SPEECH CHAIN FOR SEMI-SUPERVISED LEARNING OF JAPANESE-ENGLISH CODE-SWITCHING ASR AND TTS

Sahoko Nakayama[1], *Andros Tjandra*[1,2], *Sakriani Sakti*[1,2], *Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan
{nakayama.sahoko.nq1,andros.tjandra.ai6,ssakti,s-nakamura}@is.naist.jp

ABSTRACT

Code-switching (CS) speech, in which speakers alternate between two or more languages in the same utterance, often occurs in multilingual communities. Such a phenomenon poses challenges for spoken language technologies: automatic speech recognition (ASR) and text-to-speech synthesis (TTS), since the systems need to be able to handle the input in a multilingual setting. We may find code-switching text or code-switching speech in social media, but parallel speech and the transcriptions of code-switching data, which are suitable for training ASR and TTS, are generally unavailable. In this paper, we utilize a speech chain framework based on deep learning to enable ASR and TTS to learn code-switching in a semi-supervised fashion. We base our system on Japanese-English conversational speech. We first separately train the ASR and TTS systems with parallel speech-text of monolingual data (supervised learning) and perform a speech chain with only code-switching text or code-switching speech (unsupervised learning). Experimental results reveal that such closed-loop architecture allows ASR and TTS to learn from each other and improve the performance even without any parallel code-switching data.

Index Terms— Speech chain, semi-supervised learning, code-switching, ASR and TTS, Japanese and English languages

1. INTRODUCTION

The number of Japanese-English bilingual speakers continues to increase. One reason is that the number of children in Japan with at least one non-Japanese parent has risen gradually over the past 25 years [1]. Also the number of school-age children who have lived abroad was reported that more than doubled in 2015 [2]. The number of international travelers or residents in Japan is steadily increasing for reasons of tourism, education, or health. These changes are affecting how people communicate with each other. The phenomenon of Japanese-English code-switching is becoming more and more frequent.

Code-switching (CS), which refers to bilingual (or multilingual) speakers who mix two or more languages in discourse (often with no change of interlocutor or topic), is a hallmark of bilingual communities world-wide [3]. Nakamura [4] surveyed the code-switching of a Japanese child who lived in the United States and found that 179 switches occurred during total one hour conversation with his/her mother. Fotos investigated four hours of conversations of four bilingual children in Japan with at least one American parent and observed 153 code-switchings [5]. Both reports reveal that people actually use Japanese-English CS in everyday life. Since people may not always communicate in monolingual settings, spoken language technologies, i.e., ASR and TTS, must be developed that can handle the input in a multilingual fashion, not only Japanese or English but also Japanese-English CS.

Unfortunately, despite extensive studies of CS in bilingual communities, scant research has addressed the Japanese-English case. Moreover, the common way of developing spoken language technologies for code-switching relies on a supervised manner that requires a significant amount of CS data to train the models. Although it might still be possible to find a sufficient amount of only CS text or CS speech in social media, unfortunately, parallel speech and transcription of CS data are mostly unavailable that are suitable for training ASR and TTS. However, in contrast with human communication, many people who speak in CS languages did not learn it by a supervised training mechanism with a parallel speech and textbook. Although many language courses are available, no CS class is offered. This means that they develop strategies for speaking in CS languages by merely growing up in bilingual/multilingual environments and listening and speaking with other bilingual speakers. CS often happens unconsciously. No fundamental reason exists why spoken language technologies have to learn CS in a supervised manner.

In this paper, we utilize a speech chain framework based on deep learning [6, 7] to enable ASR and TTS to learn CS in a semi-supervised fashion. We base our system on Japanese-English conversational speech. We first separately train ASR and TTS systems with the parallel speech-text of monolingual Japanese and English data (supervised learning) that might resemble what students of multiple languages learn in school. After that, we perform a speech chain with only CS text or CS speech (unsupervised learning) that imitates how humans

simultaneously listen and speak in a CS context in a multilingual environment.

