ZERO-SHOT CODE-SWITCHING ASR AND TTS WITH MULTILINGUAL MACHINE SPEECH CHAIN

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

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan

ABSTRACT

Constructing automatic speech recognition (ASR) and text-to-speech (TTS) for code-switching in a supervised fashion poses a challenge since a large amount of code-switching speech and the corresponding transcription are usually unavailable. The machine speech chain mechanism can be utilized to achieve semi-supervised learning. The framework enables ASR and TTS to assist each other when they receive unpaired data since it allows them to infer the missing pair and optimize the models with reconstruction loss. In this study, we handle multiple language pairs of code-switching by integrating language embeddings into the machine speech chain and investigate whether the model can perform with code-switching language pairs that are never explicitly seen during training. Experimental results reveal that the proposed approach improves the performance of the multilingual code-switching language pairs with which the model was trained and can also perform with unknown code-switching language pairs without directly learning on it.

Index Terms— speech recognition, code-switching, zero-shot, machine speech chain, language embedding

1. INTRODUCTION

Code-switching (CS), which is defined as when one speaker uses two or more languages interchangeably within a conversation, is a common phenomenon among bilingual conversations [1]. Code-switching has many varieties, but it can be classified into two primary categories: inter-sentential (switch is done at the sentence boundaries) and intra-sentential (shift is done in the middle of a sentence). The standard ASR and TTS are for monolingual data. Handling the code-switching situation well for those systems is complicated since they need to be able to deal with multilingual input with unpredictable switching positions.

Several studies addressed ASR for a certain language pair code-switching, such as Mandarin-English [2, 3, 4], English-Malay [5], and Frisian-Dutch [6]. Also in TTS research, approaches for Mandarin-English [7, 8], German-English [9, 10] as well as Hindi-English, Telugu-English, Marathi-English, and Tamil-English [11] CS have been investigated. Going beyond a single-language pair of CS, White et al. [12] investigated alternatives to model the acoustics for the code-switching of multiple language pairs, and Imseng et al. [13] proposed an approach to estimate the universal phoneme posterior probabilities for mixed-language speech recognition. Another alternative is to combine language identification (LID) and ASR by Seki et al. [14]. But again, these frameworks were only applied for ASR and still relied on supervised learning.

Recently, Guo et al. proposed semi-supervised acoustic and lexicon learning for an English-Mandarin CS ASR [15]. Although it enabled semi-supervised learning, it only focused on a single language pair for ASR tasks. Another work by Nakayama et al. [16] then attempted to perform semi-supervised learning for a Japanese-English CS ASR and TTS by utilizing a machine speech chain mechanism [17, 18, 19]. They trained ASR and TTS code-switching with labeled monolingual data (supervised learning) and performed a speech chain with only code-switching text or speech (unsupervised learning). Unfortunately, this previous work was also still designed only for a single language pair code-switching.

In summary, most existing approaches suffer from one or more of the following drawbacks: (a) just developed on ASR or TTS; (b) only focused on a single language pair; (c) trained in a supervised fashion that requires a large amount of paired code-switching data, in which the speech and corresponding transcription are usually unavailable. In this study, we address the code-switching tasks of ASR and TTS on multiple language pairs based on semi-supervised learning. Inspired by previous work, we also attempted to utilize the machine speech chain. But, in contrast, in this work, we handle multiple language pairs of code-switching by integrating language embeddings into the machine speech chain and investigate whether the model can perform with code-switching language pairs that are never explicitly seen during training. To aim multilingual CS tasks, we also propose to perform multi-task learning on ASR to learn the mapping from the speech input to both the text transcription and the language information using two softmax layers. The TTS then generates speech, given the joint input of text, language, and speaker vectors. We investigate the proposed approach on multilingual code-

switching language pairs with which the model was trained as well as the code-switching language pairs that were never explicitly seen during training.

2. MULTILINGUAL MACHINE SPEECH CHAIN

Inspired by the human speech chain [20], Tjandra et al. previously designed and constructed a machine speech chain based on deep learning [17, 18, 19]. The framework consists of a sequence-to-sequence ASR [21, 22] and a sequence-to-sequence TTS [23] as well as a loop connection between these two processes. The closed-loop architecture allows us to train our model on the concatenation of both the labeled and unlabeled data. Although ASR transcribes the unlabeled speech features, TTS reconstructs the original speech waveform based on the text from ASR. In the opposite direction, ASR also attempts to reconstruct the original text transcription with the synthesized speech.

