PAPER

Code-Switching ASR and TTS Using Semisupervised Learning with Machine Speech Chain

Sahoko NAKAYAMA[†,††a)], **Andros TJANDRA**[†*], ***Nonmembers***, **Sakriani SAKTI**[†,††], ***and*** **Satoshi NAKAMURA**[†,††], ***Members***

SUMMARY The phenomenon where a speaker mixes two or more languages within the same conversation is called code-switching (CS). Handling CS is challenging for automatic speech recognition (ASR) and text-to-speech (TTS) because it requires coping with multilingual input. Although CS text or speech may be found in social media, the datasets of CS speech and corresponding CS transcriptions are hard to obtain even though they are required for supervised training. This work adopts a deep learning-based machine speech chain to train CS ASR and CS TTS with each other with semisupervised learning. After supervised learning with monolingual data, the machine speech chain is then carried out with unsupervised learning of either the CS text or speech. The results show that the machine speech chain trains ASR and TTS together and improves performance without requiring the pair of CS speech and corresponding CS text. We also integrate language embedding and language identification into the CS machine speech chain in order to handle CS better by giving language information. We demonstrate that our proposed approach can improve the performance on both a single CS language pair and multiple CS language pairs, including the unknown CS excluded from training data.

key words: *ASR, code-switching, language identification, semisupervised learning, TTS, machine speech chain*

1. Introduction

Japan's bilingual community is growing up. The number of Japanese school-age children who have lived abroad has more than doubled over the past 40 years [1]. The number of Japanese children with foreign parents has also risen compared to 25 years ago. The number of foreign tourists and residents also steadily increases due to tourism, education, and health. These changes affect how people communicate with each other.

Bilingual or multilingual speakers often alternate between languages in a conversation, which phenomenon is called code-switching (CS). CS is a characteristic of bilingual communities [2]. Nakamura studied a Japanese child in America who used 179 switches during a one-hour conversation with his mother [3]. Fotos et al. investigated the four-hour conversations of four English-Japanese bilingual children living in Japan and observed CS 153 times [4]. Both reports reveal that people use English-Japanese CS in everyday life. Therefore, CS automatic speech recognition (ASR) and text-to-speech (TTS) must be developed to handle not only monolingual but also CS.

Unfortunately, common methods of developing CS ASR and TTS are separate training, where just CS ASR or CS TTS is developed. Moreover, it relies on supervised learning that requires large amounts of CS data for training models. Although either CS text or CS speech may be found on social media, pairs of CS speech and corresponding CS transcriptions are scarce and difficult to obtain. Such a data problem hinders the development of CS ASR and TTS.

On the other hand, recently, a framework called a machine speech chain [5], [6] was proposed to achieve semisupervised learning for ASR and TTS, trainable with labeled and unlabeled data. The machine speech chain mechanism has a feedback loop between ASR and TTS, allowing them to support each other given the available unpaired speech or text data (unlabeled data). However, the existing works on machine speech chains [5], [6] have only addressed the monolingual issue.

Therefore, in this study, we propose utilizing the machine speech chain for CS task to handle not only monolingual but also bilingual. First, we train ASR and TTS with the labeled monolingual data in supervised learning. Next, we perform a machine speech chain with the unsupervised learning with only CS text or CS speech without requiring any labeled CS data. We also extend the machine speech chain to handle CS better by integrating language embedding and language identification (LID) and investigate our proposed model's performance both on a single CS language pair and multiple CS language pairs. The multiple CS language pairs include the unknown CS excluded from the training data. The task of predicting the unknown CS without training is called a zero-shot CS. It is difficult to predict the switching points and the used language in that situation since the target CS is not used as training data. We expect that language embedding and LID can solve these problems by delivering language information in the training.

Finally, this study provides more expansion than our previous CS machine speech chain works [7], [8]. We handle non-native CS as well, using the natural Mandarin-English CS data, South East Asia Mandarin-English (SEAME) corpus [9]. We control the accented problem better by utilizing efficient pronunciation-assisted subword modeling (PASM) [10].

Manuscript received January 6, 2021.
Manuscript revised April 22, 2021.
Manuscript publicized July 8, 2021.

[†]The authors are with the Augmented Human Communication Lab, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

[††]The authors are with the RIKEN, Center for Advanced Intelligence Project AIP, Ikoma-shi, 630–0192 Japan.

[*]Presently, with Facebook AI, USA

a) E-mail: nakayama.sahoko.nq1@is.naist.jp

DOI: 10.1587/transinf.2021EDP7005

2. Related Works

2.1 Code-switching

Several studies have addressed ASR for the CS of specific language pairs, such as Mandarin-English [11]–[13], English-Malay [14], and Frisian-Dutch [15]. Semisupervised acoustic and lexicon learning for English-Mandarin CS ASR have been proposed [16]. Although that work achieved CS ASR with semisupervised learning, it only focused on a single CS language pair for ASR. In TTS studies, approaches for Mandarin-English [17], [18], German-English [19], [20], Hindi-English, Telugu-English, Marathi-English, and Tamil-English [21] CS have been investigated.

Beyond a single CS language pair, White et al. [22] explored a method to model the acoustics between multiple CS language pairs, and Imseng et al. [23] proposed an approach to estimate the universal phoneme posterior probabilities of mixed language speech recognition. Another alternative is a language-independent ASR for multiple CS language pairs [24]. However, these approaches just relied on supervised learning and handled only ASR.

Most previous researches suffer from one or more of the following disadvantages: (a) developed on either only ASR or only TTS; (b) focused only on a single CS language pair; (c) trained in supervised learning that requires a large amount of labeled CS data in which the CS speech and corresponding CS transcriptions are hard to obtain. In contrast, our study builds end-to-end encoder-decoder models for both CS ASR and TTS and connects them so that they train each other. The machine speech chain framework can train CS ASR and CS TTS together in semisupervised learning, even without labeled CS data. We also handle multiple CS language pairs not only a single CS language pair. We integrate language embedding and LID into the machine speech chain and explore how well the model performs well on both a single CS language pair and multiple CS language pairs, including the unknown CS excluded from the training data, called zero-shot CS.

2.2 Zero-Shot Learning

Zero-shot learning, which was initially proposed in the field of computer vision, refers to the problem of recognizing objects that may not have appeared in the training data in multiclass classification [25]. In machine translation, zero-shot tasks faced the challenge of translating the language combinations that were excluded in training sets [26]. Unfortunately, few studies have addressed CS ASR and TTS, so this study has contributed to the zero-shot CS ASR and TTS.

2.3 Synthetic Data

Since obtaining a large amount of data takes time and money, some researchers in several fields of spoken language technologies have utilized synthetic data to improve the quality of their systems. Jia et al. [27] used synthetic data and machine translation for improving end-to-end speech-to-text translation models. Hasegawa-Johnson et al. [28] trained image-to-speech models with SPEECH-COCO [29], a synthetic speech corpus generated by TTS. Synthetic data were also used for training ASR and TTS [5]. They conducted experiments with synthetic data as well as those with natural data, and both sets of results showed the same tendency of their proposed model to improve the ASR and TTS performances. Therefore, synthetic data can be utilized for covering low-resourced data. One of the low-resourced data is CS. The existing corpus is limited to some language pairs and accents, and difficult to collect a new corpus of natural CS. Although we may find either CS speech or CS text in social media, the annotation for CS data requires high language skills. Some researchers actually utilized synthetic CS data to improve their CS system's quality [30], [31]. Similarly, we utilized synthetic data for covering low-resourced CS data even though we also experimented with natural data.

3. Code-switching Categories

3.1 Switching Positions

CS phenomena can be divided into two main categories: intra-sentential and inter-sentential. In intra-sentential CS, the language shift occurs within a sentence. The intra-sentential CS may be inserted from the length of a single word to phrases that exceed the loanwords. In inter-sentential CS, language switching occurs at the sentence boundaries. We show some English-Japanese CS examples collected from a bilingual CS user:

- **Intra-sentential CS:**
  - **Word-level CS:**
    "Kokkai ga the Equal Employment Opportunity Law ni bassoku wo moukenakatta node kuubun da toiu iken ga ari masu." (*Since the Diet did not put any teeth into the Equal Employment Opportunity Law, some believe that it is merely a scrap of paper.*)
  - **Phrase-level CS:**
    "If I could make a suggestion, kono gidai ni tsuite no tougi wo tyusyoku made ni oe te itadakereba to omoi masu ga." (*If I could make a suggestion, why do not we finish discussing this subject by lunch?*)
- **Inter-sentential CS:**
  - **Inter-sentential CS:**
    "In the end, he quit his job and followed in his father's footsteps, taking over the family business. Yappari kaeru no ko wa kaeru da ne." (*His son's a chip off the old block, all right. In the end, he quit his job and followed in his father's footsteps, taking over the family business.*)

However, CS's definition is controversial. Loanwords, which are borrowed from a foreign language, may not be included in intra-sentential word-level CSs, and quotations, which borrow part of another's text or speech, may not be included in the intra-sentential phrase-level CSs. Although they may not be CS in principle, we include them in our CS targets because we want to recognize all of the words in multilingual conversations.

3.2 Language Proficiency

CS switches between the first language (L1) and the second language (L2), where only one of the languages is the mother tongue. The proficiency level of the L2 language varies from beginners to near-native speakers. Handling them together may degrade the ASR performance since it causes a mismatch between speech and acoustic models [32]. Therefore, we categorize CS with native CS and non-native CS based on the proficiency level of the L2 language. In the native CS, the L2 language of the CS is near-native speaker level. The non-native CS tends to make distinctive non-native sounds. In this work, we handle both native CS and non-native CS.

4. Speech Chain Framework

4.1 Human Speech Chain

The human speech chain [33] is an essential mechanism for communication. We communicate by expressing our thoughts and listening to others. This speaking and listening cycle also occurs when we talk to ourselves. When we utter a word, we aurally check whether we spoke it as we intended. We simultaneously improve speaking and listening while alternately repeating sounds and words. The human speech chain is defined by such a communication cycle.

4.2 Machine Speech Chain

Tjandra et al. developed a deep learning-based monolingual machine speech chain [5], [6], [34], inspired by the human speech chain as Fig. 1 shows. Its framework is illustrated in Fig. 2. It is composed of an end-to-end ASR [35], [36] and an end-to-end TTS [37], and they are connected. The architecture can train ASR and TTS each other with their feedback. The monolingual machine speech chain [5] improves the performance of monolingual ASR and TTS. The multi-speaker machine speech chain [5], [6] is expanded to deal with multi-speakers by integrating speaker recognition (SPKREC) based on DeepSpeaker [38]. Still, they are only for monolingual language; it cannot handle CS. Therefore, in this study, we expand it to handle CS.

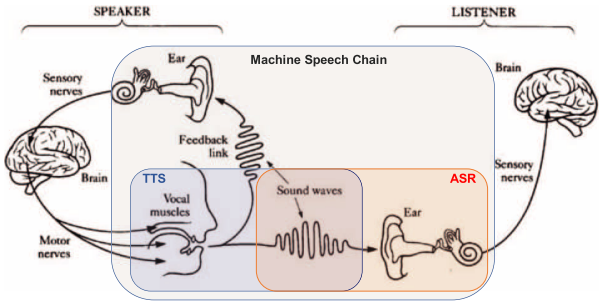


Fig. 1 Human speech chain [33] and the corresponding machine speech chain [5]. Source: adapted from [33].