2. RELATED WORKS

CS has been studied for several decades. Most researchers agree that it plays a vital role in bilingualism and is more than a random phenomenon [8]. White et al. [9] investigated alternatives to the acoustics for multilingual CS model, and Imseng et al. [10] proposed an approach that estimates the universal phoneme posterior probabilities for mixed language speech recognition. Vu et al. focused on speech recognition of Chinese and English CS [11]. They proposed approaches for phone merging in combination with discriminative training as well as the integration of a language identification system into the decoding process. Ahmed et al. proposed the automatic recognition of English-Malay CS speech. Their framework first used parallel ASR in both languages and subsequently joined and rescored the resulting lattices to estimate the most probable word sequence of English-Malay CS [12]. Recently, Yilmaz et al. investigated the impact of bilingual hidden markov model - deep neural networks (HMM/DNN) in Frisian and Dutch CS contexts [13]. Toshinwal et al. attempted to construct multilingual speech recognition with a single end-to-end model [14]. Although the model provided an effective way for a multilingual setting, it was found that the model was still unable to code-switch between languages, indicating that the language model is dominating the acoustic model.

In synthesis system researches, Chu et al. [15] constructed Microsoft Mulan, a bilingual Mandarin-English TTS system. Liang et al. also focused on Mandarin-English languages and proposed context-dependent HMM state sharing for their code-switched TTS system [16]. Sitaram et al. performed TTS experiments on code-mixed Hindi and English written in Romanized script and German and English written in their native scripts [17, 18]. SaiKrishna et al. investigated approaches to build mixed-lingual speech synthesis systems of Hindi-English, Telugu-English, Marathi-English, and Tamil-English, based on separate recordings [19].

Despite extensive studies on CS spoken language technologies in bilingual communities, the Japanese-English case has received scant research up to now. Until recently, no research work has addressed Japanese-English CS. Seki et al. developed the speech recognition of mixed language speech including the Japanese-English case with hybrid attention/CTC [20]. However, they created data that used different speakers for different languages, where the main challenge in the CS phenomenon in which the same speakers alternate between two or more languages within sentences is not addressed.

Most existing approaches, developed for bilingual CS, either mainly focused on supervised learning with CS data only for ASR or only for TTS. Furthermore, the study of Japanese-English CS is still very limited. In contrast, our study constructs sequence-to-sequence models for both Japanese-English CS ASR and TTS that are jointly trained through a loop connection. The overall closed-loop speech chain framework enables ASR and TTS to teach each other and learn CS in a semi-supervised fashion without parallel CS data.

3. JAPANESE-ENGLISH CODE-SWITCHING

CS phenomena can basically be classified into two primary categories: inter-sentential and intra-sentential. In inter-sentential CS, the language switch is done at the sentence boundaries. In intra-sentential CS, the shift is done in the middle of a sentence. However, the units and the locations of the switches in intra-sentential CS may vary widely from single word switches to whole phrases (beyond the length of standard loanword units). Below are examples of actual Japanese-English CS [4]:

- **Intra-sentential code-switching:**
  - **[Word-level code-switching]:** "Trust-`shiteru hito ni dake kashite-ageru no`." (*I only lend (it) to a person I trust.*)
  - **[Phrase-level code-switching]:** "`Kondo no doyoubi no yuugata, ohima deshitara` please come to our house for a Japanese dinner." (*If you are free this Saturday evening, please come to our house for a Japanese-style dinner.*)
- **Inter-sentential code-switching:**
  - **[Inter-sentential code-switching]:** "`Aa, soo datte nee.` On the honeymoon, they bought this." (*Oh, year, you're right. On their honeymoon, they bought this.*)

However, some CS cases remain problematic. For example, loanwords cannot be called intra-sentential word-level CS, and quotations may not be intra-sentential phrase-level CS. Although they might not theoretically be CS, we also handle such cases within a CS framework because we aim to recognize every word in Japanese-English conversations.

4. SPEECH CHAIN FOR SEMI-SUPERVISED LEARNING OF CODE-SWITCHING

We previously designed and constructed a machine speech chain based on deep learning at our laboratory [6, 7], inspired by a human speech chain [21]. Humans learn how to speak by constantly repeating their articulations and listening to the produced sounds. By simultaneously listening and speaking, a speaker can monitor her volume, articulation, and her speech's general comprehensibility. Therefore, a closed-loop speech chain mechanism with auditory feedback from the speaker's mouth to her ear is crucial.

