-to-end (E2E) automatic speech recognition (ASR) models, which consist of deep learning models, are able to perform ASR tasks using a single neural network. These models should be trained using a large amount of data; however, collecting speech data which matches the targeted speech domain can be difficult, so speech data is often used that is not an exact match to the target domain, resulting in lower performance. In comparison to speech data, in-domain text data is much easier to obtain. Thus, traditional ASR systems use separately trained language models and HMM-based acoustic models. However, it is difficult to separate language information from an E2E ASR model because the model learns both acoustic and language information in an integrated manner, making it very difficult to create E2E ASR models for specialized target domain which are able to achieve sufficient recognition performance at a reasonable cost. In this paper, we propose a method of replacing the language information within pre-trained E2E ASR models in order to achieve adaptation to a target domain. This is achieved by deleting the "implicit" language information contained within the ASR model by subtracting the source-domain language model trained with a transcription of the ASR's training data in a logarithmic domain. We then integrate a target domain language model through addition in the logarithmic domain. This subtraction and addition to replace of the language model is based on Bayes' theorem. In our experiment, we first used two datasets of the Corpus of Spontaneous Japanese (CSJ) to evaluate the effectiveness of our method. We then we evaluated our method using the Japanese Newspaper Article Speech (JNAS) and CSJ corpora, which contain audio data from the read speech and spontaneous speech domain, respectively, to test the effectiveness of our proposed method at bridging the gap between these two language domains. Our results show that our proposed language model replacement method achieved better ASR performance than both non-adapted (baseline) ASR models and ASR models adapted using the conventional Shallow Fusion method.

Keywords End-to-end speech reccognition, Implicit language information, Language model replacement

*Correspondence:
Norihide Kitaoka
kitaoka@tut.jp
[1] Department of Computer Science and Engineering, Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku-cho, Toyohashi 441-8580, Aichi, Japan
[2] Department of Creative Technology Engineering, National Institute of Technology, Anan College, 265 Aoki Minobayashi-cho, Anan 774-0017, Tokushima, Japan
[3] Graduate School of Technology, Industrial and Social Sciences, Tokushima University, 2-1 Minamijohsanjima, Tokushima 770-8506, Tokushima, Japan
[4] Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, 2-4, Hikaridai, Seika 619-0237, Kyoto, Japan

1 Introduction

Traditional automatic speech recognition (ASR) systems, such as those based on Gaussian mixture model HMM (GMM-HMM) or deep neural network HMM (DNN-HMM), are very complex, consisting of various modules such as acoustic models, dictionaries, and language models [1, 2]. On the other hand, end-to-end (E2E) ASR models, which use deep learning, can represent these complex speech and language processes using a single neural network. A wide variety of E2E ASR models have been proposed over the past few years, such as those based on long short-term memory (LSTM) [3, 4]

and on Transformer [5–8] with an attention mechanism [9], which have been used with great success in the field of natural language processing (NLP) for tasks such as machine translation [10, 11]. E2E ASR models based on connectionist temporal classification (CTC) [12] and Transducer [13–16] have also been proposed.

As a result of recent advances in ASR technology and increasing ease of use, we have seen greater use of ASR models in various commercial applications. For example, ASR models are now used in AI speakers and in speech assistants such as Alexa [17] and Siri [18], which has made ASR technology more and more familiar to the public. However, training such ASR models requires a large amount of speech and transcription data. We have found, in our own previous research, that creating a dataset for a target domain (e.g., the medical domain) for an ASR model is expensive in terms of both time and money. Therefore, in Japan, we often train ASR models for commercial use with publicly available datasets such as the Corpus of Spontaneous Japanese (CSJ) [19] or the LaboroTVSpeech corpus [20] or use publicly available, large-scale ASR models pre-trained in generic domains. However, general domain ASR models and ASR models trained with publicly available datasets may not perform as required in a target domain environment. In fact, we have had great difficulty creating an ASR model for a specialized domain.

Against this background, a method to adapt existing large-scale ASR models to a target domain would be very useful. Currently, fine-tuning is the most popular and effective domain adaptation method for ASR tasks [21]. This involves re-training a large-scale, out-of-domain ASR model with a small amount of target domain speech and transcription data, in order to create a target domain-adapted ASR model. Many studies have proposed efficient fine-tuning methods which use limited computing resources, such as Adapters [22]. To fine-tune an ASR model, it is generally necessary to prepare several hours of target domain training data, which includes speech and its transcription; thus, there is still the problem of the cost of preparing target domain training data.

Another effective method for domain adaptation of ASR models is to use ASR models in combination with external language models [23–26], the most common method of which is Shallow Fusion [27, 28]. Other methods for combining ASR models with external language models have also been proposed, including Cold Fusion [29] using gate mechanisms [30], Component Fusion [31], and Deep Fusion [11]. All of these language model integration methods improve ASR performance; however, there are some drawbacks associated with each method. The Shallow Fusion method adds the output probability of the language model for the target domain to that of the existing ASR model, which is dependent on the language information contained in the training data used. This means that Shallow Fusion adds the output probabilities of two models trained with different language information. The Deep, Component, and Cold Fusion methods require retraining of the ASR each time a new language model is integrated, so these methods have not replaced the simple Shallow Fusion method as the go-to method among most of the ASR community, since Shallow Fusion does not require model retraining, as the language model is only applied during decoding.

In the days when Gaussian mixture model-hidden Markov model (GMM-HMM) and deep neural network-hidden Markov model (DNN-HMM) ASR models were primarily being used, it was less difficult to change the domain of an ASR model because the acoustic model, dictionary, and language model could each be easily replaced with target domain versions. However, since E2E ASR models are simultaneously trained with acoustic and language information, it is very difficult to completely separate the acoustic and language information contained within an E2E ASR model.

Given these backgrounds, the goal of this study is domain adaptation by separating acoustic and language information inside the E2E ASR model. Since these two pieces of information are trained at once in a single neural network model, it is impossible to strictly separate them. In this paper, we propose to approximate the separation of these two pieces of information by subtracting the "implicit" language model probability in the log-likelihood domain. We also conducted validation on a Japanese ASR task.

Some studies have parallely been conducted using very similar formulation, which is called density ratio approach (DRA) [32] and reported its effectiveness mainly on recurrent neural network (RNN)-Transducer in English, Spanish, and Italian ASR tasks [15, 32]. In contrast to these studies, we will construct Japanese encoder-decoder ASR models and apply the language model replacement to the models [33, 34], and make an analysis of the behavior, for the first time. We believe that integrating external language models with Japanese ASR models may be more difficult than the task of dealing with these alphabetic language ASR models. The reason for this is the size of the Japanese vocabulary. The Japanese vocabulary is huge, with thousands of unique tokens (characters) used in common speech, whereas the alphabetic languages such as English, which has only 26 letters. Even using subwords, the number of tokens are at most one or two thousands. The grammer is much more strict than Japanese, so the prediction by the language model tends to be easier than Japanese. Japanese is also known to have more homonyms and pronunciation variations

than alphabetic languages. There are three types of characters in Japanese: kanji, hiragana, and katakana. In addition, Japanese has other complex and difficult prefixes and suffixes. Furthermore, Japanese has a grammar with very loose grammatical constraints, especially in spontaneous speech. In other words, Japanese does not conform to the Western SVO (subject + verb + object) grammar. It has already been reported that the linguistic information inside English, Spanish, and Italian speech recognition models can be approximated by external language models. However, it is unclear whether the linguistic information learned in character-based Japanese ASR model can be approximated by a language model.