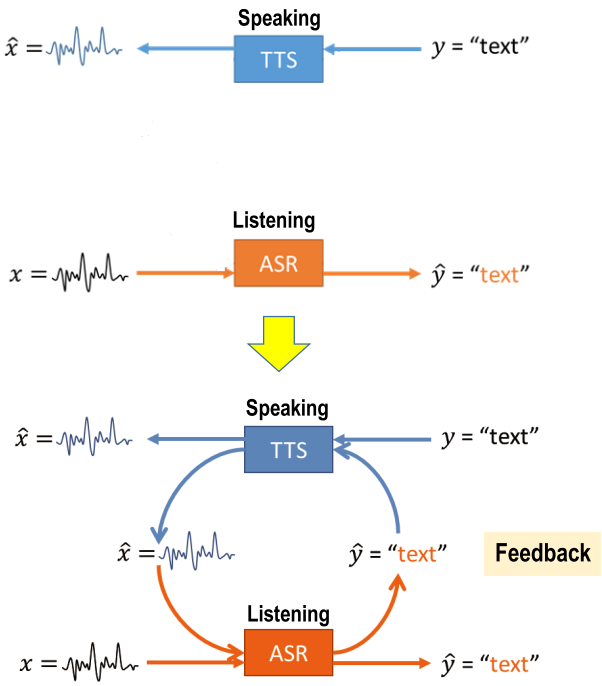


Fig. 2 The overview of machine speech chain framework [5].

5. Component Technologies: ASR, TTS, and SPKREC

5.1 ASR

We use an encoder-decoder with an attention model [35], [36] for ASR. An overview is shown in Fig. 3.

In the encoder, from the input sequences of speech features $x = (x_1, x_2, \cdots, x_T)$, it outputs a hidden vector, where we used three bidirectional LSTM layers with 256 hidden states:

$$h_t^f = LSTM(h_{t-1}^f, x_t), \quad (1)$$

$$h_t^b = LSTM(h_{t+1}^b, x_t), \quad (2)$$

where h_t^f is a forward hidden vector at time t and h_t^b is a backward hidden vector h_t^b at time t. The final hidden vector concatenated h_t^f and h_t^b, which is then denoted as h_t at time t.

An attention mechanism [39] can map between encoder

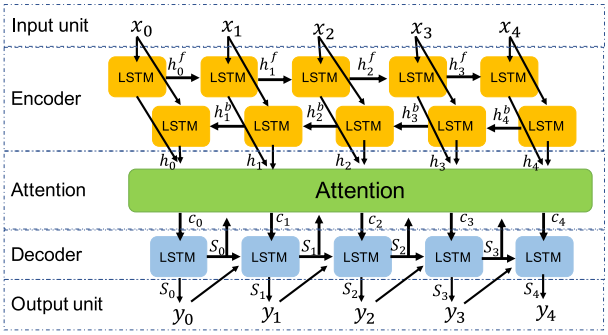


Fig. 3 Attention-based encoder-decoder.

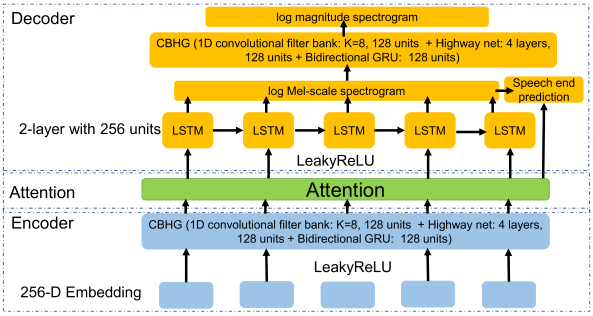


Fig. 4 Tacotron in machine speech chain for a single speaker.

and decoder. It calculates context vector c_t using attention weights α_{tj}, which obtain the most relevant encoder representation to the decoder state:

$$c_t = \sum_{j=0}^{J} \alpha_{tj} h_j, \tag{3}$$

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})}, \tag{4}$$

$$e_{tj} = Score(s_t, h_j). \tag{5}$$

The *Score* function determines how the encoder and decoder outputs are related, where s_t is the decoder's hidden vector and h_j is the encoder's hidden vector. The calculation had three ways in a previous work [40], but we adopt the calculating method by the multilayer perceptron (MLP):

$$Score(s_t, h_j) = w_a^{\mathsf{T}} tanh(W_a[s_t, h_j]), \tag{6}$$

where w_a, W_a is the weight vector and *tanh* is an activation function.

The decoder generates output y_t with all the previously predicted words $y_1, y_2, \cdots, y_{t-1}$ and context vector c_t:

$$p(y_t|y_1, y_2, \cdots, y_{t-1}, c_t) = g(s_t, y_{t-1}, c_t), \tag{7}$$

where g is an activation function that calculates the probability of y_t. The decoder hidden vector s_t is calculated with the LSTM layer:

$$s_t = LSTM(s_{t-1}, y_{t-1}, c_t). \tag{8}$$

For optimizing ASR, we attempt to decrease the negative log-likelihood loss function to maximize the probability of target sequence y with the context vector of decoder input c and previous output $y_{1:t-1}$:

$$L_{ASR} = -\sum_{t=1}^{T} \log P(y_t|c_t, y_{1:t-1}), \tag{9}$$

where posterior probability $P(y_t|c_t, y_{1:t-1})$ is calculated by a softmax function.

5.2 TTS

The TTS system of the machine speech chain is shown in Fig. 4. It is based on an encoder-decoder TTS (Tacotron) [41]. The hyperparameters are almost the same as the original Tacotron, but the activation function replaced ReLU with LeakyReLU [42]. The encoder CBHG module's convolutional filters are eight sets instead of 16 sets of original Tacotron to conserve GPU memory. The decoder replaces GRU with two stacked LSTMs that have 256 hidden states in each layer. The decoder has a process that predicts the speech's end frame, which is decided by the binary prediction of the log Mel-spectrogram and the context vector from the attention module. The loss function for training TTS used a combination of mean squared error (MSE) in the log Mel-spectrogram and MSE in the log magnitude spectrogram and binary cross-entropy in the prediction for the speech's end frame as follows:

$$L_{TTS} = \frac{1}{T}\sum_{t=1}^{T}\{(m_t - \hat{m}_t)^2 + (r_t - \hat{r}_t)^2 - (b_t \log(\hat{b}_t) + (1 - b_t)\log(1 - \hat{b}_t))\}, \tag{10}$$

where the first term of the summation is the MSE between target log Mel-spectrogram m and predicted log Mel-spectrogram $\hat{m}$, the second term is the MSE between target log magnitude spectrogram r and predicted log magnitude spectrogram $\hat{r}$, and the third term is the binary cross-entropy between the target probability end frame of speech b and predicted probability end frame of speech $\hat{b}$.

5.3 Speaker Recognition for Multi-Speaker

Since the original Tacotron is a single speaker model and cannot deal with multi-speakers, we generate speaker vectors with the DNN-based speaker recognition (SPKREC) DeepSpeaker [38] and take them into Tacotron.

In DeepSpeaker, the DNN architectures (we used Residual CNN) extracted the frame features from the utterances. After converting the frame features to a speaker representation for an utterance unit, it is embedded into a 512-dimensional representation. The embedding vectors are normalized to the unit norm and by cosine similarity between two embedding vectors:

$$cos(x_i, x_j) = x_i x_j, \tag{11}$$

where x_i, and x_j are the embedding vectors.

Finally, the model is trained using the following loss

function, which maximizes the cosine similarities of the embedding vectors from the same speaker while minimizing those from different speakers for N triplets:

$$L_{triplet} = \sum_{i=0}^{N} max((s_i^{an} - s_i^{ap} + \alpha), 0), \tag{12}$$

where s_i^{ap} is the cosine similarity between an utterance a of a speaker and another utterance p of the same speaker in triplet i. s_i^{an} is the cosine similarity between an utterance a of a speaker and an utterance n of another speaker in triplet i.

After the DeepSpeaker models are trained, we generate speaker embedding vector s. The generated speaker vector is used in the speaker embedding of the multi-speakers Tacotron (Fig. 5). The speaker vector is concatenated with the encoder output and goes through the decoder. In the loss function, we use the extension of Eq. (10) by adding the formula of speaker loss for handling multiple speakers as follows:

$$\begin{aligned} L_{TTSspeaker} = \frac{1}{T}\sum_{t=1}^{T}\{&\gamma_1((m_t - \hat{m}_t)^2 + (r_t - \hat{r}_t)^2) \\ &- \gamma_2((b_t \log(\hat{b}_t) + (1 - b_t)\log(1 - \hat{b}_t)))\} \\ &+ \gamma_3(1 - \frac{s \cdot \hat{s}}{\|s\|_2 \cdot \|\hat{s}\|_2}), \end{aligned} \tag{13}$$

where the first term is an MSE that compares target log Mel-spectrogram m with predicted log Mel-spectrogram $\hat{m}$, the second term is an MSE that compares target log magnitude spectrogram r with predicted log magnitude spectrogram $\hat{r}$, the third term is the binary cross-entropy comparing target speech's end probability b with predicted speech's end probability $\hat{b}$, and the last term is the cosine distance comparing target speaker vector s with predicted speaker vector $\hat{s}$. $\gamma_1, \gamma_2, \gamma_3$ are hyperparameters that adjust the balance among the three losses. In our experiments, we set the hyperparameters for calculating the loss as $\gamma_1 = 1, \gamma_2 = 1, \gamma_3 = 0.25$ in Eq. (13).

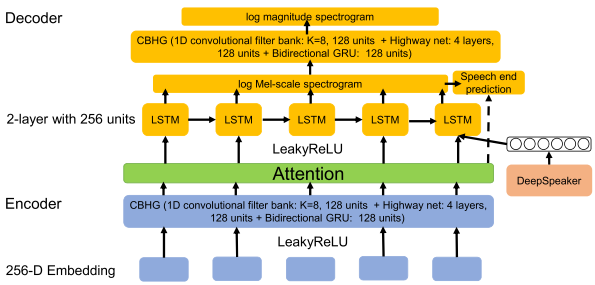


Fig. 5 Tacotron in machine speech chain for multi-speakers.

6. Proposed Machine Speech Chain for Code-Switching

Figure 6 shows the differences among the following: (a) a basic CS machine speech chain [7], (b) a multi-speaker CS machine speech chain that incorporates SPKREC for handling multiple speakers, and (c) a language-aware CS machine speech chain [8].