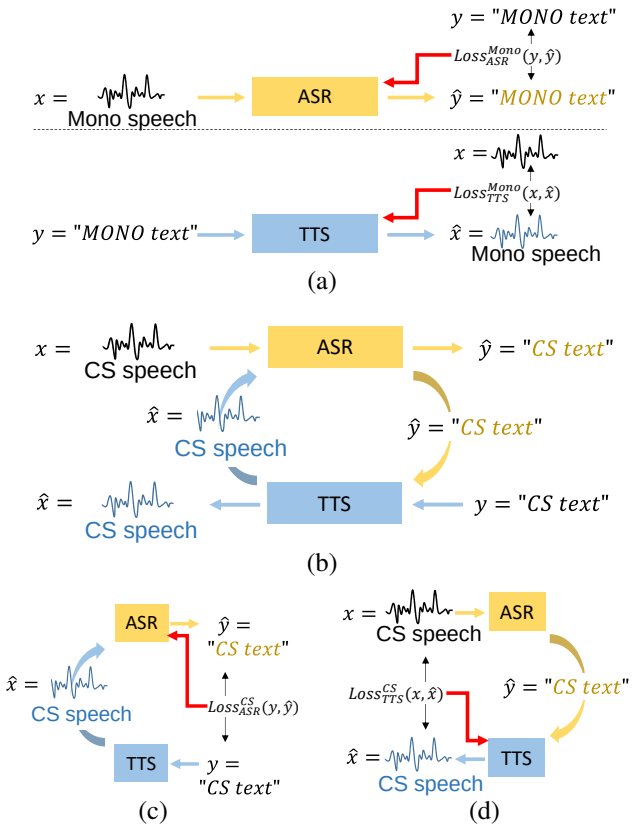

Fig. 1. Overview of proposed framework: (a) Train ASR and TTS separately with parallel speech-text monolingual data (supervised learning); (b) Train ASR and TTS simultaneously through speech chain with unparallel CS data (unsupervised learning); (c) Unrolled process from TTS to ASR given only CS text; (d) Unrolled process from ASR to TTS given only CS speech.

Over the past few decades, the development of ASR and TTS has enabled computers to either learn only how to listen through ASR or how to speak by a TTS. In contrast, a machine speech chain provides additional capability that enables computers not only to speak and listen but also to speak while listening. Its framework consists of a sequence-to-sequence ASR [22, 23] and a sequence-to-sequence TTS [24] as well as a loop connection between them. The closed-loop architecture allows us to train our model on the concatenation of both labeled and unlabeled data. While ASR transcribes the unlabeled speech features, TTS reconstructs the original speech waveform based on the ASR text. In the opposite direction, ASR also attempts to reconstruct the original text transcription given the synthesized speech.

Our CS ASR and TTS systems were built upon a speech chain framework (Fig. 1) with the following learning process:

1. **Train ASR and TTS separately with parallel speech-text monolingual data (supervised learning)**
   We first separately train the ASR and TTS systems with parallel speech-text of monolingual Japanese and English data (supervised learning) that might resemble humans who learn multiple languages at school (Fig. 1(a)). Given a speech and text pair of monolingual data (x^{Mono}, y^{Mono}) with speech length S and text length T, ASR generates text probability vector $\hat{y}^{Mono}$ with teacher-forcing using directly ground-truth samples (y^{Mono}) as decoder input, and loss $L^{Mono}_{ASR}(\hat{y}^{Mono}, y^{Mono})$ is calculated between output text probability vector $\hat{y}^{Mono}$ and reference text y^{Mono}. On the other hand, TTS also generates best predicted speech $\hat{x}^{Mono}$ by teacher-forcing using the reference (x^{Mono}), and loss $L^{Mono}_{TTS}(\hat{x}^{Mono}, x^{Mono})$ is calculated between predicted speech $\hat{x}^{Mono}$ and ground-truth speech x^{Mono}. The parameters are then updated with gradient descent optimization.

2. **Train ASR-TTS simultaneously in a speech chain with unparallel CS data (unsupervised learning)**
   After that, we then simultaneously train ASR and TTS through a speech chain with unparallel CS data (unsupervised learning) that imitate simultaneous human listening and speaking CS in a multilingual environment (Fig. 1(b)).

   To further clarify the learning process during unsupervised training, we unrolled the following architecture:

   (a) **Unrolled process from TTS to ASR given only CS text**
   Given CS text input y^{CS} only, TTS generates speech waveform $\hat{x}^{CS}$, while ASR also attempts to reconstruct original text transcription $\hat{y}^{CS}$, given the synthesized speech. Fig. 1(c) illustrates the mechanism. Here, we can also treat it as another autoencoder model, where the text-to-speech TTS serves as an encoder, and the speech-to-text ASR serves as a decoder. Then loss $L^{CS}_{ASR}(\hat{y}^{CS}, y^{CS})$ can be calculated between output text probability vector $\hat{y}^{CS}$ and input text y^{CS} to update the ASR parameters.