2 Related work

In this section, we first explain the language models used in automatic speech recognition systems; then, we introduce some methods used to integrate these language models and ASR models.

2.1 Language models

The language models used in ASR systems are probabilistic models that assign probabilities to sequences of letters and words. In other words, they are models that predict the likelihood of the occurrence of particular words or characters by inferring which words or characters are more natural in a given context. Typical language models include N-grams, which are statistical language models that use the probabilities of word chains containing N number of words, and RNN language models, which use recurrent neural networks capable of learning the properties of time-series data. In our experiments, we used an RNN language model given by the probabilities $P(y_l|y_0, ..., y_{l-1})$.

2.2 Shallow Fusion

Many studies have been conducted on methods of integrating ASR and language models, and the most common method currently used is called Shallow Fusion [28]. A conceptual diagram of Shallow Fusion is shown in Fig. 1. During Shallow Fusion, the output probabilities of an ASR model and a language model are added together in a logarithmic domain. When ASR model output y represents a sequence $y_1, y_2, \cdots, y_L$, where y_l is a symbol output at each time frame and L is the length of the output sequence, and when the input for the decoder ($output_{encoder}$ in Fig. 1) is expressed as x, the resulting output sequece $\hat{y}$ is expressed as shown in Eq. (1):

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

where $\log P_{\mathrm{ASR}}(y|x)$ is the output probability of the ASR model, which is the probability of inferring symbol label sequence y when acoustic feature sequence output x from the encoder is given. The expression $\log P_{\mathrm{LM}}(y)$ in Fig. 1 represents the prior of y given by the language model, and λ is a weighting parameter to balance the output probabilities of ASR model and language model to maximize prediction performance, which is determined through the trials using data other than the test data. When using the Shallow Fusion method, the language model is only used during inference, and the language model and ASR model are trained independently.

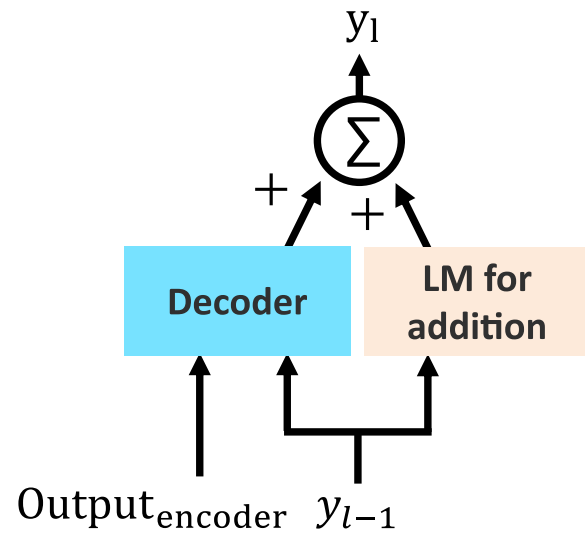

Fig. 1 Shallow Fusion

2.3 Deep Fusion

Deep Fusion [11] can be expressed as shown in Eq. (2) :

$$\begin{aligned} g_t &= \sigma\left(v^\top s_t^{\mathrm{LM}} + b\right) \\ s_t^{\mathrm{DF}} &= \left[s_t; g_t s_t^{\mathrm{LM}}\right] \\ y_t &= \mathrm{Softmax}\left(\mathrm{DNN}\left(\mathrm{s}_\mathrm{t}^{\mathrm{DF}}\right)\right), \end{aligned} \tag{2}$$

where $[s_t; g_t s_t^{\mathrm{LM}}]$ is the concatenation of vectors s_t, and where vectors 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. 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 then integrated by training a DNN, into which the hidden state information of the two pre-trained models are fed.

2.4 Cold Fusion

A modified version of the Deep Fusion method, called Cold Fusion, has also been proposed [29]. The ASR model is trained using linguistic information from a pre-trained language model. The Cold Fusion method can be expressed as shown in Eq. (3):

$$\begin{aligned} h_t^{LM} &= \mathrm{DNN}\left(\mathrm{l}_\mathrm{t}^{\mathrm{LM}}\right) \\ g_t &= \sigma\left(W\left[s_t; h_t^{\mathrm{LM}}\right] + b\right) \\ s_t^{\mathrm{CF}} &= \left[s_t; g_t \circ h_t^{\mathrm{LM}}\right] \\ y_t &= \mathrm{Softmax}(\mathrm{DNN}(\mathrm{s}_\mathrm{t}^{\mathrm{CF}})). \end{aligned} \tag{3}$$

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

As mentioned previously, several other methods for integrating the ASR model and the language model have also been proposed, and these integration methods have also been reported to improve speech recognition performance. However, some of these methods are not theoretically correct and some require additional DNN training using in-domain speech data. As will be explained later, our method does not require retraining of the whole ASR models using any additional speech data. Only the language models need to be trained. Of the integration methods discussed in this section, only Shallow Fusion shares this characteristic, while the others require retraining of the ASR model retraining. Furthermore, the formulation of our proposed method is similar to that of Shallow Fusion but is theoretically different. To compare the performance of our proposed method with that of the Shallow Fusion method, we performed an experiment, which is described in Section 4 of this paper.

3 Adaptation of LM in end-to-end ASR model using language model replacement

We propose a method of adapting the language model in a conventional, pre-trained E2E ASR model in order to improve recognition of target domain speech. This is achieved by first estimating the "implicit language information" contained inside the pre-trained ASR model. During the inference stage, this estimated language information is used to eliminate the prior language information learned from the source domain data that was used to train the original ASR model. Then, language information from the target domain, obtained from the LM of an independently trained ASR model, is combined with the language information in the adapted ASR model using a method similar to Shallow Fusion. A diagram of the proposed method is shown in Fig. 2.