6.1 Basic Code-switching Machine Speech Chain

The basic CS machine speech chain (Fig. 7) seeks to improve the ASR and TTS performance on CS without any labeled CS data. The learning process is as follows (In the case of handling multiple speakers, the speaker vector z = SPKREC(**x**) is added to the input of the TTS decoder both during supervised and unsupervised processes):

1. **Supervised learning of ASR and TTS with speech-to-text paired monolingual data**
   First, ASR and TTS are trained in supervised learning with the speech-to-text paired Japanese and English data or the speech-to-text paired Mandarin and English data (mixed data of monolingual sets constituting a CS language pair) as shown in Fig. 7 (a). Once ASR receives the speech and the corresponding text (x^{Mono}, y^{Mono}), ASR recognizes speech $\hat{y}^{Mono}$ using teacher-forcing, which is an algorithm that trains efficiently and converges faster by direct training with the target label. Then, the loss between output text $\hat{y}^{Mono}$ and reference text y^{Mono} is calculated $L_{ASR}^{Mono}(\hat{y}^{Mono}, y^{Mono})$ using Eq. (9). TTS also generates speech $\hat{x}^{Mono}$ from the input text y^{Mono}, and the loss between generated speech $\hat{x}^{Mono}$ and reference speech x^{Mono} is calculated $L_{TTS}^{Mono}(\hat{x}^{Mono}, x^{Mono})$, where the loss function in case of single-speaker is Eq. (10) and the loss function in case of multi-speaker is Eq. (13). The

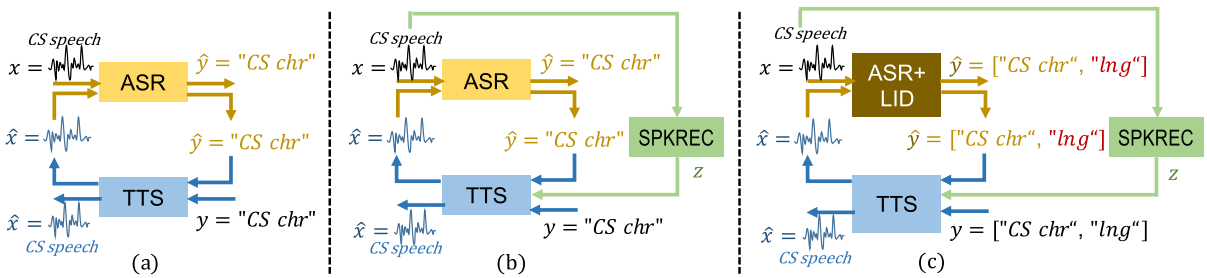


Fig. 6 Comparison among CS machine speech chain models: (a) basic CS machine speech chain [7]; (b) multi-speaker CS machine speech chain incorporating SPKREC; (c) language-aware CS machine speech chain [8].

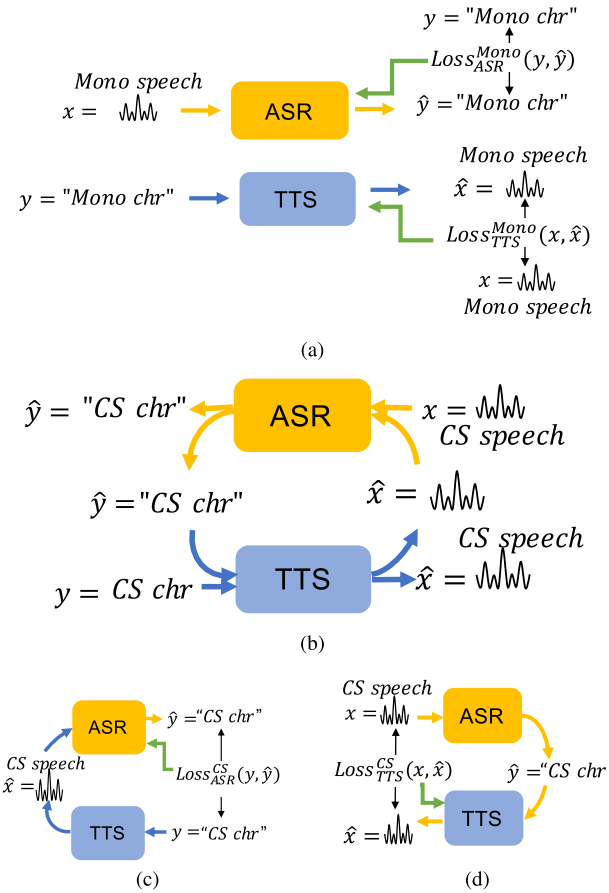


Fig. 7 Overview of the proposed framework based on [5], [7]: (a) supervised learning of ASR and TTS with speech-to-text paired monolingual data of two languages; (b) unsupervised learning of ASR and TTS together through machine speech chain with unpaired CS text data or unpaired CS speech data; (c) loop connection from TTS to ASR with only unpaired CS text data; (d) loop connection from ASR to TTS with only unpaired CS speech data.

parameters are tuned to decrease the loss with gradient descent optimization.

2. **Unsupervised learning of ASR and TTS together in a machine speech chain**
We performed a machine speech chain, where we trained ASR and TTS together with an unpaired CS text or an unpaired CS speech data (Fig. 7 (b)).
The learning process during the unsupervised learning of the machine speech chain consists of the following two processes:

   a. **Loop connection from TTS to ASR with only unpaired CS text data**
   This process (Fig. 7 (c)) only uses unpaired CS text data y^{CS}. TTS outputs speech $\hat{x}^{CS}$ from the input CS text y^{CS}, and ASR also predicts text transcription $\hat{y}^{CS}$ from the synthesized speech. Then loss $L^{CS}_{ASR}(\hat{y}^{CS}, y^{CS})$ can be computed between output text $\hat{y}^{CS}$ and input text y^{CS} to tune the ASR parameters.

   b. **Loop connection from ASR to TTS with only unpaired CS speech data**
   This process (Fig. 7 (d)) only uses unpaired CS speech data x^{CS}. Once ASR receives speech x^{CS}, ASR outputs predicted transcription $\hat{y}^{CS}$, and TTS generates speech $\hat{x}^{CS}$ from the text of the ASR output. The loss between output speech $\hat{x}^{CS}$ and original speech x^{CS} can be computed $L^{CS}_{TTS}(\hat{x}^{CS}, x^{CS})$ for tuning the TTS parameters.

During the learning process of unsupervised learning, we also continue the supervised learning process. The supervised learning loss and unsupervised learning loss are integrated into a single loss:

$$L_{Chain} = \alpha(L^{Mono}_{ASR} + L^{Mono}_{TTS}) + \beta(L^{CS}_{ASR} + L^{CS}_{TTS}), \quad (14)$$

$$\theta_{ASR} = Optim(\theta_{ASR}, \nabla_{\theta_{ASR}} L_{Chain}), \quad (15)$$

$$\theta_{TTS} = Optim(\theta_{TTS}, \nabla_{\theta_{TTS}} L_{Chain}), \quad (16)$$

where the hyperparameters α and β tune the balance of the losses. They balance the influence between the supervised and unsupervised, and between the monolingual and CS data.

6.2 Language-Aware Code-Switching Machine Speech Chain

In a language-aware CS machine speech chain, we handle CS more efficiently with language information. To achieve this, we put additional functions, LID for ASR and language embedding for TTS. As Fig. 8 shows, the LID architecture performs multi-task learning in the ASR softmax layers. The architecture trains the projection between the speech input and the two outputs of the text transcription and the language information with two softmax layers (Fig. 8). The language information is given to each character by the language ID. For language IDs, Japanese is denoted as "JA," English is denoted as "EN," Chinese is denoted as "ZH," and an unknown language is denoted as "<unk>."

The language embedding of TTS maps a one-hot vector representing a language ID into continuous vectors. Then, it concatenates with the character embedding and goes through the encoder LSTM, attention, decoder, and generates speech. In the case of handling multiple speakers, the speaker vector z = SPKREC(**x**) is added to the input of the TTS decoder both during supervised and unsupervised training.

The training process is almost same as the basic CS machine speech chain, but the language-aware CS machine speech chain trains language information. The following is the training process:

1. **Supervised learning of ASR and TTS with speech-to-text paired monolingual data**
As shown in Fig. 9 (a), we first train the ASR and TTS systems with the speech-to-text paired monolingual corpora from several languages using English (En), Japanese (Ja), and Chinese

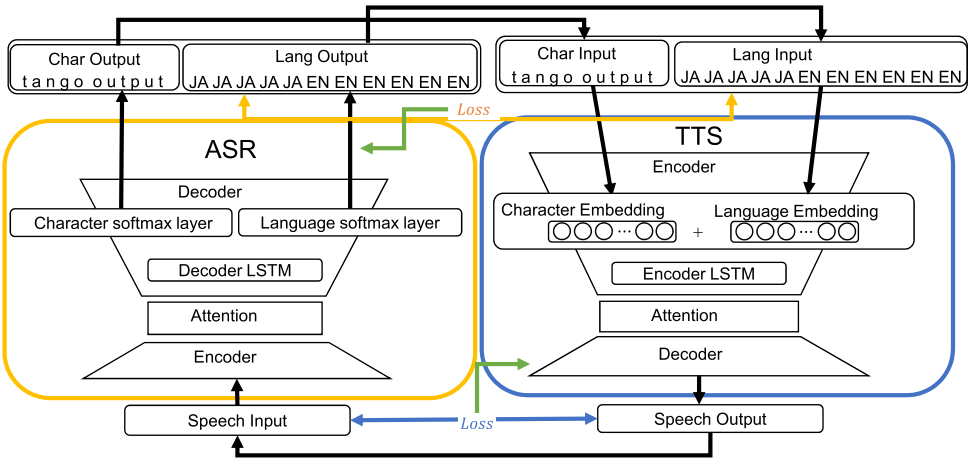
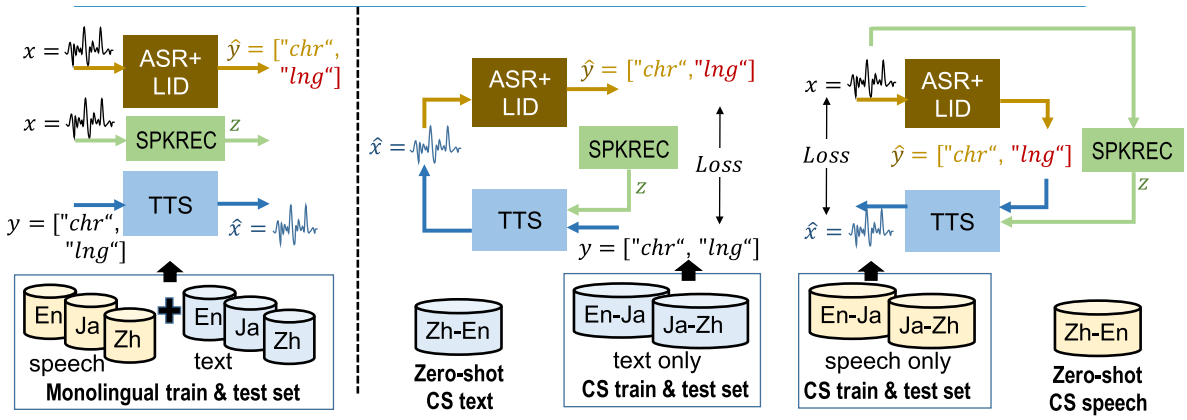


Fig. 8 Language-aware code-switching machine speech chain.



(a) Supervised learning on monolingual data

(b) Unsupervised learning on code-switching data

Fig. 9 Training overview of language-aware CS machine speech chain [8]: (a) supervised training of ASR or TTS with speech-to-text paired monolingual data; (b) unsupervised training of a machine speech chain with unpaired CS text or CS speech data.