   (b) **Unrolled process from ASR to TTS given only CS speech**
   Given unlabeled CS speech features x^{CS}, ASR transcribes unlabeled input speech $\hat{y}^{CS}$, while TTS attempts to reconstruct original speech waveform $\hat{x}^{CS}$ based on the output text from ASR. Fig. 1(d) illustrates the mechanism. We can also treat it as an autoencoder model, where the speech-to-text ASR serves as an encoder, and the text-to-speech TTS serves as a decoder. Then loss $L^{CS}_{TTS}(\hat{x}^{CS}, x^{CS})$ can be calculated between reconstructed speech waveform $\hat{x}^{CS}$ and the input

of original speech waveform x^{CS} to update the TTS parameters.

Here, we can weigh all of the loss into a single loss variable by the following formula:

$$L = \alpha * (L_{ASR}^{Mono} + L_{TTS}^{Mono}) + \beta * (L_{ASR}^{CS} + L_{TTS}^{CS}) \quad (1)$$

$$\theta_{ASR} = Optim(\theta_{ASR}, \nabla_{\theta_{ASR}} L) \quad (2)$$

$$\theta_{TTS} = Optim(\theta_{TTS}, \nabla_{\theta_{TTS}} L), \quad (3)$$

where α and β are hyperparameters to scale the loss between the supervised (parallel) and unsupervised (unparallel) loss. This idea allows us to train new matters without forgetting the old ones. If $\alpha > 0$, we can keep using some portions of the loss and the gradient provided by the paired training set; if $\alpha = 0$, we completely learn new matters with only CS speech or only CS text.

5. EXPERIMENTS

5.1. Monolingual and Code-Switching Corpora

We utilized the monolingual Japanese and English ATR Basic Travel Expression Corpus (BTEC) [25, 26], which covers basic conversations in the travel domain, such as sightseeing, restaurants, hotels, etc. The sentences were collected by bilingual travel experts from Japanese/English sentence pairs in travel domain phrasebooks. We randomly selected 50k sentences for training, 500 sentences for the development set, and 500 sentences for a test set from BTEC1-4.

Since no large Japanese-English CS dataset exists yet, we constructed one from monolingual Japanese and English BTEC sentences. Here, we created two types of intra-sentential code-switching: word-level and phrase-level CS. An overview of the text data construction is illustrated in Fig. 2, and more details are also available [27].

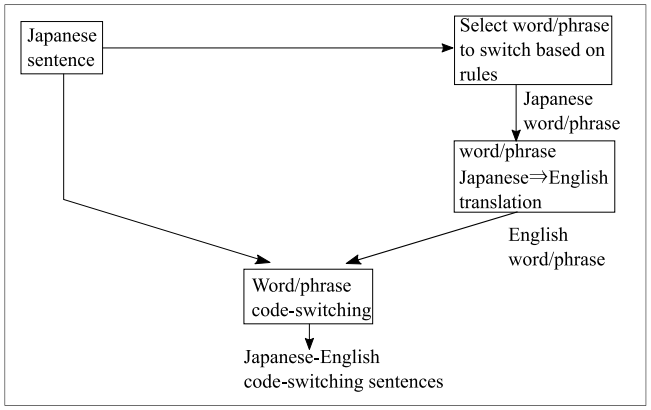

Fig. 2. Japanese-English CS text data construction.

Since collecting the natural speech of Japanese-English CS data from bilingual speakers requires much time and money, we also utilized Google TTS[1] to generate speech from the text corpora for all the text data, including monolingual Japanese, monolingual English, and Japanese-English CS.

5.2. Features Extraction

All raw speech waveforms are represented at a 16-kHz sampling rate. For the speech features, we used a log magnitude spectrogram extracted by short-time Fourier transform (STFT) from the Librosa library[2]. First, we applied wave-normalization (scaling raw wave signals into a range [-1, 1]) per utterance, followed by pre-emphasis (0.97), and extracted the spectrogram with an STFT, a 50-ms frame length, a 12.5-ms frame shift, and a 2048 point FFT. After we got the spectrogram, we took the squared magnitude and used a Mel-scale filterbank with 40 filters to extract the Mel-scale spectrogram. Next we got the Mel-spectrogram and the squared magnitude spectrogram features. In the end, we transformed all of the speech utterances into log-scale and normalized each feature into 0 mean and unit variances. Our final set included 40 dims log Mel-spectrogram features and 1025 dims log magnitude spectrograms.