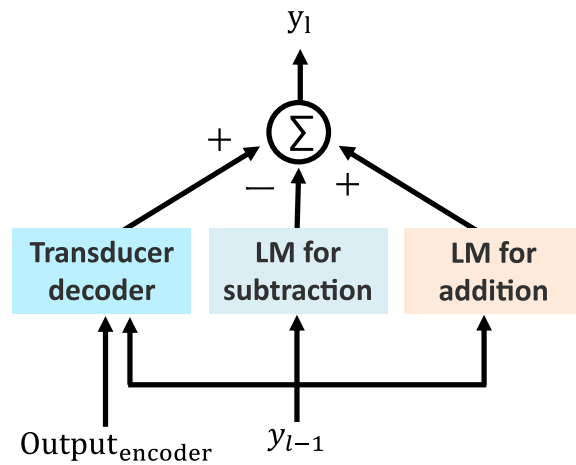

Fig. 2 Language model replacement

When a language model integration method is not used, the ASR model infers the sequence $\hat{y}$ as follows:

$$\hat{y} = \underset{y}{\mathrm{argmax}}\{\log P_{\mathrm{source}}(y|x)\}, \tag{4}$$

where $\log P_{\mathrm{source}}(y|x)$ is the log probability of output sequence y obtained from the source domain ASR model when the input sequence is x. Here, "source domain" refers to the task or activity during which the speech data used for training the ASR model was recorded, and x and y represent the input acoustic feature sequence and the output symbol label sequence, respectively. Log output probability $\log P_{\mathrm{source}}(y|x)$ from the ASR model can be expanded using Bayes' rule as follows:

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

On the right side of Eq. (5), we can see that the ASR model includes the acoustic information term $\log P_{\mathrm{source}}(x|y)$ and the language information term $\log P_{\mathrm{source}}(y)$, the latter of which is the "implicit" language information, contained in the ASR model, as described above, i.e., the statistics of the language contained in the source domain speech data used for training the ASR model. Shallow Fusion methods do not take this "implicit language information" into account. However, ASR models do take advantage of this "implicit language

information" to improve decoding when the domain of the test data is the same as the domain of the training data. But if the test domain is quite different from the training domain, this information can cause degradation of the ASR model's decoding performance.

Our method attempts to remove the "implicit language information" from the pre-trained ASR model using Bayes' rule. Assuming that the "implicit language information" from the source domain contained within the trained ASR model can be approximated by an external language model trained using text data from the same source domain, this source domain language information can be removed by subtracting the output probability of the external language model from the output probability of the ASR model for the source domain as follows:

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

where $\log \tilde{P}_{\text{source}}(y)$ is the probability of the external language model for the source domain, and λ_{sub} is a subtraction weight for balancing the acoustic and language information, which compensates for the estimation error of $P_{\text{source}}(y)$, that is, the difference between $P_{\text{source}}(y)$ and $\tilde{P}_{\text{source}}(y)$. Equation (6) can also be thought of as an estimation of the log output probability of a purely acoustic model.

The language information can then be replaced by adding the log output probabilities of the external language model trained for the target domain to Eq. (6), as follows:

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

where $\log \tilde{P}_{\text{target}}(y)$ is the probability of the external language model for the target domain, and λ_{add} is an addition weight. Here, $P_{(\text{source, target})}(y|x)$ represents a model with acoustic information from the source domain and language information from the target domain. To maximize recognition performance, λ_{sub} and λ_{add} are estimated using a grid search of a target domain dataset which is different from the test data. When the estimation of the implicit language infromation is accurate, the value of λ_{sub} is expected to be around 1.0. By using Eqs. (4) to (7), our method successfully replaces only the source domain language information within the ASR model. Therefore, an ASR model can be created for any target domain simply by preparing text data for that domain. Furthermore, this method integrates the external language model and ASR model at the inference stage as in Shallow Fusion, so only the language models used for subtraction and addition need to be trained, and the ASR model does not need retraining. Unlike Shallow Fusion, in which only the external in-domain language model is added, our method subtract implicit source domain language information, then add in-domain language information based on Bayes' rule.

As described in Section 1, several previous studies have proposed a similar formulation in which an implicit source domain language model and a target domain language model are added to the ASR model at some ratio. For example, McDermott et al. conducted experiments using an RNN-T decoder as an ASR model [32] and demonstrated that their approach is effective for English, Spanish, and Italian.

In this paper, we port this methodology into encoder-decoder ASR models and perform experiments using Japanese-language ASR tasks, which are more difficult than alphabetic language tasks due to the language's huge written vocabulary.

4 Experiments

4.1 Datasets used in experiments

This study required the use of datasets from multiple domains in order to validate the proposed language model replacement method for adapting ASR models to a target domain, which is achieved by replacing the "implicit language information" contained inside an existing ASR model with target domain information. We used three Japanese-language datasets in our experiments: the Corpus of Spontaneous Japanese (CSJ) [19], the Japanese Newspaper Article Speech (JNAS) corpus [35], and the Mainichi Shimbun (MS) newspaper articles text dataset [36].

- *Corpus of Spontaneous Japanese*. The CSJ is a dataset that contains a large amount of spontaneous, modern Japanese speech, as well as additional information for research. It contains 7 million words and 660 h of audio files. The recorded speech consists mainly of public speaking speech but also includes dialogues and readings. The speech has no particular dialectal characteristics with regard to vocabulary or grammar. Only two domains from the CSJ dataset were used in this study, the Academic Presentation Speech (APS) dataset, which includes presentations on the subjects of engineering, the humanities, and sociology, and the Simulated Public Speech (SPS) dataset, which consist of speeches without academic terminology. The APS and SPS corpora consist of 275 and 321 hours of speech data, respectively.
- *Japanese Newspaper Article Speech*. The JNAS dataset consists of read speech and text data from daily newspaper articles, as well as phoneme-balanced ATR 503 sentences extracted from newspapers,

journals, novels, letters, text books, etc., read by 306 speakers (153 males and females each). The ATR503 sentences consist of 9 sets of 50 sentences each in sets A to I, as well as a 10th J set which contains 53 sentences.

- *Mainichi Shimbun*. The MS dataset is a text dataset consisting of 58,944,516 characters from 11 years of daily newspaper articles (1991 to 2002). Since the JNAS corpus described above is a reading of sentences extracted from Mainichi Shimbun articles, the JNAS and MS datasets are equivalent domains in terms of language information.

Table 1 shows the details of the datasets used in our experiment. The CSJ APS corpus and CSJ SPS corpus were both randomly split into a training set, "dev1 set," "dev2 set," and test set, at a ratio of 9.0:0.5:0.25:0.25, respectively. The training and "dev1" sets were used as training and development data, respectively, when training the ASR models, while the "dev2" set was used for tuning language model weights, and the test set was used for performance evaluation. The number of unique Japanese characters contained in the CSJ data used in this experiment was 3262. The MS dataset is a text dataset consisting of a total of 58,944,516 non-unique characters from daily newspaper articles and was only used to train the external "target domain" language model. The JNAS dataset was created by having speakers read aloud excerpts from Mainichi Shimbun newspaper articles; thus, the JNAS speech data is from the same domain as the MS text data. JNAS dataset was split into a "dev" set and test set. The "dev" set was used as data for language model tuning and the test set as evaluation data for our second experiment.

4.2 ESPnet

The End-to-End Speech Processing Toolkit, ESPnet, is an open-source speech processing toolkit specialized for end-to-end models [37, 38], which contains several types of ASR models. We used the RNN (Hybrid CTC/Attention Architecture) and Transformer (Joint CTC Attention Transformer) models in our experiments.

4.2.1 Hybrid CTC/Attention Architecture

Figure 3 shows a diagram of a Hybrid CTC/Attention Architecture ASR model. First, the input acoustic features are formatted using VGGnet [39]; then, they are converted into intermediate representation H by six BLSTM (Bidirectional LSTM) layers which are used as the encoder. The decoder consists of one LSTM layer and one Linear layer. An additional Linear layer is used as the CTC decoder.