(Zh). With speech-to-text paired monolingual data (x^{Mono}, $y^{MonoChr}$, and $y^{MonoLng}$), ASR generates sequences of characters $\hat{y}^{MonoChr}$ and language information $\hat{y}^{MonoLng}$ with teacher-forcing and calculates the sum of losses $L_{ASR}^{MonoChr}(\hat{y}^{MonoChr}, y^{MonoChr})$ and $L_{ASR}^{MonoLng}(\hat{y}^{MonoLng}, y^{MonoLng})$. The loss function for optimizing ASR (Eq. (9)) changes to the following function in accordance with the incorporating LID loss:

$$L_{ASR}^{LngAwr} = \lambda_{Chr} L_{ASR}^{Chr} + \lambda_{Lng} L_{ASR}^{Lng}, \tag{17}$$

where it's a summation of two negative log-likelihood, tuning those weights by the hyperparameters λ_{Chr} and λ_{Lng}. TTS generates sequence of speech features $\hat{x}^{Mono}$ with teacher-forcing, and we calculate loss $L_{TTS}^{Mono}(\hat{x}^{Mono}, x^{Mono})$. The TTS loss function does not change from Eq. (13) since the TTS output does not change. The parameters are tuned to reduce the loss with gradient descent optimization.

2. **Unsupervised training of ASR and TTS together in a machine speech chain**

   a. **Loop connection from TTS to ASR with only unpaired CS text data of characters and language information**

   This process (Fig. 9 (b), left side) uses only unpaired CS text data of characters and language information [y^{CSChr}, and y^{CSLng}]. TTS outputs speech $\hat{x}^{CS}$ from the unpaired CS text data of the characters and language information [y^{CSChr}, y^{CSLng}]. The generated speech is transcribed by ASR to the CS text [$\hat{y}^{CSChr}, \hat{y}^{CSLng}$]. Then the sum of losses $L_{ASR}^{CSChr}(\hat{y}^{CSChr}, y^{CSChr})$ and $L_{ASR}^{CSLng}(\hat{y}^{CSLng}, y^{CSLng})$ can be computed to update the ASR parameters.

   b. **Loop connection from ASR to TTS with only unpaired CS speech data**

   This process (Fig. 9 (b), right side) only uses

CS speech x^{CS} as input. With unlabeled CS speech x^{CS}, ASR generates sequences of characters $\hat{y}^{CSChr}$ and language information $\hat{y}^{CSLng}$. TTS generates CS speech $\hat{x}^{CS}$ with output CS characters and language information from ASR. Then TTS parameters are tuned to decrease loss $L^{CS}_{TTS}(\hat{x}^{CS}, x^{CS})$.

In the end, the losses of the supervised monolingual and unsupervised CS losses are combined into a single loss:

$$\begin{aligned} L^{LngAwr}_{Chain} &= \alpha((\lambda_{Chr}L^{MonoChr}_{ASR} + \lambda_{Lng}L^{MonoLng}_{ASR}) \\ &\quad + L^{Mono}_{TTS}) \\ &\quad + \beta((\lambda_{Chr}L^{CSChr}_{ASR} + \lambda_{Lng}L^{CSLng}_{ASR}) \\ &\quad + L^{CS}_{TTS}), \end{aligned} \tag{18}$$

$$\theta_{ASR} = Optim(\theta_{ASR}, \nabla_{\theta_{ASR}} L^{LngAwr}_{Chain}), \tag{19}$$

$$\theta_{TTS} = Optim(\theta_{TTS}, \nabla_{\theta_{TTS}} L^{LngAwr}_{Chain}), \tag{20}$$

where the hyperparameters α and β tune the balance of the losses. They balance the influence between the supervised and unsupervised, and between the monolingual and CS data. After training, we can perform the ASR and TTS on zero-shot CS.

7. Experiments

7.1 Datasets

7.1.1 BTEC

We used the Basic Travel Expression Corpus (BTEC) collected by the Advanced Telecommunications Research Institute International (ATR) [43], [44] and prepared datasets for a single CS language pair and for multiple CS language pairs respectively.

For a single CS language pair, we used monolingual Japanese and English BTEC and chose randomly 50k sentences for training, 500 sentences for a development set, and 500 sentences for a test set from BTEC1-4. We also constructed an English-Japanese CS dataset from monolingual Japanese and English BTEC sentences and created word-level and phrase-level intra-sentential CS. Figure 10 shows an overview of the construction of the CS text data, and more details are described in [45]. We generated their speech utilizing the Google TTS (gTTS) python library [46] because collecting the speech data for CS from bilingual speakers is time-consuming and expensive. Table 1 shows the statistics of these datasets used for a single CS language pair's machine speech chain.

For multiple CS language pairs, we used the monolingual Japanese, English, and Chinese of BTEC [43], [44]. We chose sentences that could be separated into phrases by commas. From those sentences, we randomly selected 25k Japanese sentences, 25k English sentences, and 25k

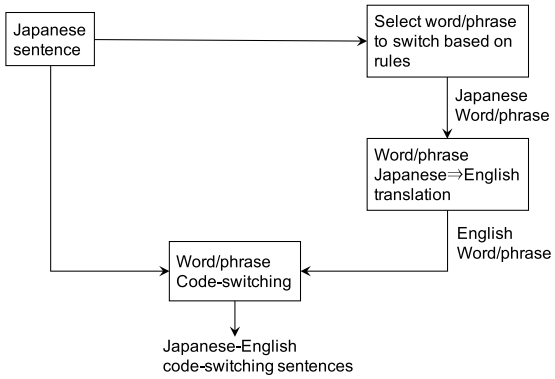


Fig. 10 Japanese-English CS text data construction [45].

Table 1 Statistics of BTEC for a single CS language pair's machine speech chain.

subset			hours	utterances
train	mono	Ja25k+En25k (JaTTS)	50.7	50000
		Ja25k+En25k (MixTTS)	39.6	50000
	CS	EnJaCS10k (JaTTS)	9.5	10000
		EnJaCS10k (MixTTS)	8.9	10000
		EnJaCS20k (JaTTS)	19.0	20000
		EnJaCS20k (MixTTS)	17.8	20000
		EnJaCS20k (Ja+MixTTS)	18.4	20000
test	mono	TstJa (JaTTS)	0.7	500
		TstEn (EnTTS)	0.6	500
	CS	TstEnJaCS (JaTTS)	1.1	500
		TstEnJaCS (MixTTS)	0.7	500

Chinese sentences: "Ja25k+En25k+Zh25k." We also selected 500 Japanese sentences, 500 English sentences, and 500 Chinese sentences for the development and test sets. We artificially created CS sentences from the selected monolingual BTEC sentences by translating the first phrase switched at the comma of the sentences to the other languages and inserting it into the original sentences again (similar to our previous approach [45]). We constructed an English-Japanese CS, "EnJaCS," a Japanese-Chinese CS, "JaZhCS," a Chinese-English CS "ZhEnCS," an English-French CS "EnFrCS," and a French-Chinese CS, "FrZhCS." They were also generated with Google TTS [46]. For the CS natural speech corpus, we also collected 1k utterances made by an English-Japanese bilingual CS user who often uses CS in his daily life. He created natural CS text from a BTEC English-Japanese translation corpus. After that, we recorded the reading speech by another English-Japanese bilingual speaker. We used every 100 utterances for a development set and a test set and the remaining 800 utterances of data for a training set. We divided the collected 1k utterances into 0.2k labeled data, 0.7k unlabeled data denoted as "NatEnJaCS," and 0.1k test data denoted as "TstNatEnJaCS." The labeled data were also divided between Japanese and English, which can be used for monolingual data. Those data will be later called "NatJa" and "NatEn." Table 2 shows the statistics of these corpora used for multiple CS language pairs' machine speech chain.

Table 2 Statistics of BTEC for multiple CS language pairs' machine speech chain.

subset			hours	utterances
train	mono	Ja25k+En25k+Zh25k	82.1	78000
		NatJa0.2k+NatEn0.2k	0.3	400
	CS	EnJaCS10k	11.8	10000
		JaZhCS10k	10.4	10000
		ZhEnCS10k	10.9	10000
		EnFrCS10k	9.6	10000
		NatEnJaCS0.7k	1.1	700
test	mono	Ja	0.9	500
		En	0.7	500
		Zh	0.8	500
	CS	EnJaCS	0.8	500
		JaZhCS	0.7	500
		ZhEnCS	0.7	500
		TstNatEnJaCS	0.2	100

7.1.2 LibriSpeech

We used LibriSpeech [47] for supporting monolingual training when training the machine speech chain with the natural speech of the SEAME data. LibriSpeech, an English speech corpus, is based on free public domain audiobooks read by volunteers. We used an officially prepared 100-hour subset.

7.1.3 AISHELL-1

We used AISHELL-1 [48] for supporting monolingual training when training the machine speech chain with the natural speech of the SEAME data. AISHELL-1 is a read Mandarin speech corpus. Most of its speakers are from Northern China, and some are from Southern China, Guangdong-Guangxi-Fujian, and others. It contains 150 hours of speech, but we only used 100 hours.

7.1.4 SEAME

We used the SEAME [9] corpus for a single CS language pair's machine speech chain with natural speech. SEAME is a conversational Mandarin-English CS corpus, collected from Singaporean and Malaysian speakers. We divided it into train, dev_{man}, and dev_{sge} datasets in accordance with previous works [49]. dev_{man} is a test set dominated by Mandarin words, and dev_{sge} is a test set dominated by English words. We also divided the train dataset into monolingual and CS subsets. We combined the SEAME monolingual subset with a 100-hour subset of LibriSpeech and a 100-hour subset of AISHELL-1 to train the monolingual model. We also used the SEAME CS subset for machine speech chain training. We applied data augmentation of speed perturbation [50], [51] with 90%, 100%, and 110% to both the monolingual and CS training data. For controlling the accented problems, we utilized PASM sub-word units, which are sub-word units optimized for accents by taking the alignments between phonemes and characters [10]. Table 3 shows the statistics of these datasets used for a single CS language pair's machine speech chain with natural speech.

Table 3 Statistics of LibriSpeech, AISHELL-1, and SEAME for a single CS language pair's machine speech chain with natural speech.

subset			speakers	hours	utterances
train	mono	LibriSpeech	251	100.6	28539
		AISHELL-1	340	100.0	80066
		SEAME	134	31.5	42911
		all	725	232.1	151516
	CS	SEAME	134	69.6	51027
test	dev_{man}	*mono*	10	1.6	2228
		CS	10	5.9	4303
		all	10	7.5	6531
	dev_{sge}	*mono*	10	1.8	3156
		CS	10	2.1	2165
		all	10	3.9	5321

7.2 Settings

We sampled all the speech signals at a sampling rate of 16-kHz. Then we applied pre-emphasis and normalized the speech signals between −1 and 1. We extracted the spectrogram features using a short-time Fourier transform (STFT) with the Librosa library [52]. The frame had a 50-ms length and a 12.5-ms shift, and the FFT points are 2048. From the spectrogram, we computed the magnitude spectrogram and mapped it to the Mel-scale spectrogram. Those features were transformed to log-scale and normalized into 0 mean and unit variances. Finally, we got 80 dimensions of log Mel-spectrogram features and 1025 dimensions of log magnitude spectrograms.

All the text characters were converted to lowercase letters and punctuation marks [, : ? .] were removed. We converted all BTEC characters into the lowercase alphabet. For Japanese words, we applied a morphological analyzer Mecab [53] to convert into katakana. Then we converted the katakana into English letters with pykakasi [54]. We also used pypinyin to the Chinese characters [55] and converted them into pinyin. The text consists of 26 letters (a-z), one mark (-) for stretching Japanese sounds, and three tags that denote the start of sentences (<s>), the end of sentences (</s>), and the spaces between words (<spc>). In the case of LibriSpeech, AISHELL-1, and SEAME, we utilized PASM sub-word units, which is a sub-word unit optimized for accents by taking alignments between phonemes and characters [10].