Figure 1 illustrates the differences among the followings: (a) a basic machine speech chain for a monolingual ASR-TTS [17] or a single-pair code-switching ASR-TTS [16]; (b) a multi-speaker machine speech chain for a monolingual ASR-TTS [18]; and (c) our proposed multi-speaker multilingual machine speech chain for a monolingual, multilingual, and code-switching ASR-TTS. In this version, the machine speech chain incorporates language recognition within ASR. In other words, ASR performs multi-task learning for text transcription and language information using two softmax layers. We gave language information for each character by using language ID. The language ID is "JA" for Japanese, "EN" for English, "ZH" for Chinese, and "<unk>" for unknown languages. The TTS then generates speech, given the joint input of text, language, and speaker vector.

The training process is described as follows:

1. **Separately supervised training ASR and TTS with parallel speech-text monolingual data**
   We first separately train the ASR and TTS systems with parallel speech-text of the monolingual corpora from several languages shown in Fig. 2(a) using English (En), Japanese (Ja), and Chinese (Zh). Given parallel speech and text (character and language label sequences) of monolingual data ($\boldsymbol{x}^{Mono}, \boldsymbol{y}^{MonoChr}$, and $\boldsymbol{y}^{MonoLng}$), ASR generates sequence of character $\hat{\boldsymbol{y}}^{MonoChr}$ and language vectors $\hat{\boldsymbol{y}}^{MonoLng}$ with teacher-forcing directly using ground-truth ($\boldsymbol{y}^{MonoChr}$) and ($\boldsymbol{y}^{MonoLang}$) as decoder input and calculates the sum of loss $L_{ASR}^{MonoChr}(\hat{\boldsymbol{y}}^{MonoChr}, \boldsymbol{y}^{MonoChr})$ and $L_{ASR}^{MonoLng}(\hat{\boldsymbol{y}}^{MonoLng}, \boldsymbol{y}^{MonoLang})$. TTS also generates best predicted speech $\hat{\boldsymbol{x}}^{Mono}$ by teacher-forcing using reference ($\boldsymbol{x}^{Mono}$) from input character ($\boldsymbol{y}^{MonoChr}$) and language ($\boldsymbol{y}^{MonoLang}$) vectors, and we calculate the loss of $L_{TTS}^{Mono}(\hat{\boldsymbol{x}}^{Mono}, \boldsymbol{x}^{Mono})$. The parameters are then updated with gradient descent optimization. Here we also train the speaker recognition with Deep Speaker [24] (here denoted as "SPKREC") to produce speaker embedding vector $z = \text{SPKREC}(\mathbf{x})$ from speech input.

2. **Simultaneous unsupervised training ASR-TTS in a machine speech chain with unpaired CS data**

   (a) **Given only CS text (TTS→ASR)**
   Since this process has only CS text [y^{CSChr}, and y^{CSLng}] as input, we first generate a random speaker vector $\tilde{z} = \text{SPKREC}(\tilde{\mathbf{x}})$ from SPKREC. TTS learns to output speech waveform $\hat{\boldsymbol{x}}^{CS}$ from the input sequence of $[y^{CSChr}, y^{CSLng}]$, and ASR then predicts the sequence of character and language vectors [$\hat{\boldsymbol{y}}^{CSChr}$,$\hat{\boldsymbol{y}}^{CSLng}$], given the synthesized speech.
   Then the sum of losses $L_{ASR}^{CSChr}(\hat{\boldsymbol{y}}^{CSChr}, \boldsymbol{y}^{CSChr})$ and $L_{ASR}^{CSLng}(\hat{\boldsymbol{y}}^{CSLng}, \boldsymbol{y}^{CSLng})$ can be computed to update the ASR parameters (Fig. 2(b), left side).