For the English text, we converted all of the sentences into lowercase letters and removed all the punctuation marks [,:?.]. For the Japanese text, we applied a morphological analyzer Mecab[3] to extract the katakana characters and converted them into English letters using pykakasi[4]. We have 26 letters (a-z), one punctuation mark (-) for extending the sound of Japanese, and three special tags (<s>, </s>, <spc>) that denote the start and end of sentences and the spaces between words.

5.3. ASR and TTS Systems

Our ASR system is a standard encoder-decoder with an attention mechanism [22]. On the encoder side, we used a log-Mel spectrogram as the input features. The input features were projected by a fully connected layer and a LeakyReLU ($l = 1e - 2$) [28] activation function and processed by three stacked BiLSTM layers with 256 hidden units for each direction (total 512 hidden units). We applied sequence subsampling [29, 23] to the last two top layers and reduced the length of the speech features by a factor of 4. On the decoder side, the input characters were projected with a 128 dims embedding layer and fed into one layer LSTM with 512 hidden units. Then we calculated the attention matrix with an MLP scorer [30], followed by a fully connected layer and a softmax function. Both the ASR and TTS models were implemented with the PyTorch library[5].

[1] https://pypi.python.org/pypi/gTTS
[2] Librosa–https://librosa.github.io/librosa/0.5.0/index.html
[3] MeCab is a morphological analyzer–https://github.com/taku910/mecab
[4] Pykakasi–https://github.com/miurahr/pykakasi
[5] https://github.com/pytorch/pytorch

The TTS system is based on a sequence-to-sequence TTS (Tacotron) [24]. Its hyperparameters are almost the same as with the original Tacotron, except we generally used LeakyReLU instead of ReLU. On the encoder sides, CBHG used $K = 8$ different filter banks instead of 16 to reduce our GPU memory consumption. For the decoder sides, we used two stacked LSTMs instead of a GRU with 256 hidden units. Our TTS predicts four consecutive frames at one time step to reduce the number of time steps in the decoding process.

As described in Section 3, we first separately trained the ASR and TTS systems with parallel speech-text of monolingual Japanese and English data (supervised learning). After that, we performed a speech chain with only CS text or CS speech (unsupervised learning). For the α and β hyperparameters to scale the loss between the supervised (parallel) and unsupervised (unparallel) loss, we used the same $\alpha = 0.5$, $\beta = 1$ for most of our experiments.

6. EXPERIMENTAL RESULTS

We conducted our evaluation on four types of test sets: (1) **TstMonoJa (JaTTS)**: a Monolingual Japanese text and corresponding speech created by a Japanese TTS; (2) **TstCSWord+Phr (JaTTS)**: an intra-sentential word and phrase-level CS Japanese-English text and corresponding speech created by a Japanese TTS; (3) **TstCSWord+Phr (MixTTS)**: an intra-sentential word and phrase-level CS Japanese-English text and corresponding speech created by a mixed Japanese-English TTS; (4) **TstMonoEn (EnTTS)**:a Monolingual English text and corresponding speech created by an English TTS. Note that the combination of TstMonoJa (JaTTS) and TstMonoEn (EnTTS) can also be considered as inter-sentential code-switching test set. The ASR performance was evaluated by calculating the character error rate (CER), which is the edit distance between the reference data (ground-truth) and the system's hypothesis transcription. For the TTS evaluation, we calculated the difference in the L2-norm squared between the ground-truth and the predicted log-Mel spectrogram.

6.1. Baseline Systems

Figures 3 and 4 respectively show the performance of the baseline systems for ASR and TTS. The baseline systems were trained with supervised learning using a standard sequence-to-sequence ASR or a TTS framework without the speech chain framework. Eight types of baselines were evaluated: (1) **MonoJa50k (JaTTS)**: an ASR or TTS system trained with 50k monolingual Japanese text and corresponding speech created by a Japanese TTS; (2) **CSWord50k (JaTTS)** and (3) **CSPhr50k (JaTTS)**: ASR or TTS system trained with a 50k intra-sentential word or phrase-level CS Japanese-English text and corresponding speech created by a Japanese TTS; (4) **CSWord50k (MixTTS)** and (5) **CSPhr50k (MixTTS)**: ASR or TTS system trained with a 50k intra-sentential word or phrase-level CS Japanese-English text and corresponding speech created by the mixed of Japanese and English TTS; (6) **Ja25k+En25k (JaTTS)**: an ASR or TTS system trained with a 25k monolingual Japanese text plus a 25k monolingual English text and corresponding speech created by a Japanese TTS (inter-sentential CS); (7) **Ja25k+En25k (MixTTS)**: using the same text data as Ja25k+En25k (JaTTS) but with corresponding speech created by a Japanese TTS and a English TTS (inter-sentential CS); (8) **MonoEn50k (EnTTS)**: an ASR or TTS system trained with a 50k monolingual English text and corresponding speech created by an English TTS.