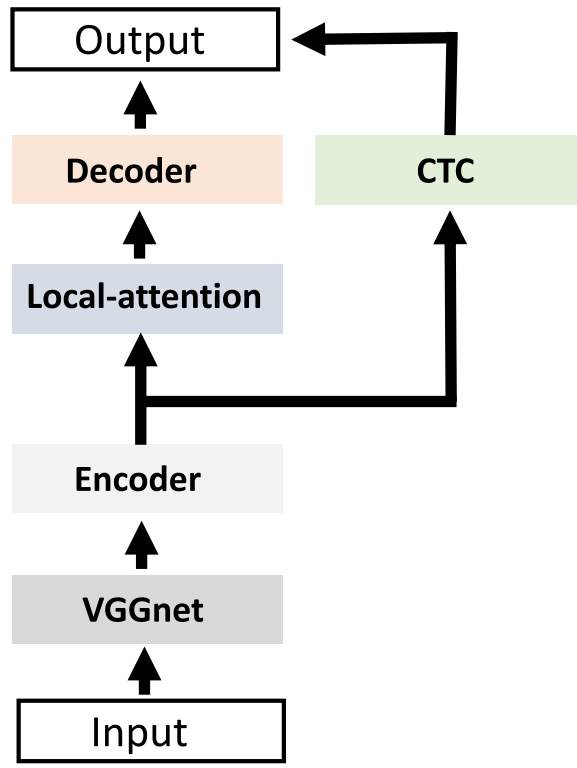

Fig. 3 Hybrid CTC/Attention Architecture

Table 1 Details of datasets used in experiment

Domain	Split	Speakers	Utterances	Characters	Duration
APS (CSJ)	Train	889	139,396	5,041,328	253 h
	Dev1	49	5,272	195,416	10 h
	Dev2	25	3,001	106,293	6 h
	Test	25	3,163	117,252	6h
SPS (CSJ)	Train	1,555	221,994	5,503,402	303 h
	Dev1	80	8,612	258,651	14 h
	Dev2	40	4,278	124,528	7 h
	Test	40	4,740	121,465	7 h
MS	Train	—	—	58,944,516	—
JNAS	Dev	23	500	230,821	0.9 h
	Test	23	500	238,717	0.9 h

4.2.2 Joint CTC Attention Transformer

Figure 4 shows a diagram of a Joint CTC Attention Transformer ASR 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 employs a residual connection around each of the two sub-layers, followed by

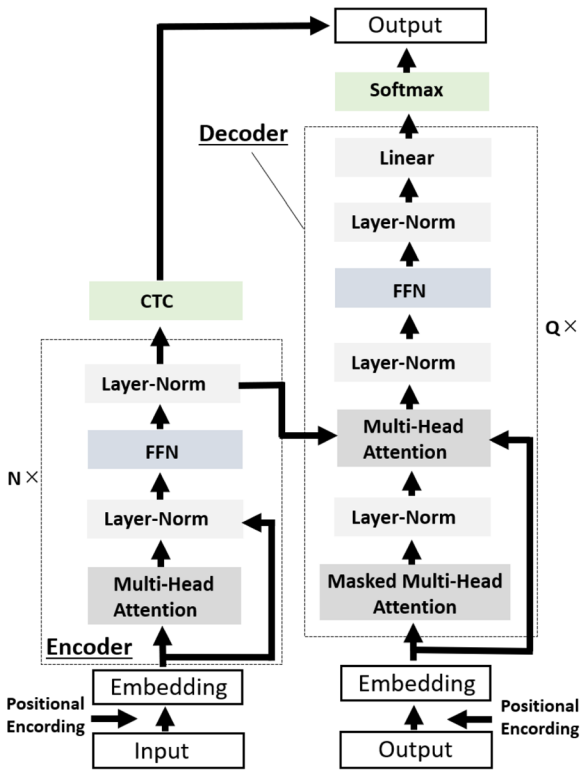

Fig. 4 Joint CTC Attention Transformer

layer normalization. Thus, the output of each sublayer is the Layer-Norm.

The decoder consists of a stack of $Q = 6$ identical layers. In addition to the two sub-layers used in each encoder layer, the decoder contains three additional sub-layers to perform multi-head attention on the output of the encoder stack. Similar to the encoder, it uses a residual connection around each sublayer, followed by layer normalization. It also modifies the self-attention sublayer of the decoder stack so that positions do not join subsequent positions. Further, in order to supplement the input acoustic information, a CTC composed of one Linear layer is used at the top of the encoder.

4.2.3 Implementation of language model replacement to ESPnet models

Figure 5 shows a diagram of the proposed language model replacement (LMR) method when applied to an ESPnet2 encoder-decoder ASR model using Transformer as encoder-decoder, where language model replacement is applied to the decoder[1]. The decoder output of the

[1] Encoder-decoder models from ESPnet2 use a CTC decoder to align output in monotonic order. Strictly speaking, the CTC decoder output must also be compensated for during language model replacement. Theoretically, the implicit language information in the CTC decoder can be simulated using unigram [40]; however, context-dependency has been observed; thus, language model replacement is difficult. This remains a task for future work.

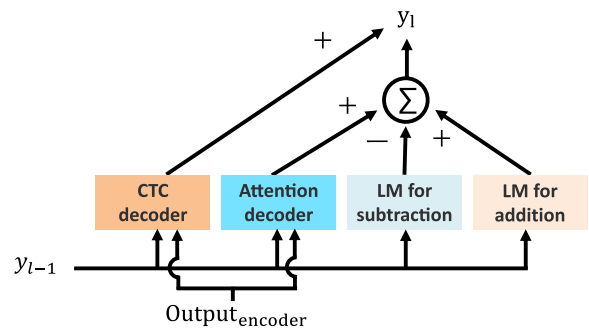

Fig. 5 Language model replacement for ESPnet2 encoder-decoder ASR model

RNN model is constructed using an RNN layer such as an LSTM, which attempts to predict the current state of an utterance from its past state. The decoder of the Transformer ASR model is composed of an attention layer and a feed-forward layer, but it performs masking and is also trained to predict the current state from the past state, i.e., the decoder has been trained to include linguistic 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 ASR model trained in the source domain. The subtraction of this language model is intended to remove the language information contained in the decoder. Then, the language information in the language model can be tuned to the target domain by adding the output probabilities of the language model for the target domain. This method is also effective for ASR models which use a beam search, such as the Conformer model.

4.3 ASR and language models

In this section, we describe the six ASR models and four language models used in our experiments.

- *ASR models*. A Hybrid CTC/Attention Architecture ASR model and a Joint CTC Attention Transformer ASR model were each trained using either APS corpus, the SPS corpus, or both the APS and SPS corpora. We used ESPnet's Hybrid CTC/Attention Architecture and Joint CTC Attention Transformer models, as described in Section 4.2.1. All of the models were set up using ESPnet's CSJ recipe.
- *Language models*. Four LSTM language models were trained using either APS text data, SPS text data, APS and SPS text data, or Mainichi Shimbun text data. We used the LSTM language model provided by ESPnet, which consists of an Embedding layer, two LSTM layers, and a Linear layer, for all four of the language models.