   (b) **Given only CS speech (ASR→TTS)**
   This process has only speech features $\boldsymbol{x}^{CS}$ as input. Given unlabeled CS speech features x^{CS}, ASR generates sequence of character $\hat{\boldsymbol{y}}^{CSChr}$ and language $\hat{\boldsymbol{y}}^{CSLng}$ vectors, and SPKREC provides a speaker-embedding vector $z = \text{SPKREC}(\mathbf{x})$. TTS then predicts speech waveform $\hat{\boldsymbol{x}}^{CS}$, given output character and language texts from ASR. Then loss $L_{TTS}^{CS}(\hat{\boldsymbol{x}}^{CS}, \boldsymbol{x}^{CS})$ can be computed to update the TTS parameters (Fig. 2(b), right side).

Finally, all of the losses, including the monolingual and CS losses, are combined into a single loss while maintaining the balance between the supervised monolingual losses and unsupervised CS losses using hyperparameters (α, β):

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

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

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

where 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 must completely learn new matters with only CS speech or CS text.

3. EXPERIMENTAL SETUP

3.1. Monolingual and Code-Switching Corpora

We utilized the monolingual Ja, En, and Zh of the ATR Basic Travel Expression Corpus (BTEC) [25, 26], which

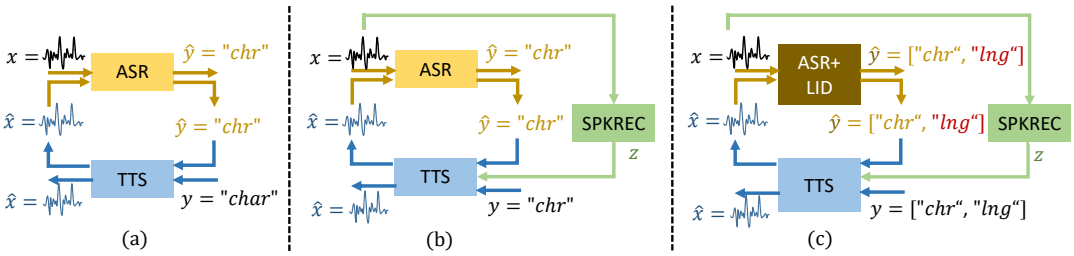

Fig. 1. Overview of machine speech chain models: (a) basic machine speech chain for monolingual ASR-TTS [17] or single-pair code-switching ASR-TTS [16]; (b) multi-speaker machine speech chain for monolingual ASR-TTS [18]; (c) proposed multi-speaker multilingual machine speech chain for monolingual, multilingual, and code-switching ASR-TTS.

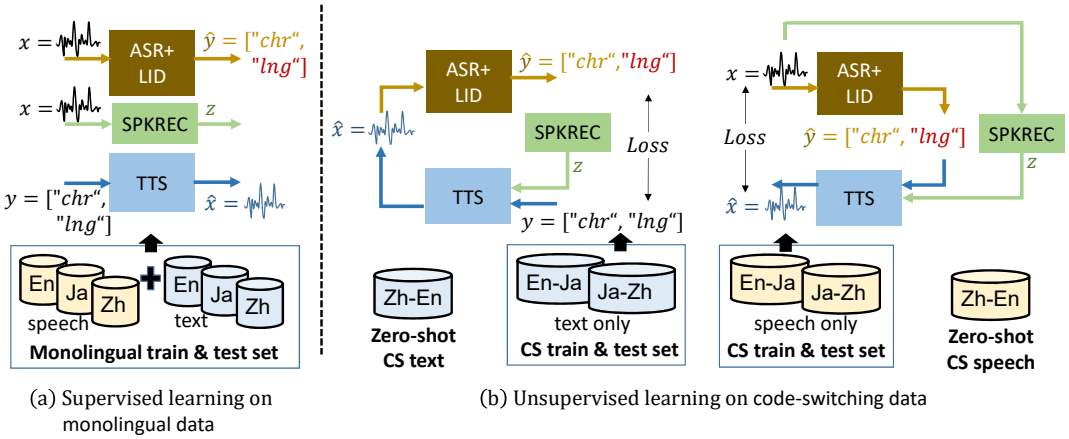

Fig. 2. Training process for CS machine speech chain: (a) separately supervised training ASR and TTS with parallel speech-text monolingual data; (b) simultaneously unsupervised training ASR-TTS in a machine speech chain with unpaired CS data, given only CS text (left-side) or CS speech (right-side).

covers basic conversations in the travel domain. The sentences are parallel translations among the three languages. For each language, we selected sentences that contain at least two phrases (separated with commas) and randomly selected 25k sentences for training (This set will be later called "Ja25k+En25k+Zh25k"), 500 for the development set, and another 500 for the test set. It corresponds to a total of about 85 hours. Here we artificially constructed code-switching sentences from the selected monolingual BTEC sentences by translating one phrase to the other languages (similar to a previous approach [27]), resulting in En-Ja, Ja-Zh, Zh-En, En-Fr, and Fr-Zh code-switching corpora (These sets will be later called "EnJaCS", "JaZhCS", "ZhEnCS", "EnFrCS", and "FrZhCS"). The switching position was chosen at the first comma.