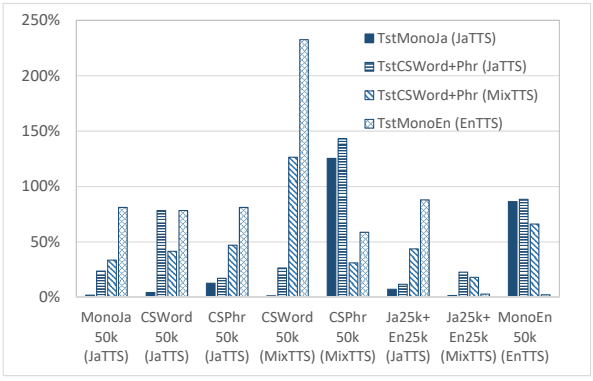

Fig. 3. Performances of ASR baseline in CER.

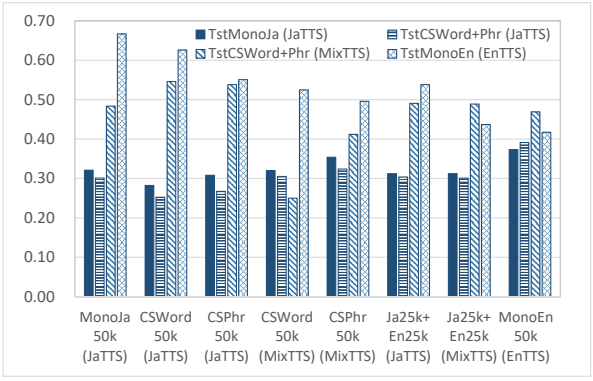

Fig. 4. Performances of TTS baseline in L2-norm squared of log-Mel spectrogram.

As seen in Fig. 3, the MonoJa50k (JaTTS) ASR system was good in the Japanese test set, but terrible in the English test set. On the other hand, the MonoEn50k (EnTTS) ASR system provided very low CER in the English test set, but a high CER in the Japanese test set. The difference between the MonoJa50k (JaTTS), CSWord50k (JaTTS), CSPhr50k (JaTTS), and Ja25k+En25k (JaTTS) systems is that MonoJa50k (JaTTS) only learn Japanese sentence, while the others learn several English words, phrases or sentences. However, since the training data of those systems are only generated by Japanese TTS, the CER in the English test set is still high. The TTS results also show the same tendency, although they are less extreme than the ASR case. The CSWord50k (MixTTS), CSPhr50k (MixTTS),

Table 1. ASR & TTS performances (in CER & L2-norm squared, respectively) of proposed CS speech chain framework.

	TstMonoJa (JaTTS)		TstCSWord+Phr (MixTTS)		TstMonoEn (EnTTS)	
	ASR	**TTS**	**ASR**	**TTS**	**ASR**	**TTS**
Baseline: paired speech-text Ja25k+En25k (MixTTS) → Supervised training						
Ja25k+En25k (MixTTS)	1.71%	0.312	18.11%	0.489	2.99%	0.437
Speech chain: paired Ja25k+En25k (MixTTS) + unpaired CSWord+Phr (JaTTS) → Semi-supervised training						
+CSWord+Phr10k (JaTTS)	1.85%	0.311	19.66%	0.484	4.79%	0.444
+CSWord+Phr20k (JaTTS)	1.85%	0.306	17.21%	0.489	4.65%	0.441
Speech chain: paired Ja25k+En25k (MixTTS) + unpaired CSWord+Phr (MixTTS) → Semi-supervised training						
+CSWord+Phr10k (MixTTS)	1.81%	0.312	5.35%	0.374	3.69%	0.437
+CSWord+Phr20k (MixTTS)	1.85%	0.310	5.54%	0.368	3.64%	0.440
Speech chain: paired Ja25k+En25k (MixTTS) + unpaired CSWord+Phr (Mix+JaTTS) → Semi-supervised training						
+CSWord+Phr20k (Ja+MixTTS)	1.82%	0.305	5.08%	0.372	4.05%	0.439

and Ja25k+En25k (MixTTS) were trained using Japanese text with Japanese TTS and English text with English TTS, but surprisingly only Ja25k+En25k (MixTTS) system that could handle the balance among Japanese, English, and Japanese-English CS languages. The CSWord50k (MixTTS), CSPhr50k (MixTTS) exceeded 100% CER as the number of errors that produced by the model was much larger than the number of character in the text references. This means that simply by mixing the languages will not solve the problems. Furthermore, because it was trained with the data that used different speakers for different languages switched within utterances, the TTS speech output still sounds like the mix between two speakers from two languages.