Table 2 Perplexities for each language model and test set

Language model	Perplexity for test set			
(RNN)	APS	SPS	APS+SPS	JNAS
APS	17.33	33.37	24.23	147.66
SPS	34.79	17.38	24.39	100.64
APS+SPS	15.78	16.77	16.28	80.53
MS	95.82	56.58	73.18	33.43

4.4 Differences among language domains

We calculated test set perplexities for the language models to analyze the differences among the language domains of the datasets, using four RNN language models. Perplexity P of a probability distribution p_{LM}, describing a language model which output a probability of y_l when given history y_1, is shown in Eq. (8):

$$P = \exp\left(-\frac{1}{L}\sum_{l=1}^{L}\log p_{LM}\left(y_l|y_1,\cdots,y_{l-1}\right)\right), \quad (8)$$

where y_l indicates the l-th character in the test set reference transcription and where the sequence $y_l, (l = 1,\cdots,L)$ is a transcription of the test set. Table 2 shows the perplexities of the test sets for each model.

When the APS language model is evaluated with the APS test set, the perplexity is 17.33. On the other hand, when the APS language model is evaluated with the SPS test set, the perplexity is 33.37. Almost the same result was obtained when evaluating the SPS language model with the SPS and APS test sets. This indicates that the data in the APS and SPS datasets are from linguistically different domains (academic and non-academoic presentation speech, respectively).

When evaluating the language models trained using APS+SPS with the APS+SPS test set and JNAS test set, we obtained perplexities of 16.28 and 80.53, respectively, demonstrating that the difference between the domains of the APS+SPS dataset and the JNAS dataset is very large. This is because the APS and SPS are corpora of spontaneous speech, whereas the JNAS dataset is a read speech corpus of newspaper articles. As a result, the language model trained with Mainichi Simbun data obtained a much lower perplexity when processing the JNAS test set.

4.5 Experiment 1: ASR tasks involving different language domains

We compared ASR performance when the source and target domain language models were not integrated (baseline method), when a conventional, Shallow Fusion language model adaptation method was used, and when our proposed language model replacement (LMR) method was used, using each of the six encoder-decoder E2E ASR models and the four language models described in Section 4.3. We evaluated each model's performance using its character error rate (CER) when processing the test set. Our ASR system uses Japanese characters as units for recognition. We counted the number of substitutions (S), deletions (D), insertions (I), and correct characters (C) when calculating the CER. When N represents the number of characters in the reference, and $N = S + D + C$, CER is calculated as described as below:

$$CER = \frac{S+D+I}{N}. \quad (9)$$

The APS and SPS test sets were used, for cross-domain evaluation, respectively. We used the dev2 sets to tune addition weight λ_{add} of the Shallow Fusion equation and subtraction and addition weights λ_{sub} and λ_{add} of the LMR equation. The language model weights were optimized using a greedy search in the range of 0.1 to 1.1 in increments of 0.2. Experimental results for the Hybrid CTC/Attention Architecture and Joint CTC Attention Transformer ASR models are shown in Tables 3 and 4, respectively. For reference, we also show the CERs of the APS ASR model when using the APS test set, and the SPS ASR model when using the SPS test set, i.e., the result under matched domain conditions.

As shown in Table 3, when we used the APS-trained Hybrid CTC/Attention Architecture ASR model with the SPS test set, our proposed LMR method achieved a CER of 13.4%, outperforming the Shallow Fusion integration method which achieved a CER of 14.9%. When we use the SPS-trained Hybrid CTC/Attention Architecture ASR model with the APS test set, the LMR method achieved a CER of 16.7%, while the Shallow Fusion method achieved a CER of 18.5%, a relative reduction in error of 9.7%. As shown in Table 4, when using the APS-trained Joint CTC Attention Transformer ASR model and the SPS test set, the LMR integration method obtained a CER of 9.8%, better performance than the Shallow Fusion integration method, which achieved a CER of 10.8%. When using the SPS-trained Joint CTC Attention Transformer ASR model with the APS test set, the LMR integration method achieved a CER of 12.7%, while the CER when using Shallow Fusion was 14.3%. Thus, our proposed LMR integration method achieved better recognition results than Shallow Fusion in encoder-decoder E2E ASR models when performing cross domain recognition tasks in Japanese.

We then optimized the language model weights using the dev2 datasets. CERs when using various weights are shown in Tables 5 and 6. The horizontal axis values represent language model subtraction weights, while the vertical axis values represent addition weights. Vertical

Table 3 Speech recognition results (CER (%)) for Hybrid CTC/Attention Architecture model for baseline (B/L), Shallow Fusion (SF), and language model replacement (LMR) methods for ASR tasks involving different language domains

Method	Model	Test set	Language model integration details	λ_{sub}	λ_{add}	CER (%)
B/L	APS	SPS	—	—	—	15.7
SF			$ASR_{APS} + \lambda_{add}\, LM_{SPS}$	—	0.3	14.9
LMR			$ASR_{APS} - \lambda_{sub}\, LM_{APS} + \lambda_{add}\, LM_{SPS}$	0.7	0.5	**13.4**
B/L	SPS	APS	—	—	—	19.8
SF			$ASR_{SPS} + \lambda_{add}\, LM_{APS}$	—	0.3	18.5
LMR			$ASR_{SPS} - \lambda_{sub}\, LM_{SPS} + \lambda_{add}\, LM_{APS}$	0.9	0.7	**16.7**
Matched	APS	APS	—	—	—	9.2
	SPS	SPS	—	—	—	7.6

Table 4 Speech recognition results (CER (%)) for Joint CTC Attention Transformer model for baseline (B/L), Shallow Fusion (SF), and language model replacement (LMR) methods for ASR tasks involving different language domains

Method	Model	Test set	Language model integration details	λ_{sub}	λ_{add}	CER (%)
B/L	APS	SPS	—	—	—	11.3
SF			$ASR_{APS} + \lambda_{add}\, LM_{SPS}$	—	0.3	10.8
LMR			$ASR_{APS} - \lambda_{sub}\, LM_{APS} + \lambda_{add}\, LM_{SPS}$	0.5	0.7	**9.8**
B/L	SPS	APS	—	—	—	15.2
SF			$ASR_{SPS} + \lambda_{add}\, LM_{APS}$	—	0.3	14.3
LMR			$ASR_{SPS} - \lambda_{sub}\, LM_{SPS} + \lambda_{add}\, LM_{APS}$	0.5	0.7	**12.7**
Matched	APS	APS	—	—	—	6.8
	SPS	SPS	—	—	—	5.6

column $\lambda_{sub} = 0$ represents results when using the Shallow Fusion integration method. Overall, we can see that the optimal addition weights for LMR are larger than the optimal additional weights for Shallow Fusion. This suggests that LMR integration allows the ASR model to use language from the target domain more effectively than Shallow Fusion. In each experiment, the optimal subtraction and addition weights for the dev2 set are almost the same as those for the test set when either LMR or Shallow Fusion were applied; thus, optimization of these weights is stable.