We implemented both ASR and TTS with the PyTorch library [56]. For the hyperparameters balancing between the supervised and unsupervised loss, most of our experiments used $\alpha = 0.5, \beta = 1$.

7.3 Experimental Results on Single Code-Switching Language Pair (Synthetic Speech; Single Speaker)

In this experiment, we used the BTEC corpus prepared for single CS language pair.

Baseline Systems

We had four types of test sets for our evaluation: (1) **TstJa (JaTTS)**: a Monolingual Japanese test set generated using a Japanese TTS; (2) **TstEnJaCS**

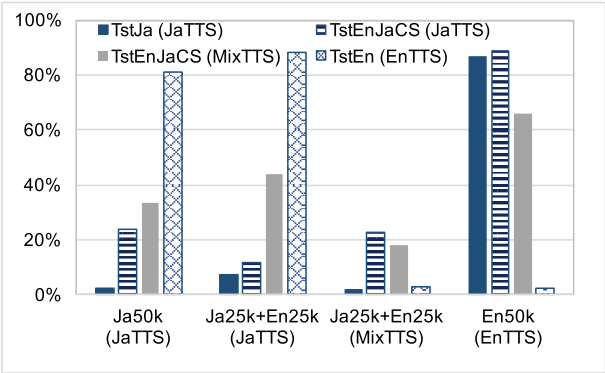


Fig. 11 ASR baseline performances of single CS language pair (synthetic speech; single speaker) in CER.

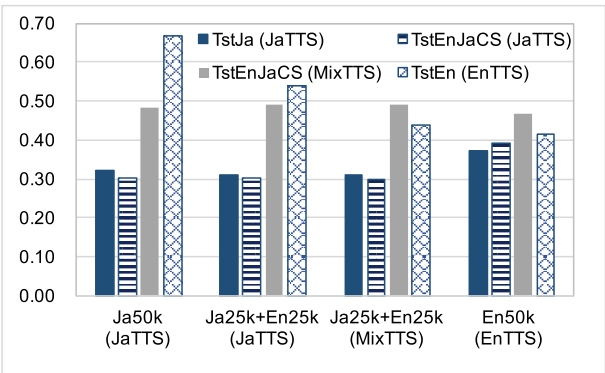


Fig. 12 TTS baseline performances of single CS language pair (synthetic speech; single speaker) in L2-norm squared of the log-Mel spectrogram.

(JaTTS): an English-Japanese intra-sentential CS test set, where both the Japanese part and the English part of the TstEnJaCS are generated using a Japanese TTS; (3) **TstEnJaCS (MixTTS)**: an English-Japanese intra-sentential CS test set generated using a mixed English-Japanese TTS, where we concatenated the speech generated by English TTS for the English part of CS and the speech generated by Japanese TTS for the Japanese part of CS; (4) **TstEn (EnTTS)**: a Monolingual English test set generated using an English TTS. Although we do not have an inter-sentential CS test set, the TstJa (JaTTS) and TstEn (EnTTS) combination are identical to the inter-sentential CS. We evaluated the generated transcription by the character error rate (CER). CER is the edit distance between the reference and the predicted transcription. We also assessed the statistical significance compared to the baseline systems by a matched-pair sentence-segment word error test [57]. For the TTS evaluation, we used the L2-norm squared of the log-Mel spectrogram between the reference and the predicted speech features.

The baseline system performances of ASR and TTS are individually shown in Figs. 11 and 12. The baseline systems were trained in supervised learning using an attention-based encoder-decoder model framework without a machine speech chain framework. Four types of baselines were evaluated: (1) **Ja50k (JaTTS)**: a 50-k monolingual Japanese ASR or TTS trained with Japanese speech generated using a Japanese TTS; (2) **Ja25k+En25k (JaTTS)**: a 25-k monolingual Japanese plus a 25-k English ASR or TTS trained with Japanese speech generated using a Japanese TTS (inter-sentential CS); (3) **Ja25k+En25k (MixTTS)**: a 25-k monolingual Japanese plus a 25-k English ASR or TTS trained with Japanese speech generated using a Japanese TTS and English speech generated using an English TTS (inter-sentential CS); (4) **En50k (EnTTS)**: a 50-k monolingual English ASR or TTS trained with speech generated using an English TTS.

As Fig. 11 shows, the CER of the Ja50k (JaTTS) ASR was low in the Japanese test, but very high in the English test. In the same way, the CER of the En50k (EnTTS) ASR was very low in the English test but increased in the Japanese test. The Ja25k+En25k (JaTTS) learned English sentences and the Japanese sentences, but when the speech was generated using Japanese TTS, the English test performance remained unsatisfactory. A similar tendency was slightly shown in the TTS results. Ja25k+En25k (MixTTS), which was trained using speech generated by a Japanese TTS and an English TTS, controlled the balance well among the Japanese, English, and English-Japanese CS test sets. Therefore, we use this Ja25k+En25k (MixTTS) as our baseline model.

Proposed Systems

Our proposed models aim to improve ASR and TTS to handle CS input well even without labeled CS for training while keeping the performance of the monolingual test. Table 4 shows the ASR and TTS performances of the proposed CS machine speech chain framework. After we individually trained ASR and TTS using labeled monolingual Ja25k and En25k, Ja25k+En25k (MixTTS), we carried out a machine speech chain on the following settings: (1) **EnJaCS (JaTTS)**: semisupervised learning with unlabeled code-switching EnJaCS (JaTTS); (2) **EnJaCS (MixTTS)**: semisupervised learning with unlabeled code-switching EnJaCS (Mix TTS); (3) **EnJaCS (Ja+MixTTS)**: a semisupervised learning with unlabeled code-switching EnJaCS (JaTTS) and unlabeled code-switching EnJaCS (MixTTS). We excluded TstEnJaCS (JaTTS) from the test sets because many English words are pronounced as Japanese words generated by the Japanese TTS.

Our results show that our proposed model with unlabeled EnJaCS20k (Ja+MixTTS) improved the ASR performance on the CS test, TstEnJaCS (MixTTS), from 18.1% CER to 5.1%, which reduced the absolute CER by 13.0%. It has statistically significant difference. Monolingual performances are often damaged by optimizing the CS performance, but our proposed model maintained its performance on the monolingual test. It only slightly changed from 1.7% CER to 1.8%

Table 4 ASR and TTS performances of a single CS language pair (synthetic speech; single speaker) machine speech chain in CER% and L2-norm squared, respectively. Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side is λ_{Lng} value during the unsupervised learning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$. The p-values compared to the baseline system for statistical significance are presented using ***, **, * and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

Model	LID type	TstJa (JaTTS)		TstEnJaCS (MixTTS)		TstEn (EnTTS)	
		ASR	**TTS**	**ASR**	**TTS**	**ASR**	**TTS**
[Baseline] Supervised learning: labeled mono							
labeled Ja25k+En25k (MixTTS)	No LID	1.7	.312	18.1	.489	3.0	.437
[Proposed Machine Speech Chain] Semisupervised learning: labeled mono + unlabeled CS							
+ unlabeled EnJaCS10k (JaTTS)	No LID	1.9	.311	19.7	.484	4.8	.444
+ unlabeled EnJaCS20k (JaTTS)	No LID	1.9	.306	17.2	.489	4.7	.441
+ unlabeled EnJaCS10k (MixTTS)	No LID	1.8	.312	5.4 ***	.374	3.7	.437
+ unlabeled EnJaCS20k (MixTTS)	No LID	1.9	.310	5.5 ***	.368	3.6	.440
+ unlabeled EnJaCS20k (Ja+MixTTS)	No LID	1.8	.305	5.1 ***	.372	4.1	.439
+ unlabeled EnJaCS20k (Ja+MixTTS)	LID (0.25→0.0)	1.7	.394	3.7 ***	.353	3.2	.563
+ unlabeled EnJaCS20k (Ja+MixTTS)	LID (0.25→0.1)	1.7	.384	3.4 ***	.347	3.3	.558
[Topline] Supervised learning: labeled mono + labeled CS							
+ labeled EnJaCS20k (Ja+MixTTS)	No LID	5.1	.321	3.5 ***	.276	9.8	.595

Table 5 Performance comparison between ASR systems trained in proposed machine speech chain with different λ_{Lng} (where $\lambda_{Chr} = 1 - \lambda_{Lng}$) in CER %. Left side of arrows is λ_{Lng} value during supervised learning process with labeled Ja25k+En25k (MixTTS) and right side is λ_{Lng} value during unsupervised learning process with unlabeled EnJaCS20k (Ja+MixTTS).

λ_{Lng}	**TstJa (JaTTS)**	**TstEnJaCS (MixTTS)**	**TstEn (EnTTS)**
No LID	1.8	5.1	4.1
0.25→0.0	1.7	3.7	**3.2**
0.25→0.1	**1.7**	**3.4**	3.3
0.25→0.25	1.9	3.9	3.5
0.25→0.5	1.8	4.1	3.8
0.25→0.75	2.1	4.5	3.9
0.5→0.0	1.9	3.5	3.5
0.5→0.1	1.9	4.2	3.7
0.5→0.25	1.7	3.6	3.5
0.5→0.5	2.0	4.3	3.8
0.5→0.75	1.8	5.7	4.2

for the Japanese test and from 3.0% CER to 4.1% for the monolingual English test. It also improved the TTS performance on the CS test TstEnJaCS (MixTTS), where the L2-norm squared decreased from 0.489 to 0.372; the performance on the Japanese and monolingual English tests was maintained. Compared with the topline model that uses full-set data (speech+text), the proposed model reached a similar performance.

Moreover, we investigated the performance of ASR trained by the language-aware CS machine speech chain with unlabeled EnJaCS20k (Ja+MixTTS), which model is denoted as "+LID" under "the unlabeled EnJaCS20k (Ja+MixTTS)". The performances among systems with some different hyperparameters λ_{Lng} are shown in Table 5. The best performance on TstEnJaCS (MixTTS) is 3.4% CER with λ_{Lng} (0.25→0.1), which improved even more than the Basic CS machine speech chain. The model with λ_{Lng} (0.25→0.0) performed the best performance on TstJa (JaTTS), so Table 4 shows both LID results of (0.25→0.0) and (0.25→0.1). Here, 0.0 indicates that the language information is used for character prediction of the unsupervised learning process while maintaining the language information trained during the supervised learning process. Both cases of the LID showed statistically significant difference with $p < .001$.

7.4 Experimental Results on Single Code-Switching Language Pair (Natural Speech; Multi-Speaker)

In this experiment, we used SEAME, AISHELL-1, and LibriSpeech corpora.

Evaluation of our End-to-end ASR against Previous Researches

First, we compared our attention-based encoder-decoder models with Hybrid CTC/attention approaches [49] using supervised learning of the SEAME data. Following the counterpart's evaluation criterion, in this experiment, we evaluated a character-based model and a sub-word-based model with the token error rate (TER). The TER, which is calculated by the Word Error Rate (WER) for English and the Character Error Rate (CER) for Mandarin, is frequently adopted for evaluating the ASR of Mandarin-English CS because it is not affected by segmentation algorithms. For sub-words, we utilized PASM [10], whose effectiveness has already been shown for overcoming the byte-pair encoding (BPE) of sub-word units [58]. As shown in Table 6, our encoder-decoder-based model can be similar performances as the CTC-based models.