For the English text, we converted all of the sentences into lowercase letters and removed all the punctuation marks [,:?.]. For the Japanese text, we extracted the katakana characters with a morphological analyzer called KyTea [28] and converted them into alphabet letters using pykakasi [29]. For the Chinese text, we applied pypinyin [30] to convert from Chinese characters to pinyin. 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 the end of sentences and the spaces between words.

Finally, all the text was synthesized to generate speech using Google TTS [31]. The Japanese part was synthesized by Japanese Google TTS, the English part was synthesized by English Google TTS, and the Chinese part was synthesized by Chinese (Mandarin) Google TTS.

Additionally, we also collected 1k utterances of natural text and speech CS. First, a Japanese-English bilingual speaker, who uses code-switching in his daily life, created natural code-switching text from BTEC Japanese-English parallel corpus. After that, we recorded the reading speech by another Japanese-English bilingual speaker. We divided the collected 1k utterances to 0.2k as paired data, 0.7k as unpaired data (This set will be later called "NatEnJaCS"), and 0.1k as test data (This set will be later called "Natural EnJaCS"). We also splitted the paired data to part of Japanese and part of English to use as monolingual data, which data we denote as "NatJa" and "NatEn".

3.2. Feature Extraction

All the raw speech waveforms were represented at a 16-kHz sampling rate. For the speech features, we used a log magnitude spectrogram extracted by the short-time Fourier transform (STFT) from the Librosa library [32]. First, we applied wave-normalization (scaling raw wave signals into the 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 frameshift, and a 2048 point FFT. 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.

3.3. ASR and TTS Systems

Our ASR system is an attention-based encoder-decoder model [21] that consists of three stacked BiLSTM encoders, a single layer LSTM, and multilayer perceptron (MLP)-based attention [33] components. The log-scaled Mel-spectrogram were fed into a fully connected layer and transformed by a LeakyReLU ($l = 1e-2$) [34] activation function. This model doesn't need any language model or any word dictionary. For the TTS system, our model is based on a sequence-to-sequence TTS (Tacotron) [23]. Although its hyperparameters are almost the same as the original Tacotron, we used LeakyReLU instead of ReLU. Also on the encoder, although the original Tacotron uses 16 sets of convolutional filters in the CBHG module, we used eight sets of different filter banks to reduce the GPU memory consumption. The encoder also has a language-embedding layer as well as a character-embedding layer. The decoder changed the GRU into two stacked LSTMs with 256 hidden units. Since the original Tacotron is a single speaker model, it cannot deal with multi-speakers. So we used a DNN-based speaker recognition model called Deep Speaker [24] to generate a speaker vector and incorporated a speaker-embedding layer into Tacotron.

As described before, we first separately trained the ASR and TTS systems with parallel speech-text of monolingual data (supervised learning). After that, we performed a machine speech chain with only CS text or CS speech (unsupervised learning). Both the ASR and TTS models were implemented with the PyTorch library [35]. We trained the Deep Speaker model with all of the speech utterances, including monolingual Japanese, English, Chinese and code-switching. For the α and β hyperparameters that scale the loss between the supervised (paralleled) and unsupervised (unparalleled) loss, we used the same $\alpha = 0.5$, $\beta = 1$ for most of our experiments.

4. EXPERIMENT RESULTS

4.1. ASR Evaluation

First, we investigated the impact of the additional language recognition on the baseline system. This is to confirm whether that additional information will not destruct the original quality. The baseline is an ASR **Ja25k+En25k+Zh25k** that was trained with a 25k monolingual Ja, a 25k monolingual En, a 25k monolingual Zh speech, and the corresponding text (character transcription and language information). Table 1 compares the performance (in CER) between the baselines that utilized the single-task ASR that only generates character transcription and a multi-task ASR that generates both character transcription and language information. The results show that an additional task on language recognition could even help the ASR performance. Therefore, we utilize the multi-task ASR model for further experiments with CS data.