In general, when using Shallow Fusion, CER increases after the the language model addition weight exceeds a certain value, while for LMR, this increase is observed for the both the addition and subtraction weights. In other words, providing excessive or insufficient linguistic information when using LMR both leads to a decrease in ASR performance.

4.6 Experiment 2: ASR tasks involving domains with different speaking styles

We also investigated ASR performance when integrating source and target doamains with different speaking styles. The ASR model trained with the APS+SPS dataset, which combines data from two spontaneous speech corpora, was adapted to the JNAS read newspaper article speech domain using the language model trained with the MS dataset. The weights for the language model were tuned using the JNAS dev set and optimized using a greedy search with a range of 0.1 to 1.1 and a step size of 0.2. This range for the subtraction weight is used because if the estimation of the implicit language information is accurate, the search result value is expected around 1.0, but if estimation is inaccurate, the subtracted data might be 'noise' and thus the search result value becomes low. As for the addition weight, the search result value depends on the degree of correlation with the test data. Thus, if the language model information matches the test data, the search result value is expected to be around 1.0 and if it is not well matched, the search result value will be low. ASR performance was evaluated using CER when processing the JNAS test set. Our experimental results are shown in Tables 7 and 8. CERs when using the LMR and Shallow Fusion adaptation methods with the Hybrid CTC/Attention Architecture ASR model were 15.9% and 16.6%, respectively, a reduction in relative error of 4.2% when using LMR method. When using the Joint CTC Attention Transformer ASR model, the LMR

Table 5 Hybrid CTC/Attention Architecture model results (CER) for ASR tasks involving different language domains when using various language model addition and subtraction weights, for Shallow Fusion ($\lambda_{sub} = 0.0$) and for the language model replacement method

$\lambda_{add} \backslash \lambda_{sub}$	**0.0**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**1.1**
CERs for SPS ASR model + APS language model for APS dev2 set							
0.1	17.4	17.3	17.9	20.5	27.6	11.1	271.7
0.3	16.5	16.3	16.2	17.0	19.8	28.8	172.6
0.5	16.5	15.9	15.2	15.3	16.6	20.3	33.4
0.7	17.3	16.4	15.0	14.6	15.1	16.9	22.2
0.9	19.0	17.6	15.5	14.5	**14.4**	15.3	18.2
1.1	21.9	19.7	16.8	15.1	**14.4**	14.7	16.5
CERs for SPS ASR model + APS language model for APS test set							
0.1	19.1	19.0	19.8	22.3	29.5	114.5	250.4
0.3	18.5	18.2	18.0	19.0	21.8	31.1	174..7
0.5	18.6	18.0	17.4	17.4	18.6	22.5	35.8
0.7	19.6	18.6	17.4	16.8	17.1	19.0	24.4
0.9	21.3	20.0	18.0	17.0	**16.7**	17.5	20.7
1.1	24.1	22.1	19.3	17.7	16.9	17.1	18.8
CERs for APS ASR model + SPS language model for SPS dev2 set							
0.1	14.6	14.5	15.1	17.2	23.7	72.6	296.5
0.3	14.2	13.8	13.6	14.5	17.0	25.2	120.2
0.5	14.4	13.9	13.1	13.2	14.4	17.7	29.2
0.7	15.4	14.5	13.4	**12.9**	13.3	14.9	19.7
0.9	16.9	15.8	14.1	13.2	13.1	13.7	16.2
1.1	19.2	17.6	15.4	14.0	13.3	13.5	14.8
CERs for APS ASR model + SPS language model for SPS test set							
0.1	15.3	15.2	15.8	18.0	24.5	75.6	302.8
0.3	14.9	14.5	14.2	15.1	17.8	26.4	127.5
0.5	15.1	14.6	13.7	13.8	14.9	18.5	30.8
0.7	16.1	15.2	14.0	**13.4**	13.7	15.5	20.6
0.9	17.5	16.3	14.8	13.8	**13.4**	14.1	16.9
1.1	19.5	18.1	15.9	14.6	13.8	13.9	15.4

integration method achieved areduction of 0.5% in absolute CER compared to the Shallow Fusion method. Thus, we were able to confirm that the proposed LMR adaptation method was more effective for ASR tasks involving domains with different speaking styles, even when acoustic adaptation was not applied.

ASR model performance for experiment 2 when using various language model addition and subtraction weights are shown in Tables 9 and 10. Compared to the result of domain adaptation experiment 1, using only the APS and SPS corpora, the optimal values for the subtraction and addition weights were smaller in experiment 2. As shown in Table 2, we observed that JNAS test set perplexity for the MS language model is 33.43, while the matched perplexities for the APS and SPS language models are both about 17; therefore, the linguistic constraints of the MS language model on the JNAS corpus are relatively small, which might be the reason for the smaller optimal language model weights in experiment 2.

5 Conclusion

In this paper, we have proposed a method for replacing "implicit" source domain language information contained within the language model of an ASR model with language information from a target domain language model, in order to efficiently adapt pre-trained E2E ASR models to a target domain. This method is based on the Bayes' rule and does not require re-training of the ASR model for adaptation.

Our first language model adaptation experiment, conducted using encoder-decoder models trained with the APS and SPS corpora of the Corpus of Spontaneous Japanese, showed that the proposed "language model replacement" (LMR) method achieved better ASR performance than the conventional Shallow Fusion method

Table 6 Joint CTC Attention Transformer model results (CER) for ASR tasks involving different language domains when using various language model addition and subtraction weights, for Shallow Fusion ($\lambda_{sub} = 0.0$) and for the language model replacement method

$\lambda_{add} \backslash \lambda_{sub}$	**0.0**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**1.1**
CERs for SPS ASR model + APS language model for APS dev2 set							
0.1	13.0	12.9	13.2	18.1	41.0	70.5	117.3
0.3	12.3	11.9	11.5	12.1	19.9	44.0	77.9
0.5	12.3	11.8	11.0	10.8	11.7	24.1	49.7
0.7	12.9	12.1	11.0	**10.4**	10.7	12.6	30.7
0.9	14.1	13.1	11.6	10.7	**10.4**	11.1	15.8
1.1	16.0	14.5	12.7	11.4	10.8	11.0	12.6
CERs for SPS ASR model + APS language model for APS test set							
0.1	14.6	14.5	14.9	19.6	42.6	71.5	116.2
0.3	14.3	13.8	13.4	14.0	21.7	45.9	78.2
0.5	14.5	13.9	13.0	12.9	13.9	26.3	51.5
0.7	15.3	14.5	13.3	**12.7**	12.9	14.9	32.9
0.9	16.8	15.6	14.0	13.0	**12.7**	13.5	18.3
1.1	18.8	17.4	15.2	13.9	13.2	13.3	15.0
CERs for APS ASR model + SPS language model for SPS dev2 set							
0.1	10.4	10.3	10.4	14.2	37.9	73.3	118.0
0.3	10.3	10.0	9.7	9.9	16.9	14.8	84.7
0.5	10.5	10.1	9.5	9.5	10.0	21.3	52.2
0.7	10.9	10.5	9.7	**9.4**	9.5	10.8	28,1
0.9	11.8	11.2	10.2	9.7	9.5	10.0	13.4
1.1	13.4	12.4	11.0	10.3	9.9	10.0	11.1
CERs for APS ASR model + SPS language model for SPS test set							
0.1	11.0	10.9	10.9	14.9	39.4	78.8	124.5
0.3	10.8	10.5	10.2	10.5	17.6	45.3	92.7
0.5	11.1	10.6	10.0	9.9	10.5	22.0	55.6
0.7	11.6	11.1	10.3	**9.8**	9.9	11.2	29.4
0.9	12.5	11.8	10.8	10.1	10.0	10.4	14.0
1.1	13.9	13.0	11.6	10.8	10.3	10.4	11.6