Baseline and Proposed Systems

Next we conducted machine speech chain experiments with the SEAME data. We first trained the base model with LibriSpeech, AISHELL-1, and the SEAME monolingual data. Then we performed a speech chain by the unlabeled SEAME CS data while continuing the supervised training of LibriSpeech and AISHELL-1 and the SEAME monolingual data. Ta-

ble 7 shows the ASR results in TER. The baseline is the model trained with the labeled monolingual data of LibriSpeech and AISHELL-1 and SEAME. The proposed machine speech chain model improved the ASR performances on both the *CS* test sets of dev_{man} and dev_{sge} more than the baseline performances: from 47.7% to 37.4% and from 57.7% to 47.1%. Optimizing the CS performances only slightly degraded the performances on the monolingual evaluation sets. Still, our proposed model also improved the ASR on the overall performances from 44.9% to 37.6% on dev_{man} and from 53.6% to 49.5% on dev_{sge}. The topline is the model retrained with the labeled SEAME CS data from the model trained with the labeled monolingual data of LibriSpeech and AISHELL-1 and SEAME. Compared to the topline model, the proposed model achieves a similar performance, although it did not use the labeled CS at all while the topline model used only labeled data. Label propagation is a semisupervised model retrained by newly labeling with the supervised model's output. Label propagation without any labeled CS data (semisupervised learning: labeled mono + unlabeled CS) performed the worst on *all* evaluation set of dev_{man} and dev_{sge}. It used the CS label generated from the monolingual model, which does not know the CS speech and text, for the retraining model. As a result, although the performance on monolingual is better than the proposed model since it retrains with the hypothesis generated from the model based on monolingual, it degraded the performances both on *mono* and *CS* test sets than the baseline model in TER. Therefore we removed 300 utterances as unlabeled CS and added 300 utterances (0.2% of the total CS) as labeled CS. The label propagation result (semisupervised learning: labeled mono + labeled CS + unlabeled CS) improved slightly from the baseline on the *CS* test, but it required labeled CS and not better than the proposed model. On the other hand, our proposed speech chain model improved the ASR performance without any labeled CS. It showed statistically significant improvements with $p < .001$ on *CS* and *all* of both evaluation sets. Moreover, the performance with LID is even better. The LID(0.25→0.0) produced 32.5% TER on dev_{man} *CS* and 42.3% TER on dev_{sge} *CS*.

We also checked the CER for the ASR performances (Table 8) and the L2-norm squared of a log-Mel spectrogram for the TTS performances (Table 9). They showed the same tendency as the TER results. Therefore, the proposed machine speech chain model improved the ASR and TTS performances on SEAME data without any labeled CS data.

Table 6 ASR comparison between our attention-based encoder-decoder models and Hybrid CTC/attention approaches with SEAME data (in TER %).

Model	dev_{man}	dev_{sge}
Hybrid CTC/attention char [49]	26.5	38.4
+LID [49]	25.6	37.0
Hybrid CTC/attention sub-word (BPE) [49]	26.4	36.1
+LID [49]	26.0	35.8
Att Enc-Dec char (ours)	26.2	37.8
Att Enc-Dec sub-word (PASM) (ours)	25.7	36.6

7.5 Experimental Results on Multiple Code-Switching Language Pairs

In this experiment, we used the BTEC corpus prepared for multiple CS language pairs.

ASR Evaluation

First, we checked the influence of the additional LID architecture to confirm whether that additional information hindered the original quality. We used the baseline model, an ASR **Ja25k+En25k+Zh25k (labeled)** trained with labeled monolingual data of 25k Japanese and 25k English and 25k Chinese. Table 10 shows the baseline performance of ASR without LID that only generated character transcription and of ASR with LID that generated both character and language information sequences. In the case of $(\lambda_{Chr}, \lambda_{Lng}) = (1, 1)$, we

Table 7 ASR performances of a single CS language pair (natural speech; multi-speaker) machine speech chain in TER %. Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side is λ_{Lng} value during the unsupervised learning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$. The p-values compared to the baseline system for statistical significance are presented using ***, **, *, and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

Model	dev_{man}			dev_{sge}		
	mono	*CS*	*all*	*mono*	*CS*	*all*
Supervised learning: labeled mono						
Baseline	33.3	47.7	44.9	47.8	57.7	53.6
Semisupervised learning: labeled mono + unlabeled CS						
Label propagation	37.7	48.4	46.4	54.6	59.2	57.3
Proposed speech chain	38.6	37.4 ***	37.6 ***	52.9	47.1 ***	49.5 ***
+LID (0.25→0.0)	34.0	32.5 ***	32.8 ***	48.8	42.3 ***	45.0 ***
+LID (0.25→0.1)	35.1	33.7 ***	33.9 ***	50.0	42.8 ***	45.8 ***
Semisupervised learning: labeled mono + labeled CS + unlabeled CS						
Label propagation	34.5	45.4 ***	43.3 ***	50.7	54.5 ***	52.9
Supervised learning: labeled mono + labeled CS						
Topline	34.4	28.6 ***	29.7 ***	51.4	39.1 ***	44.2 ***

Table 8 ASR performances of a single CS language pair (natural speech; multi-speaker) machine speech chain in CER %. Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side is λ_{Lng} value during the unsupervised learning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$. The p-values compared to the baseline system for statistical significance are presented using ***, **, * and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

Model	dev_{man}			dev_{sge}		
	mono	*CS*	*all*	*mono*	*CS*	*all*
Supervised learning: labeled mono						
Baseline	32.4	56.9	52.4	34.9	59.3	47.0
Semisupervised learning: labeled mono + unlabeled CS						
Label propagation	36.4	55.5 ***	52.0	40.9	59.3	50.1
Proposed speech chain	39.8	39.6 ***	39.6 ***	39.1	44.4 ***	41.7 ***
+LID (0.25→0.0)	34.0	33.3 ***	33.5 ***	35.2	38.3 ***	36.7 ***
+LID (0.25→0.1)	35.5	34.6 ***	34.7 ***	36.3	39.0 ***	37.6 ***
Semisupervised learning: labeled mono + labeled CS + unlabeled CS						
Label propagation	33.4	53.1 ***	49.5 ***	37.0	55.4 ***	46.1 *
Supervised learning: labeled mono + labeled CS						
Topline	35.2	28.7 ***	29.9 ***	37.9	36.5 ***	37.2 ***

Table 9 TTS performances of a single CS language pair (natural speech; multi-speaker) machine speech chain in L2-norm squared. Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side is λ_{Lng} value during the unsupervised learning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$.

Model	dev_{man}			dev_{sge}		
	mono	*CS*	*all*	*mono*	*CS*	*all*
Supervised learning: labeled mono						
Baseline	.310	.613	.503	.315	.597	.426
Semisupervised learning: labeled mono + unlabeled CS						
Label propagation	.331	.630	.512	.326	.618	.433
Proposed speech chain	.289	.559	.465	.297	.553	.397
+LID (0.25→0.0)	.277	.549	.456	.285	.542	.390
+LID (0.25→0.1)	.273	.543	.451	.281	.536	.385
Semisupervised learning: labeled mono + labeled CS + unlabeled CS						
Label propagation	.292	.592	.489	.314	.592	.416
Supervised learning: labeled mono + labeled CS						
Topline	.276	.529	.441	.284	.522	.379

Table 10 Comparison performance (in CER%) between ASR baselines with/without LID. The p-values compared to the baseline system for statistical significance are presented using ***, **, * and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

Train: Ja25k+En25k+Zh25k	(λ_{Chr},λ_{Lng})	**Ja**	**En**	**Zh**
ASR without LID [chr]	No LID	8.8	9.1	5.8
ASR with LID [chr,lng]	(1,1)	8.9	8.5	5.1
ASR with LID [chr,lng]	(0.75,0.25)	7.3 ***	7.3 ***	5.1 *

found there was no statistically significant difference from ASR without LID in any of the tests. However, in the case of $(\lambda_{Chr}, \lambda_{Lng}) = (0.75, 0.25)$, which are the λ values during the supervised learning process of the best model on a single CS (Table 5), the results raised the possibility that the architecture with LID could assist the ASR performance.

Next, we investigate how our proposed approach performed on multiple CS language pairs, including the unknown CS excluded from the training data.

We individually trained ASR and TTS using labeled monolingual Ja25k, En25k, and Zh25k and carried out a machine speech chain on the following three different scenarios: (1) **EnJaCS10k+JaZhCS10k (unlabeled)**: semisupervised learning with unlabeled EnJaCS 10k and JaZhCS 10k (ZhEnCS for a zero-shot target); (2) **EnJaCS10k+ZhEnCS10k (unlabeled)**: semisupervised learning with unlabeled EnJaCS 10k and ZhEnCS 10k (JaZhCS for a zero-shot target); (3) **EnZhCS10k+ZhJaCS10k (unlabeled)**: semisupervised learning with unlabeled EnZhCS 10k and ZhJaCS 10k (EnJaCS for a zero-shot target).

There are four types for LID: (1) the "No LID" systems without using language information; (2) "LngChr" systems, which output the language information with character together like (Jp-a, Jp-b, Jp-c, ..., Jp-z, En-a, En-b, En-c, ..., En-z, Zh-a, Zh-b, Zh-c, ..., Zh-z); (3) LID (0.25→0.0), where 0.25 is λ_{Lng} value during supervised learning process and 0.0 is λ_{Lng} value during unsupervised learning process; (4) LID (0.25→0.1), where 0.25 is λ_{Lng} value during supervised learning process and 0.1 is λ_{Lng} value during unsupervised learning process.

As Table 11 shows, compared to the baseline model, our proposed model of any language pairs improved the ASR performance on all CS, including zero-shot CS. Compared to the No LID and LngChr, the LID (0.25→0.0) overcame them on all CS test sets. All of the machine speech chain models showed statistical significance with $p < .001$.