Table 1. Comparison performance (in CER) of baselines between single-task and multi-task SR

Train:Ja25k+En25k+Zh25k	**Ja**	**En**	**Zh**
Single-task ASR [chr]	8.83%	9.08%	5.75%
Multi-task ASR [chr,lng]	8.85%	8.48%	5.11%

Next, we investigated our proposed approach on the multilingual code-switching language pairs with which the model was trained as well as the code-switching language pairs that were never explicitly seen during training. Please note that, in this research, the idea is not to show that the proposed method can outperform the baseline that only trained with a small set of data, but to learn whether we can improve the performance when only unpaired data is available. Specifically, our aim is to investigate whether there is still possible to improve the performance when paired data is not available and the CS language-pair has not been seen during training.

After separately training ASR and TTS using a parallel speech-text monolingual Ja25k, En25k, and Zh25k, we performed a speech chain using two language pairs of CS on three different setups: (1) **EnJaCS10k+JaZhCS10k**: an EnJaCS 10k and JaZhCS 10k training set with ZhEnCS as a zero-shot target. (2) **EnJaCS10k+ZhEnCS10k**: a EnJaCS 10k and ZhEnCS 10k training set with a JaZhCS as a zero-shot target. (3) **EnZhCS10k+ZhJaCS10k**: an EnZhCS 10k and ZhJaCS 10k training set with EnJaCS as a zero-shot target. As shown in Table 2, just by using the unpaired CS data and letting ASR and TTS teach each other, our proposed speech-chain model improved the ASR system in the multilingual CS test set, which includes not only the CS language pairs that were used during the speech chain training but also an unknown language pair compared with a baseline ASR system.

Additionally, we also investigated whether our proposed

Table 2. CER of proposed machine speech chain with language embedding on zero-shot CS and known language (The bold figures indicate a zero-shot training).

	Monolingual			Code-switching		
	Ja	**En**	**Zh**	**EnJaCS**	**JaZhCS**	**ZhEnCS**
Baseline: Supervised training on monolingual data only						
Ja25k+En25k+Zh25k (paired)	8.85%	8.48%	5.11%	**14.06%**	**16.91%**	**16.04%**
Proposed Machine Speech chain: Semi-supervised training on two CS data						
+EnJaCS10k+JaZhCS10k (unpaired)	9.18%	12.71%	5.93%	11.56%	8.31%	**10.52%**
+EnJaCS10k+ZhEnCS10k (unpaired)	8.93%	12.34%	5.67%	11.18%	**9.21%**	9.71%
+ZhEnCS10k+JaZhCS10k (unpaired)	8.91%	14.45%	6.08%	**11.85%**	10.40%	11.29%
Topline: Supervised training on CS data						
+EnJaCS10k+JaZhCS10k (paired)	10.18%	12.32%	7.93%	8.94%	6.70%	**8.09%**
+EnJaCS10k+ZhEnCS10k (paired)	11.04%	10.91%	7.48%	10.81%	**7.26%**	8.07%
+ZhEnCS10k+JaZhCS10k (paired)	10.98%	11.57%	7.22%	**10.34%**	7.72%	7.98%
+EnJaCS10k+JaZhCS10k+ZhEnCS10k (paired)	10.48%	10.43%	6.88%	8.68%	6.98%	8.05%

Table 3. CER of proposed machine speech chain with language embedding on zero-shot CS and known language and natural CS (The bold figures indicate a zero-shot training).

	Monolingual			Code-switching			
	Ja	**En**	**Zh**	**EnJaCS**	**JaZhCS**	**ZhEnCS**	**Natural EnJaCS**
Baseline: Supervised training on monolingual data only							
Ja25k+En25k+Zh25k plus NatJa0.2k+NatEn0.2k (paired)	15.22%	17.14%	6.36%	**20.23%**	**21.23%**	**19.49%**	**66.11%**
Proposed Machine Speech chain: Semi-supervised training on two CS data and one natural CS data							
+EnJaCS10k+JaZhCS10k plus NatEnJaCS0.7K (unpaired)	15.68%	18.26%	6.69%	12.29%	15.57%	**15.64%**	29.99%
Topline: Supervised training on two CS data and one natural CS data							
+EnJaCS10k+JaZhCS10k plus NatEnJaCS0.7K (paired)	16.66%	18.41%	8.09%	8.70%	6.98%	**8.94%**	22.51%