Table 7 Hybrid CTC/Attention Architecture model results (CER) for baseline (B/L), Shallow Fusion (SF), and language model replacement (LMR) methods for language models with different speaking styles

Method	**Model**	**Test set**	**Language model integration details**	λ_{sub}	λ_{add}	**CER**
B/L	APS+SPS	JNAS	—	—	—	18.9
SF			$ASR_{APS+SPS} + \lambda_{add}\, LM_{MS}$	—	0.3	16.6
LMR			$ASR_{APS+SPS} - \lambda_{sub}\, LM_{APS+SPS} + \lambda_{add}\, LM_{MS}$	0.3	0.5	**15.9**

Table 8 Joint CTC Attention Transformer model results (CER) for baseline (B/L), Shallow Fusion (SF), and language model replacement (LMR)methods for language models with different speaking styles

Method	**Model**	**Test set**	**Language model integration details**	λ_{sub}	λ_{add}	**CER**
B/L	APS+SPS	JNAS	—	—	—	12.8
SF			$ASR_{APS+SPS} + \lambda_{add}\, LM_{MS}$	—	0.5	11.0
LMR			$ASR_{APS+SPS} - \lambda_{sub}\, LM_{APS+SPS} + \lambda_{add}\, LM_{MS}$	0.3	0.7	**10.5**

Table 9 Hybrid CTC/Attention Architecture model results (CER) for ASR tasks involving different speaking styles when using various language model addition and subtraction weights, for Shallow Fusion ($\lambda_{sub} = 0.0$) and language model replacement

$\lambda_{add} \backslash \lambda_{sub}$	**0.0**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**1.1**
CERs for JNAS dev set							
0.1	16.6	16.7	17.5	20.7	33.4	484.1	416.2
0.3	15.3	15.0	15.3	16. 3	21.1	51.3	763.7
0.5	15.3	14.8	**14.5**	15.4	19.8	33.3	537.4
0.7	16.4	15.6	14.9	15.9	20.9	35.9	116.3
0.9	19.1	17.3	16.2	18.0	24.2	40.4	80.1
1.1	23.3	19.9	19.2	21.6	29.6	46.6	85.1
CERs for JNAS test set							
0.1	17.7	17.8	18.9	22.2	37.0	498.0	417.4
0.3	16.6	16.2	16.2	18.0	23.4	57.6	777.7
0.5	16.6	16.2	**15.9**	17.0	21.9	36.2	591.7
0.7	20.5	17.2	16.7	17.6	23.0	38.5	129.7
0.9	20.1	18.6	20.9	19.6	26.7	42.3	87.5
1.1	24.9	22.1	20.6	22.5	31.7	48.1	93.7

Table 10 Joint CTC attention Transformer model results (CER) for ASR tasks involving different speaking styles when using various language model addition and subtraction weights, for Shallow Fusion ($\lambda_{sub} = 0.0$) and language model replacement

$\lambda_{add} \backslash \lambda_{sub}$	**0.0**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**1.1**
CERs for JNAS dev set							
0.1	11.8	11.8	11.9	13.2	28.2	55.5	93.6
0.3	10.8	10.8	10.8	11.0	13.3	32.2	73.1
0.5	10.5	10.4	10.2	10.4	11.4	20.0	52.7
0.7	10.6	10.3	**10.1**	**10.1**	11.4	18.9	40.6
0.9	11.1	10.5	10.3	10.4	12.9	21.1	36.8
1.1	12.5	11.2	10.7	11.5	15.5	23.8	37.6
CERs for JNAS test set							
0.1	12.1	12.1	12.5	15.1	35.3	74.8	116.9
0.3	11.1	11.1	11.1	11.4	16.9	44.6	102.4
0.5	11.0	10.7	**10.5**	10.7	13.2	30.3	79.7
0.7	11.2	10.9	10.6	10.9	14.9	27.0	60.1
0.9	11.5	11.3	10.9	11.9	17.2	28.7	51.0
1.1	12.5	12.1	12.0	14.0	21.3	32.5	50.5

when integrating language models for different domains. In a second experiment, language models trained using corpora with different speaking styles were integrated. A language model trained with spontaneous Japanese presentation speech and a language model trained with Japanese newspaper article read speech were integrated. Our proposed LMR method also outperformed Shallow Fusion in this experiment.

Finally, our analysis of the magnitude of the language model weights used to add linguistic information implied that the proposed "language model replacement" method made better use of the target domain language information than the Shallow Fusion integration method, based on ASR performance in terms of CERs.

Abbreviations

ASR	Automatic speech recognition
E2E	End-to-end
HMM	Hidden Markov model
GMM	Gaussian mixture model
DNN	Deep neural network
LSTM	Long short-term memory
BLSTM	Bi-directional long short-term memory
CTC	Connectionist temporal classification
DRA	Density ratio approach
RNN	Recurrent neural network
LM	Language model
NLP	Natural language processing
CSJ	Corpus of spontaneous Japanese

APS	Academic presentation speech
SPS	Simulated public speech
JNAS	Japanese newspaper article speech
MS	Mainichi shimbun
LMR	Language model replacement
CER	Character error rate

Acknowledgements
This work was partially supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers 21K13641, 22H04863, and 23H00493.

Authors' contributions
DM proposed the methodology, conducted the experiments, and wrote the manuscript. KO, RN, and AO supervised the design of the experiments and refined the manuscript. NK supervised the entire research project and refined the manuscript. All authors read and approved the final manuscript.