6.2. Proposed Systems

Our proposed system's objective is to enhance ASR and TTS to handle CS input (even without parallel CS data) while maintaining good performance in the monolingual setting. Here, we utilize the speech chain. We can use any available baseline, but since only the Ja25k+En25k (MixTTS) system provided quite reasonable performances given any inputs, we only report the speech chain results of the top Ja25k+En25k (MixTTS) system. Here, the TstCSWord+Phr (JaTTS) test set is discharged since it still consists with many English words that are incorrectly pronounced as Japanese words by Japanese TTS.

Table 1 shows the ASR-TTS performances (in CER and L2-norm squared) of the proposed CS speech chain framework from multiple scenarios: (1) **[paired Ja25k+En25k (MixTTS)]+[unpaired CSWord+Phr (JaTTS)]**: an ASR or TTS system trained in semi-supervised learning using monolingual Ja25k+En25k (MixTTS) as paired data and code-switching CSWord+Phr (JaTTS) as unpaired data; (2) **[paired Ja25k+En25k (MixTTS)]+[unpaired CS Word+Phr (Mix TTS)]**: an ASR or TTS system trained in semi-supervised learning using monolingual Ja25k+En25k (Mix TTS) as paired data and code-switching CSWord+Phr (Mix TTS) as unpaired data; (3) **[paired Ja25k+En25k (MixTTS)] +[unpaired CSWord+Phr (Mix+JaTTS)]**: an ASR or TTS system trained in semi-supervised learning using monolingual Ja25k+En25k (MixTTS) as paired data and both code-switching CSWord+Phr (MixTTS) and CSWord+Phr (Ja TTS) as unpaired data.

Using just unpaired CS data and letting ASR and TTS teach each other, our proposed speech-chain model significantly improved the ASR system in the CS test set, TstCSWord+ Phr (MixTTS), from 18.11% CER to 5.08% (13.03% absolute CER reduction) and maintained a good performance in the monolingual setting (only a slight CER reduction to 0.14% and 1.8% for the Japanese and English monolingual test sets, respectively). The same tendency was also shown in the TTS results, which also improved the TTS system in the CS test set TstCSWord+Phr (MixTTS) from 0.489 to 0.372 L2-norm squared and maintained a similar performance for the Japanese and English monolingual test sets.

7. CONCLUSION

We introduced a speech chain for semi-supervised learning of Japanese-English CS ASR and TTS. We first separately trained ASR and TTS systems with parallel speech-text of monolingual data (supervised learning) and performed a speech chain with only CS text or CS speech (unsupervised learning). Experimental results revealed that such closed-loop architecture allows ASR and TTS to teach each other and improved the performance even without any parallel CS data. Our proposed speech-chain model significantly improved the ASR and TTS systems in a CS setting and maintained a good performance in a monolingual setting. Note that although this study focuses only on Japanese-English conversion, the framework can be applied to any bilingual cases without significant modification. In the future, we will investigate natural, multi-speaker Japanese-English CS speech data.

8. ACKNOWLEDGEMENT

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237.

9. REFERENCES

[1] Japanese Ministry of Health, Labour and Welfare, "Overview of the population statistics in 2016 [in Japanese]," *http://www.mhlw.go.jp/*, 2016.

[2] Japanese Ministry of Education, Culture, Sports, Science, and Technology, "School basic survey in 2015 [in Japanese]," *http://www.mext.go.jp/*, 2015.

[3] Shana Poplack, *Code-switching (linguistic)*, chapter International Encyclopedia of the Social and Behavioral Sciences, pp. 918–925, Elsevier Science Ltd, 2nd edition edition, 2015.

[4] Masayo Nakamura, "Developing codeswitching patterns of a Japanese/English bilingual child," in *Proc. of the 4th International Symposium on Bilingualism*, Somerville, MA, USA, 2005, pp. 1679–1689.