speech-chain model could also improve the ASR system in the multilingual CS test set with natural speech CS. The natural CS sentences tend to be more complex than the synthetic one because CS speakers may switch twice or more within a single utterance. Furthermore, we only have limited available 1k natural CS data. As can be seen from Table 3, the performances were worse than using only synthetic data. Nevertheless, the results can still reveal that our proposed speech-chain model could also improve the ASR system in the multilingual CS test set with natural CS.

Furthermore, we also investigated the performance of unseen CS language pairs of French and Chinese (FrZhCS) given in the situation where Fr language is unknown and monolingual Fr paired data is not available at all. Thus, the language recognition would not have the chance to identify the language, and the system did not have the chance to learn French in the supervised learning or never been taught the association of the French speech and the ground-truth of the corresponding transcription. Table 4 shows the ASR performance. As can be seen, even the Fr language is unknown and any paired data of monolingual Fr is unavailable, through a multilingual machine speech chain mechanism that learns unpaired EnFrCS data, we can still improve the performance on FrZhCS test data. Surprisingly, the topline CER with paired training data got worse than the unpaired counterpart. As we challenged one-shot learning, although the topline model was trained with paired training data, it only had the paired data of EnJaCS+JaZhCS+EnFrCS data. It was never be trained with the target FrZhCS. The results might indicate that as non-target CS training data become more massive, the mismatch increase and the performance becomes worse. In any case, these results also reveal that the proposed framework can still improve the performance even for zero-shot CS that includes unknown language.

4.2. TTS Evaluation

We performed an AB preference subjective evaluation between the speech sentence pairs generated by the proposed CS that has language-embedding $[y^{CSChr}, y^{CSLng}]$ in the input

Table 4. CER of the proposed machine speech chain with language embedding on zero-shot CS Fr language is unknown and monolingual Fr paired data is not available at all (no supervised learning for Fr).

Model	Train data	**FrZhCS**
Baseline	Ja25k+En25k+Zh25k (paired)	36.30%
Proposed	+EnJaCS10k+JaZhCS10k+EnFrCS5k (unpaired)	28.95%
	+EnJaCS10k+JaZhCS10k+EnFrCS10k (unpaired)	26.77%
Topline	+EnJaCS10k+JaZhCS10k+EnFrCS5k (paired)	22.42%
	+EnJaCS10k+JaZhCS10k+EnFrCS10k (paired)	31.27%

and a CS that does not (only $[y^{CSChr}, y^{CSLng}]$). 10 bilingual speakers for each language pair participated in subjective tests. For each paired test stimuli in the overall evaluation, the subjects were shown the transcription and listened to two speech utterances, and were asked to choose from the following answers in terms of being more native or not: a) the former is better; b) the latter is better; c) cannot tell any difference or which is better.

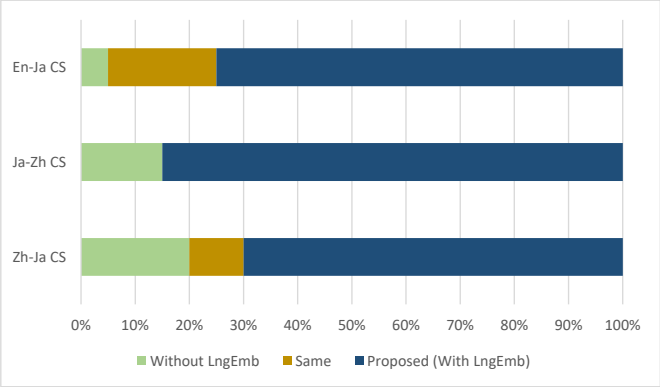

Fig. 3. AB preference subjective evaluation between CS TTS with language embedding and without

We selected 20 generated speech utterances from the test set and randomized the order of the stimuli presentation of the sentences. The results (Fig. 3) indicate that although incorporating language recognition complicates multi-task ASR, it helps TTS essentially maintain the synthesis speech quality, especially on the switch position between two languages.