Availability of data and materials
The CSJ and JNAS corpora used in the experiments for this paper are publicly available for nominal fees. Access is provided after the submission and review of a "Pledge of Use" to the Speech Resources Consortium. The Mainichi Shimbun database is also available for a small fee because it is the property of the Mainichi Newspapers Company.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 23 January 2024 Accepted: 17 June 2024
Published online: 20 July 2024

References

1. S. Mirsamadi, J.H.L. Hansen, in *iInterspeech 2015,*, A study on deep neural network acoustic model adaptation for robust far-field speech recognition (2015), pp. 2430–2435
2. K. Yao, D. Yu, F. Seide, H. Su, L. Deng, Y. Gong, in *2012 IEEE Spoken Language Technology Workshop (SLT)*, Adaptation of context-dependent deep neural networks for automatic speech recognition (IEEE, 2012), pp. 366-369
3. I. Sutskever, O. Vinyals, Q.V. Le, in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, Sequence to sequence learning with neural networks (NeurIPS Foundation, 2014), pp. 3104–3112
4. M.-T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation (2015). arXiv preprint arXiv:1508.04025
5. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, in *Proceedings of the 31st International Coference o NeuralInformation Processing Systms (NIPS)*, Attention is all you need. (NeurIPS Foundtio,2017), pp. 6000–6010
6. A. Zeyer, P. Bahar, K. Irie, R. Schlüter, H. Ney, in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, A comparison of Transformer and LSTM encoder-decoder models for ASR (IEEE, 2019), pp. 8-15
7. J.-X. Zhang, Z.-H. Ling, L.-R. Dai, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Forward attention in sequence-to-sequence acoustic modeling for speech synthesis (IEEE, 2018), pp. 4789-4793
8. S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N.E.Y. Soplin, R. Yamamoto, X. Wang, et al., in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, A comparative study on Transformer vs RNN in speech applications (IEEE, 2019), pp. 449-456
9. J.K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, in *Intentinal Confernce o Neural Informatin Processing Systms (NIPS)*, Attention-based models for speech recognition. (2015), pp. 577–585
10. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate (2014). arXiv preprint arXiv:1409.0473
11. C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, Y. Bengio, On using monolingual corpora in neural machine translation (2015). arXiv preprint arXiv:1503.03535
12. A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, in *Proceedings of the 23rd International Conference on Machine Learning*, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks (IMLS, 2006), pp. 369–376
13. A. Graves, Sequence transduction with recurrent neural networks (2012). arXiv preprint arXiv:1211.3711
14. A. Graves, A.-r. Mohamed, G. Hinton, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Speech recognition with deep recurrent neural networks (IEEE, 2013), pp. 6645-6649
15. G. Saon, Z. Tüske, D. Bolanos, B. Kingsbury, in *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Advancing RNN Transducer technology for speech recognition (IEEE, 2021), pp. 5654–5658
16. Y. He, T.N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K.C. Sim, T. Bagby, S.-y. Chang, K. Rao, A. Gruenstein, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Streaming end-to-end speech recognition for mobile devices (IEEE, 2019), pp. 6381–6385
17. I. Lopatovska, K. Rink, I. Knight, K. Raines, K. Cosenza, H. Williams, P. Sorsche, D. Hirsch, Q. Li, A. Martinez, Talk to me: Exploring user interactions with the Amazon Alexa. J. Librariansh. Inf. Sci. **51**(4), 984–997 (2019)
18. A. Kaplan, M. Haenlein, Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Bus. Horiz. **62**(1), 15–25 (2019)
19. K, Maekawa, in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, Corpus of Spontaneous Japanese: Its design and evaluation (ISCA & IEEE, 2003)
20. S. Ando, H. Fujihara, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Construction of a large-scale Japanese ASR corpus on TV recordings (IEEE, 2021), pp. 6948-6952
21. H.-J. Chang, H.-y. Lee, L.-s. Lee, Towards lifelong learning of end-to-end ASR (2021), arXiv preprint arXiv:2104.01616
22. B. Thomas, S. Kessler, S. Karout, in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Efficient adapter transfer of self-supervised speech models for automatic speech recognition (IEEE, 2022), pp. 7102-7106
23. T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, S. Khudanpur, in INTERSPEECH. Recurrent neural network based language model **2**, 1045–1048 (2010)
24. T. Mikolov, et al., PhD Thesis of Brno University of Technology, Statistical language models based on neural networks (Brno University of Technology, 2012).
25. Y. Bengio, R. Ducharme, P. Vincent, in *Advances in Neural Information Processing Systems*, ed. by T. Leen, T. Dietterich, V. Tresp. A neural probabilistic language model, vol. 13 (MIT Press, 2000)
26. A. Pauls, D. Klein, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Faster and smaller n-gram language models (ACL, 2011), pp. 258–267
27. T. Hori, S. Watanabe, Y. Zhang, W. Chan, in *INTERSPEECH 2017*, Advances in Joint CTC-Attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM (ISCA, 2017), pp. 949–953
28. A. Kannan, Y. Wu, P. Nguyen, T.N. Sainath, Z. Chen, R. Prabhavalkar, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, An analysis of incorporating an external language model into a sequence-to-sequence model (IEEE, 2018), pp. 5824–5828
29. A. Sriram, H. Jun, S. Satheesh, A. Coates, Cold Fusion: Training seq2seq models together with language models (2017). arXiv preprint arXiv:1708.06426
30. Z. Yang, B. Dhingra, Y. Yuan, J. Hu, W.W. Cohen, R. Salakhutdinov, Words or characters? Fine-grained gating for reading comprehension (2016). arXiv preprint arXiv:1611.01724
31. C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, L. Xie, in *ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Component Fusion: Learning replaceable language model

component for end-to-end speech recognition system (IEEE, 2019), pp. 5361-5635
32. E. McDermott, H. Sak, E. Variani, in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, A density ratio approach to language model fusion in end-to-end automatic speech recognition (IEEE, 2019), pp. 434-441
33. D. Mori, K. Ohta, R. Nishimura, A. Ogawa, N. Kitaoka, in *APSIPA ASC 2021*, Advanced language model fusion method for encoder-decoder model in Japanese speech (APSIPA, 2021), pp. 503-510
34. D. Mori, K. Ohta, R. Nishimura, N. Kitaoka, in *ICAICTA2022*, Implicit language information replace method in Japanese encoder-decoder ASR model (IEEE, 2022)
35. K. Itou, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, S. Itahashi, JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research. J. Acoust. Soc. Jpn. (E) **20**(3), 199–206 (1999)
36. The Mainichi Newspapers Co. Mainichi newspaper article database. http://mainichi.jp/contents/edu/maisaku/. Accessed 23 Jan 2023
37. S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.E.Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, T. Ochiai, in *Proc. Interspeech 2018*, ESPnet: End-to-End Speech Processing Toolkit (ISCA, 2018), pp. 2207–2211
38. S. Watanabe, F. Boyer, X. Chang, P. Guo, T. Hayashi, Y. Higuchi, T. Hori, W.-C. Huang, H. Inaguma, N. Kamo, et al., in *2021 IEEE Data Science and Learning Workshop (DSLW)*, The 2020 ESPnet update: New features, broadened applications, performance improvements, and future plans (IEEE, 2021), pp. 1–6
39. K. Simonyan, A. Zisserman, in *ICLR2015*, Very deep convolutional networks for large-scale image recognition (ICLR, 2015)
40. T. Takagi, N. Kitaoka, A. Ogawa, Y. Wakabayashi, in *APSIPA ASC 2023*, Streaming end-to-end ASR using CTC decoder and DRA for linguistic information substitution (APSiPA, 2023)

Publisher's Note