We also investigated whether our proposed machine speech chain improved the ASR performance on multiple CS language pairs, including the natural speech CS. We applied the last model among the trained models of 60 epochs since we could not include the natural speech CS in the development sets. Since natural CS may switch multiple times within a single utterance, it tends to be more complicated than the synthetic one. Besides, the natural CS was just only 1k, which is insufficient for training. As shown in Table 12, the performances were not as good as only the synthetic data. However, our proposed machine speech chain model improved ASR, showing statistical significance in the multiple CS language pairs, including the natural

Table 11 ASR performance in CER% of multiple CS language pairs machine speech chain (The bold figures show the unused CS during training). Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side of arrows is λ_{Lng} value during the unsupervised learning or fine-tuning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$. The p-values compared to the baseline system for statistical significance are presented using ***, **, * and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

		Monolingual			Code-switching		
	LID type	**Ja**	**En**	**Zh**	**EnJaCS**	**JaZhCS**	**ZhEnCS**
[Baseline] Supervised learning of only labeled monolingual data							
Ja25k+En25k+Zh25k (labeled)	No LID	8.8	9.1	5.8	**11.5**	**12.3**	**13.3**
	LID (0.25)	7.3 ***	7.3 ***	5.1 *	**10.1**	**11.2** **	**11.4** ***
[Machine Speech Chain] semisupervised learning of unlabeled two CS data							
+ EnJaCS10k+JaZhCS10k (unlabeled)	No LID	8.8	9.7	5.9	8.2 ***	6.9 ***	**7.7** ***
	LngChr	8.9	9.2	5.4	7.9 ***	7.2 ***	**7.2** ***
	LID (0.25→0.0)	8.3	7.6	5.2	7.7 ***	6.7 ***	**7.1** ***
	LID (0.25→0.1)	8.6	8.4	5.1	8.6 ***	6.9 ***	**7.4** ***
+ EnJaCS10k+ZhEnCS10k (unlabeled)	No LID	8.9	9.9	5.9	8.5 ***	**7.0** ***	7.5 ***
	LngChr	9.1	9.3	5.6	8.3 ***	**7.1** ***	7.4 ***
	LID (0.25→0.0)	8.5	7.4	5.7	7.8 ***	**6.8** ***	7.1 ***
	LID (0.25→0.1)	8.7	8.8	5.3	8.1 ***	**7.4** ***	7.2 ***
+ ZhEnCS10k+JaZhCS10k (unlabeled)	No LID	9.0	10.2	5.9	**8.6** ***	7.0 ***	7.6 ***
	LngChr	9.0	9.4	5.5	**8.3** ***	6.9 ***	7.4 ***
	LID (0.25→0.0)	8.5	7.5	5.2	**7.8** ***	6.8 ***	6.9 ***
	LID (0.25→0.1)	8.8	8.8	5.2	**8.6** ***	7.0 ***	7.7 ***
[Topline] Supervised learning of labeled two or three CS data							
+ EnJaCS10k+JaZhCS10k (labeled)	LID (0.25→0.0)	8.4	8.5	7.9	7.8 ***	6.4 ***	**6.8** ***
+ EnJaCS10k+ZhEnCS10k (labeled)	LID (0.25→0.0)	8.3	8.0	7.2	7.7 ***	**6.5** ***	6.6 ***
+ ZhEnCS10k+JaZhCS10k (labeled)	LID (0.25→0.0)	9.3	9.4	5.2	**7.8** ***	6.6 ***	6.7 ***
+ EnJaCS10k+JaZhCS10k+ZhEnCS10k (labeled)	LID (0.25→0.0)	8.1	8.1	7.0	7.6 ***	6.4 ***	6.6 ***

Table 12 ASR performance in CER% of multiple CS language pairs machine speech chain using natural CS (The bold figures show the unused CS during training). Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side of arrows is λ_{Lng} value during the unsupervised learning or fine-tuning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$. The p-values compared to the baseline system for statistical significance are presented using ***, **, * and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

		Monolingual			Code-switching			
	LID type	**Ja**	**En**	**Zh**	**EnJaCS**	**JaZhCS**	**ZhEnCS**	**TstNatEnJaCS**
[Baseline] Supervised learning of labeled monolingual data								
Ja25k+En25k+Zh25k plus	No LID (0.25)	8.8	9.7	5.4	**11.5**	**13.4**	**14.0**	**33.0**
NatJa0.2k+NatEn0.2k (labeled)	LID	8.6	7.7	5.1	**10.9**	**11.0** ***	**11.0** ***	**31.8**
[Machine Speech Chain] Semisupervised learning of unlabeled two CS and one natural CS data								
+ EnJaCS10k+JaZhCS10k plus	No LID	9.2	12.1	5.4	9.3 ***	7.7 ***	**8.7** ***	14.2 ***
NatEnJaCS0.7K (unlabeled)	LngChr	9.2	11.0	6.3	9.0 ***	8.0 ***	**9.6** ***	14.0 ***
	LID(0.25→0.0)	8.8	10.1	5.6	8.7 ***	7.2 ***	**7.9** ***	11.8 ***
	LID(0.25→0.1)	9.3	11.0	5.3	9.4 ***	9.1 ***	**8.5** ***	13.7 ***
[Topline] Supervised learning of labeled two CS and one natural CS data								
+ EnJaCS10k+JaZhCS10k plus NatEnJaCS0.7K (labeled)	LID (0.25→0.0)	8.7	9.0	7.1	7.6 ***	6.8 ***	**6.9** ***	9.6 ***

speech CS.

We also investigated French and Chinese CS (FrZhCS) performance with French as an unknown language. Since the system has never been trained with French data in supervised learning, it did not have a chance to learn the relation between French speech and the corresponding transcription. LID did not have an opportunity to identify the French language. Table 13 shows the ASR performance. The results reveal that the proposed model still improved the ASR performance on the FrZhCS test data even though no monolingual French labeled data are available, and even though the French language is unknown. Amazingly, when the amount of EnFrCS training data increased to 10k, the topline results worsened. The topline model was trained in supervised learning, but the target FrZhCS was never trained. This condition might degrade the supervised learning performance because of the mismatch as the non-target CS training data increased. The LngChr model become worsen by the speech chain training, showing it was difficult to handle the unknown language. However, our proposed model improved the performance even in an unknown language, zero-shot CS.

TTS Evaluation

We evaluated the zero-shot CS speech generated by the TTS of the proposed multilingual and zero-shot CS machine speech chain. We conducted an AB prefer-

Table 13 ASR performance (in CER%) of a multiple CS language pairs machine speech chain on the zero-shot CS with the unknown language, where labeled monolingual French data are unavailable, and the French language is unknown. Left side of arrows in the LID is λ_{Lng} value during the supervised learning process, and right side of arrows is λ_{Lng} value during the unsupervised learning or fine-tuning process, where $\lambda_{Chr} = 1 - \lambda_{Lng}$. The p-values compared to the baseline system for statistical significance are presented using ***, **, * and no-star ($^{***}p < .001$, $^{**}p < .01$, $^{*}p < .05$).

Train data	LID type	**FrZhCS**
Baseline		
Ja25k+En25k+Zh25k (labeled)	No LID	33.7
	LID (0.25)	33.0
Machine Speech Chain		
+EnJaCS10k+JaZhCS10k+ EnFrCS5k (unlabeled)	No LID	23.6 ***
	LngChr	48.0
	LID (0.25→0.0)	23.6 ***
	LID (0.25→0.1)	22.1 ***
+EnJaCS10k+JaZhCS10k+ EnFrCS10k (unlabeled)	No LID	24.0 ***
	LngChr	44.8
	LID (0.25→0.0)	22.4 ***
	LID (0.25→0.1)	21.6 ***
Topline		
+EnJaCS10k+JaZhCS10k+ EnFrCS5k (labeled)	LID (0.25→0.0)	14.5 ***
+EnJaCS10k+JaZhCS10k+ EnFrCS10k (labeled)	LID (0.25→0.0)	16.4 ***

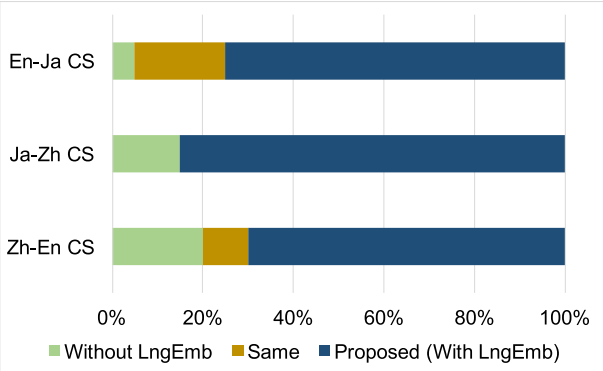


Fig. 13 Comparison of AB preference subjective evaluation between generated zero-shot CS speech from the model with/without language-embedding.

ence subjective evaluation between the generated zero-shot CS speech from the model with/without language-embedding [y^{CSChr}, y^{CSLng}]. All language pairs of the zero-shot CS were evaluated by ten bilingual speakers who compared two speech utterances while looking at the transcription and chose which speech was better in terms of being more native. They were also given the option to admit they could not determine which sounded more native. They compared 20 pairs shown randomly. The results (Fig. 13) show our method supports the quality of synthesized speech, particularly on the switching places between two languages.

8. Conclusion

We introduced a machine speech chain for semisupervised learning of CS ASR and TTS. We first individually trained ASR and TTS systems with labeled monolingual data in supervised learning. Then, we carried out a machine speech chain with unsupervised learning of either CS text or CS speech. We investigated the improvements to CS ASR and TTS with natural CS data as well as synthetic data. Our results revealed that such a mutually complementary architecture of machine speech chain trains ASR and TTS together and improves performance even without any labeled CS data. Our proposed machine speech chain model improved the performance of the CS ASR and CS TTS while maintaining the performance of the monolingual input.

We also introduced a language-aware CS machine speech chain. We expanded our model to handle CS better by integrating language embedding and LID into the machine speech chain. We confirmed that the machine speech chain model with language embedding and LID could produce satisfactory performances both on a single CS language pair and multiple CS language pairs, including unknown CS that were excluded from the training data.

Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP21H03467.

References

[1] Japanese Ministry of Health, Labour and Welfare, "Overview of the population statistics in 2017 [in Japanese]." https://www.mhlw.go.jp/toukei/saikin/hw/jinkou/kakutei17/xls/29toukei.xls, 2017.
[2] S. Poplack, Code switching: Linguistic, pp.2062–2065, Elsevier, 2001.
[3] M. Nakamura, "Developing codeswitching patterns of a Japanese/English bilingual child," Proc. ISB4, Somerville, MA, USA, pp.1679–1689, 2005.
[4] S.S. Fotos, "Japanese-English code switching in bilingual children," JALT Journal, vol.12, no.1, pp.75–98, 1990.
[5] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," Proc. ASRU, Okinawa, Japan, pp.301–308, IEEE, 2017.
[6] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain with one-shot speaker adaptation," Proc. INTERSPEECH, Hyderabad, India, pp.887–891, IEEE, 2018.
[7] S. Nakayama, A. Tjandra, S. Sakti, and S. Nakamura, "Speech chain for semi-supervised learning of Japanese-English code-switching ASR and TTS," Proc. SLT, Athens, Greece, IEEE, 2018.
[8] S. Nakayama, A. Tjandra, S. Sakti, and S. Nakamura, "Zero-shot code-switching ASR and TTS with multilingual machine speech chain," Proc. ASRU, Sentosa, Singapore, IEEE, 2019.
[9] D.C. Lyu, T.P. Tan, E.S. Chng, and H. Li, "SEAME: a Mandarin-English code-switching speech corpus in South-East Asia," Proc. INTERSPEECH, pp.1986–1989, 2010.
[10] H. Xu, S. Ding, and S. Watanabe, "Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling," Proc. ICASSP, pp.7110–7114, IEEE, 2019.
[11] N.T. Vu, D.-C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, "A first speech recognition system for Mandarin-English code-switch conversational speech," Proc. ICASSP, Kyoto, Japan, pp.4889–4892, IEEE, 2012.
[12] N. Luo, D. Jiang, S. Zhao, C. Gong, W. Zou, and X. Li, "Towards end-to-end code-switching speech recognition," CoRR, vol.abs/1810.13091, 2018.