5. RELATED WORKS

Zero-shot learning originally refers to multiclass classification problems in the field of computer vision that recognizes objects whose instances may not have appeared in the training data [36]. Some new zero-shot learning methods were mainly proposed in previously summarized image processing researches [37].

In neural machine translation, zero-shot translation has been studied, where the translation between the unknown language pairs that have never seen in the training set can be conducted [38]. One experiment demonstrated that two translation models that were trained with Portuguese (Pt)-English (En) and English (En)-Spanish (Es) can generate reasonable translation quality on Portuguese (Pt)-Spanish (Es) without ever being seen during training.

Unfortunately, since little work has addressed code-switching ASR and TTS, our study contributes to zero-shot code-switching ASR and TTS researches.

6. CONCLUSION

We introduced a zero-shot code-switching ASR and TTS with a multilingual machine speech chain. The previous research utilized a machine speech chain and achieved semi-supervised learning of ASR and TTS by optimizing the parameters from back-propagating errors through the whole system. However, that system was designed only for the code-switching of a single language pair. In this study, we expanded the model to handle multilingual code-switching by integrating a neural language that was embedded in the machine speech chain. We also investigated whether it can perform on code-switching language pairs that were never explicitly seen during training. Experimental results reveal that a single machine speech chain architecture that integrated the language embedding improved the performance of the multilingual code-switching language pairs with which the model was trained and performed well on the unknown language set of code-switching without directly learning that code-switching language set.

7. ACKNOWLEDGEMENT

Part of this work is supported by JSPS KAKAENHI Grant Numbers JP17H06101 and JP17K00237 as well as NII CRIS Contract Research 2019 and Google AI Focused Research Awards Program.

8. REFERENCES

[1] Lesley Milroy and Pieter Muysken, *One speaker, two languages: Cross-disciplinary perspectives on code-switching*, Cambridge University Press, 1995.

[2] 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.

[3] Ne Luo, Dongwei Jiang, Shuaijiang Zhao, Caixia Gong, Wei Zou, and Xiangang Li, "Towards end-to-end code-switching speech recognition," *arXiv preprint arXiv:1810.13091*, 2018.

[4] Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie, "Investigating end-to-end speech recognition for mandarin-english code-switching," in *Proc. of ICASSP*. IEEE, 2019, pp. 6056–6060.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] Pengcheng Guo, Haihua Xu, Lei Xie, and Eng Siong Chng, "Study of semi-supervised approaches to improving english-mandarin code-switching speech recognition," *arXiv preprint arXiv:1806.06200*, 2018.

[16] Sahoko Nakayama, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Speech chain for semi-supervised learning of japanese-english code-switching asr and tts," in *Proc. of IEEE Spoken Language Technology (SLT)*, Athens, Greece, 2018.

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

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

[19] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," in *Proc. of ICASSP*, Brighton, UK, 2019, p. to appear.

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

[21] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. of ICASSP*. IEEE, 2016, pp. 4945–4949.

[22] 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.

[23] 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.

[24] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[25] Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita, "Multilingual spoken language corpus development for communication research," *Proc. of 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] Graham Neubig, Yosuke Nakata, and Shinsuke Mori, "Pointwise prediction for robust, adaptable japanese morphological analysis," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 2011, pp. 529–533.

[29] Hiroshi Miura, "pykakasi – kakasi library in python," *https://pypi.org/project/pykakasi/*.

[30] Huang Huang, "pypinyin – pinyin library in python," *https://pypi.org/project/pypinyin/*.

[31] Pierre Nicolas Durette, "gTTS – Google Text-to-Speech," *https://pypi.org/project/gTTS/*.

[32] Brian McFee, Matt McVicar, Oriol Nieto, Stefan Balke, Carl Thome, Dawen Liang, Eric Battenberg, Josh Moore, Rachel Bittner, Ryuichi Yamamoto, et al., "librosa 0.5.0," *https://librosa.github.io/librosa/0.5.0/index.html*, 2017.

[33] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[34] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[36] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio, "Zero-data learning of new tasks.," in *Proc. of AAAI*, 2008, vol. 1, p. 3.

[37] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata, "Zero-shot learning: the good, the bad and the ugly," *arXiv preprint arXiv:1703.04394*, 2017.

[38] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al., "Google ' s multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.