[5] Sandra S. Fotos, "Japanese-English code switching in bilingual children," *JALT Journal*, vol. 12, no. 1, pp. 75–98, 1990.

[6] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. of IEEE ASRU*, Okinawa, Japan, 2017, pp. 301–308.

[7] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Machine speech chain with one-shot speaker adaptation," in *Proc. of INTERSPEECH*, Hyderabad, India, 2018, p. to appear.

[8] Jeff McSwan, "The architecture of the bilingual language faculty: Evidence from intrasentential code switching," *Bilingualism: Language and Cognition*, vol. 3, no. 1, pp. 37–54, 2000.

[9] Christopher M. White, Sanjeev Khudanpur, and James K. Baker, "An investigation of acoustic models for multilingual code switching," in *Proc. of INTERSPEECH*, Brisbane, Australia, 2008, pp. 2691–2694.

[10] David Imseng, Herve Bourlard, Mathew Magimai-Doss, and John Dines, "Language dependent universal phoneme posterior estimation for mixed language speech recognition," in *Prof. of ICASSP*, Prague, Czech Republic, 2011, pp. 5012–5015.

[11] Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li, "A first speech recognition system for Mandarin-English code-switch conversational speech," in *Proc. of ICASSP*, Kyoto, Japan, 2012, pp. 4889–4892.

[12] Basem H.A. Ahmed and Tien-Ping Tan, "Automatic speech recognition of code switching speech using 1-best rescoring," in *Proc. of International Conference on Asian Language Processing (IALP)*, Hanoi, Vietnam, 2012, pp. 137–140.

[13] Emre Yilmaz, Henkvan den Heuvel, and David van Leeuwen, "Investigating bilingual deep neural networks for automatic recognition of code-switching Frisian speech," *Procedia Computer Science*, vol. 81, pp. 159 – 166, 2016, SLTU - The 5th Workshop on Spoken Language Technologies for Under-resourced languages.

[14] S. Toshniwal, T. N. Sainath, R. J. Weiss, P. Moreno B. Li, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. of ICASSP*, Calgary, Canada, 2018.

[15] Min Chu, Hu Peng, Yong Zhao, Zhengyu Niu, and Eric Chang, "Microsoft Mulan-a bilingual TTS system," in *Proc. of ICASSP*, Hong Kong, China, 2003, pp. 264–267.

[16] Hui Liang, Yao Qian, and Frank K. Soong, "Microsoft Mulan-a bilingual TTS system," in *Proc. of ISCA Speech Synthesis Workshop (SSW6)*, Bonn, Germany, 2007, pp. 137–142.

[17] Sunayana Sitaram and Alan W. Black, "Speech synthesis of code-mixed text," in *Proc. of LREC*, Miyazaki, Japan, 2016, pp. 3422–3428.

[18] Sunayana Sitaram, SaiKrishna Rallabandi, Shruti Rijhwani, and Alan W. Black, "Experiments with cross-lingual systems for synthesis of code-mixed text," in *Proc. of ISCA Speech Synthesis Workshop (SSW9)*, Sunnyvale, CA, USA, 2016.

[19] SaiKrishna Rallabandi and Alan W. Black, "On building mixed lingual speech synthesis systems," Stockholm, Sweden, 2017, pp. 52–56.

[20] Hiroshi Seki, Shinji Watanabe, Takaaki Hori, Jonathan Le Roux, and John R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," Calgary, Canada, 2018.

[21] Peter B. Denes and Elliot N. Pinson, *The Speech Chain: The Physics And Biology Of Spoken Language*, Anchor books. Worth Publishers, 1993.

[22] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," *CoRR*, 2015.

[23] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell: A neural network

for large vocabulary conversational speech recognition," in *Proc. of ICASSP*, Shanghai, China, 2016, pp. 4960–4964.

[24] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," in *Proc. of INTERSPEECH*, Stockholm, Sweden, 2017, pp. 4006–4010.

[25] Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro 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.

[26] Genichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto, "Creating corpora for speech-to-speech translation," in *Proc. of EUROSPEECH*, Geneva, Switzerland, 2003, pp. 381–384.

[27] Sahoko Nakayama, Takatomo Kano, Quoc Truong Do, Sakriani Sakti, and Satoshi Nakamura, "Japanese-english code-switching speech data construction," in *Proc. of Oriental COCOSDA*, Miyazaki, Japan, 2018.

[28] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, 2015.

[29] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, 2014.

[30] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, 2015.