[13] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Investigating end-to-end speech recognition for Mandarin-English code-switching," Proc. ICASSP, pp.6056–6060, IEEE, 2019.
[14] B.H. Ahmed and T.-P. Tan, "Automatic speech recognition of code switching speech using 1-best rescoring," Proc. IALP, Hanoi, Vietnam, pp.137–140, 2012.
[15] E. Yılmaz, H. van den Heuvel, and D. van Leeuwen, "Investigating bilingual deep neural networks for automatic recognition of code-switching frisian speech," Procedia Computer Science, vol.81, pp.159–166, 2016.
[16] P. Guo, H. Xu, L. Xie, and E.S. Chng, "Study of semi-supervised approaches to improving English-Mandarin code-switching speech recognition," Proc. INTERSPEECH, pp.1928–1932, ISCA, Sept. 2018.
[17] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan-a bilingual TTS system," Proc. ICASSP, Hong Kong, China, pp.264–267, IEEE, 2003.
[18] H. Liang, Y. Qian, and F.K. Soong, "An HMM-based bilingual (Mandarin-English) TTS," Proc. ISCA SSW6, Bonn, Germany, pp.137–142, 2007.
[19] S. Sitaram and A.W. Black, "Speech synthesis of code-mixed text," Proc. LREC, Miyazaki, Japan, pp.3422–3428, 2016.
[20] S. Sitaram, S. Rallabandi, S. Rijhwani, and A.W. Black, "Experiments with cross-lingual systems for synthesis of code-mixed text," Proc. ISCA SSW9, Sunnyvale, CA, USA, 2016.
[21] S. Rallabandi and A.W. Black, "On building mixed lingual speech synthesis systems," pp.52–56, ISCA, Aug. 2017.
[22] C.M. White, S. Khudanpur, and J.K. Baker, "An investigation of acoustic models for multilingual code switching," Proc. INTERSPEECH, Brisbane, Australia, pp.2691–2694, 2008.
[23] D. Imseng, H. Bourlard, M. Magimai-Doss, and J. Dines, "Language dependent universal phoneme posterior estimation for mixed language speech recognition," Proc. ICASSP, Prague, Czech Republic, pp.5012–5015, 2011.
[24] H. Seki, S. Watanabe, T. Hori, J.L. Roux, and J.R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," Proc. ICASSP, Calgary, Canada, IEEE, 2018.
[25] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," Proc. AAAI, p.3, 2008.
[26] M. Johnson, M. Schuster, Q.V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," Transactions of the Association for Computational Linguistics, vol.5, pp.339–351, Dec. 2017.
[27] Y. Jia, M. Johnson, W. Macherey, R.J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," Proc. ICASSP, pp.7180–7184, IEEE, 2019.
[28] M. Hasegawa-Johnson, A. Black, L. Ondel, O. Scharenborg, and F. Ciannella, "Image2speech: Automatically generating audio descriptions of images," Proc. ICNLSSP, vol.1, no.1, pp.19–27, 2017.
[29] W. Havard, L. Besacier, and O. Rosec, "Speech-coco: 600k visually grounded spoken captions aligned to mscoco data set," ISCA Workshop on Grounding Language Understanding (GLU2017), 2017.
[30] G.I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switched language models using neural based synthetic data from parallel sentences," Proc. CoNLL, pp.271–280, IEEE, 2019.
[31] Y. Sharma, B. Abraham, K. Taneja, and P. Jyothi, "Improving low resource code-switched asr using augmented code-switched TTS," Proc. INTERSPEECH, pp.4771–4775, IEEE, 2020.
[32] Z. Tan, X. Fan, H. Zhu, and E. Lin, "Addressing accent mismatch in Mandarin-English code-switching speech recognition," Proc. ICASSP, pp.8259–8263, 2020.
[33] P.B. Denes and E.N. Pinson, The Speech Chain: The Physics And Biology Of Spoken Language, Anchor books, Worth Publishers, 1993.
[34] A. Tjandra, S. Sakti, and S. Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," Proc. ICASSP, Brighton, UK, pp.6281–6285, IEEE, 2019.
[35] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," Proc. ICASSP, pp.4945–4949, IEEE, 2016.
[36] W. Chan, N. Jaitly, Q.V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," Proc. ICASSP, Shanghai, China, pp.4960–4964, IEEE, 2016.
[37] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R.J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R.A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," Proc. INTERSPEECH, Stockholm, Sweden, pp.4006–4010, IEEE, 2017.
[38] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," CoRR, vol.abs/1705.02304, 2017.
[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Proc. ICLR, pp.1–15, San Diego, CA, USA, 2015.
[40] T. Luong, H. Pham, and C.D. Manning, "Effective approaches to attention-based neural machine translation," Proc. EMNLP, Lisbon, Portugal, pp.1412–1421, Association for Computational Linguistics, Sept. 2015.
[41] Y. Wang, R.J. Skerry-Ryan, D. Stanton, Y. Wu, R.J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q.V. Le, Y. Agiomyrgiannakis, R. Clark, and R.A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," CoRR, vol.abs/1703.10135, 2017.
[42] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," CoRR, 2015.
[43] T. Takezawa, G. Kikui, M. Mizushima, and E. Sumita, "Multilingual spoken language corpus development for communication research," The Association for Computational Linguistics and Chinese Language Processing, vol.12, no.3, pp.303–324, 2007.
[44] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," Proc. ISCA EUROSPEECH, Geneva, Switzerland, pp.381–384, 2003.
[45] S. Nakayama, T. Kano, Q.T. Do, S. Sakti, and S. Nakamura, "Japanese-English code-switching speech data construction," Proc. O-COCOSDA, Miyazaki, Japan, IEEE, 2018.
[46] P.N. Durette, "gTTS – Google Text-to-Speech." https://pypi.org/project/gTTS/.
[47] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," Proc. ICASSP, pp.5206–5210, 2015.
[48] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline," Proc. O-COCOSDA, pp.1–5, 2017.
[49] Z. Zeng, Y. Khassanov, V.T. Pham, H. Xu, E.S. Chng, and H. Li, "On the end-to-end solution to Mandarin-English code-switching speech recognition," Proc. INTERSPEECH, pp.2165–2169, 2019.
[50] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," Proc. INTERSPEECH, pp.3586–3589, 2015.
[51] T. Ko, V. Peddinti, D. Povey, M.L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," Proc. ICASSP, pp.5220–5224, IEEE, 2017.
[52] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto, et al., "librosa 0.5.0," https://librosa.github.io/librosa/0.5.0/index.html, 2017.
[53] T. Kudo, "Mecab: Yet another part-of-speech and morphological analyzer," http://taku910.github.io/mecab, 2006.
[54] H. Miura, "pykakasi – kakasi library in python." https://pypi.org/project/pykakasi/.
[55] H. Huang, "pypinyin – pinyin library in python." https://pypi.org/project/pypinyin/.

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," Proc. NIPS Autodiff Workshop, 2017.
[57] D.S. Pallet, W.M. Fisher, and J.G. Fiscus, "Tools for the analysis of benchmark speech recognition tests," Proc. ICASSP, pp.97–100, IEEE, 1990.
[58] P. Gage, "A new algorithm for data compression," C Users Journal, vol.12, no.2, pp.23–38, 1994.
[59] S. Nakayama, "Speech chain for semi-supervised learning of Japanese-English code-switching ASR," Master's thesis, Nara Institute of Science and Technology, Japan, 2019.

Sahoko Nakayama received a B.A. degree from Waseda University, Tokyo, Japan in 2012 and an M.S. degree in 2019 from Nara Institute of Science and Technology, Japan. She is currently pursuing a Ph.D. degree at the Nara Institute of Science and Technology, Japan. She is also working as a Research Assistant at RIKEN, Center for Advanced Intelligence Project AIP, Japan. Her research interests include machine learning (deep learning), speech recognition, code-switching, speech synthesis, and natural language processing.

Andros Tjandra received his B.Sc degree (cum laude) and his M.Sc degree (cum laude) from the Faculty of Computer Science, Universitas Indonesia, Indonesia, in 2014 and 2015, respectively. Later on, he received his Ph.D. degree from the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan in 2020. He is currently a research scientist in Facebook AI, USA. Previously, he was a research scientist intern at Google Brain and Facebook AI. His research interests include machine learning, speech recognition, speech synthesis, and natural language processing.

Sakriani Sakti is a research associate professor at the Augmented Human Communication Laboratory, NAIST, Japan, as well as a research scientist at RIKEN, Center for Advanced Intelligent Project AIP, Japan. She received her B.E. degree in Informatics (cum laude) from Bandung Institute of Technology, Indonesia, in 1999. In 2000, she received DAAD-Siemens Program Asia 21st Century Award to study in Communication Technology, University of Ulm, Germany, and received her MSc degree in 2002. During her thesis work, she also worked with Speech Understanding Department, Daimler Chrysler Research Center, Ulm, Germany. Between 2003-2009, she worked as a researcher at ATR SLC Labs, Japan, and during 2006-2011, she worked as an expert researcher at NICT SLC Groups, Japan. While working with ATR-NICT, Japan, she continued her study (2005-2008) at University of Ulm, Germany, and received her PhD degree in 2008. She was actively involved in collaboration activities such as Asian Pacific Telecommunity Project (2003-2007), ASTAR and U-STAR (2006-2011). In 2009-2011, she served as a visiting professor of Computer Science Department, University of Indonesia (UI), Indonesia. From 2011, she has been an assistant professor at the Augmented Human Communication Laboratory, NAIST, Japan. She served also as a visiting scientific researcher of INRIA Paris-Rocquencourt, France, in 2015-2016, under "JSPS Strategic Young Researcher Overseas Visits Program for Accelerating Brain Circulation". In 2011-2017, she served as an assistant professor at the Augmented Human Communication Laboratory, NAIST, Japan. Now she is a research associate professor at the Augmented Human Communication Laboratory, NAIST, Japan, as well as a research scientist at RIKEN, Center for Advanced Intelligent Project AIP, Japan. She is a member of JNS, SFN, ASJ, ISCA, IEICE and IEEE.

Satoshi Nakamura is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, Honorarprofessor of Karlsruhe Institute of Technology, Germany, and ATR Fellow. He received his B.S. from Kyoto Institute of Technology in 1981 and Ph.D. from Kyoto University in 1992. He was Associate Professor of Graduate School of Information Science at Nara Institute of Science and Technology in 1994-2000. He was Director of ATR Spoken Language Communication Research Laboratories in 2000-2008 and Vice president of ATR in 2007-2008. He was Director General of Keihanna Research Laboratories and the Executive Director of Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan in 2009-2010. He is currently Director of Augmented Human Communication laboratory and a full professor of Graduate School of Information Science at Nara Institute of Science and Technology. He is interested in modeling and systems of speech-to-speech translation and speech recognition. He is one of the leaders of speech-to-speech translation research and has been serving for various speech-to-speech translation research projects in the world including C-STAR, IWSLT and A-STAR. He received Yamashita Research Award, Kiyasu Award from the Information Processing Society of Japan, Telecom System Award, AAMT Nagao Award, Docomo Mobile Science Award in 2007, ASJ Award for Distinguished Achievements in Acoustics. He received the Commendation for Science and Technology by the Minister of Education, Science and Technology, and the Commendation for Science and Technology by the Minister of Internal Affairs and Communications. He also received LREC Antonio Zampoli Award 2012. He has been Elected Board Member of International Speech Communication Association, ISCA, since June 2011, IEEE Signal Processing Magazine Editorial Board Member since April 2012, IEEE SPS Speech and Language Technical Committee Member since 2013, and IEEE Fellow since 2016.