Recognition and Translation of Code-switching Speech Utterances

Sahoko Nakayama
Nara Institute of Science & Technology
Japan
nakayama.sahoko.nq1@is.naist.jp

Takatomo Kano
Nara Institute of Science & Technology
Japan
kano.takatomo.km0@is.naist.jp

Andros Tjandra
Nara Institute of Science & Technology
Japan
andros.tjandra.ai6@is.naist.jp

Sakriani Sakti
Nara Institute of Science & Technology
RIKEN, Advanced Intelligence Project AIP
Japan
ssakti@is.naist.jp

Satoshi Nakamura
Nara Institute of Science & Technology
RIKEN, Advanced Intelligence Project AIP
Japan
s-nakamura@is.naist.jp

***Abstract*—Code-switching (CS), a hallmark of worldwide bilingual communities, refers to a strategy adopted by bilinguals (or multilinguals) who mix two or more languages in a discourse often with little change of interlocutor or topic. The units and the locations of the switches may vary widely from single-word switches to whole phrases (beyond the length of the loanword units). Such phenomena pose challenges for spoken language technologies, i.e., automatic speech recognition (ASR), since the systems need to be able to handle the input in a multilingual setting. Several works constructed a CS ASR on many different language pairs. But the common aim of developing a CS ASR is merely for transcribing CS-speech utterances into CS-text sentences within a single individual. In contrast, in this study, we address the situational context that happens during dialogs between CS and non-CS (monolingual) speakers and support monolingual speakers who want to understand CS speakers. We construct a system that recognizes and translates from code-switching speech to monolingual text. We investigated several approaches, including a cascade of ASR and a neural machine translation (NMT), a cascade of ASR and a deep bidirectional language model (BERT), an ASR that directly outputs monolingual transcriptions from CS speech, and multi-task learning. Finally, we evaluate and discuss these four ways on a Japanese-English CS to English monolingual task.**

***Index Terms*—code-switching, speech recognition, speech and text translation, BERT, multi-task learning**

I. Introduction

The number of international travelers and residents in Japan is monotonously increasing for such purposes as tourism, or education. According to a survey of the Ministry of Health, Labour and Welfare (MHLW), there were 21,457 international marriages in Japan in 2017, an increase of about 3.5 times in 40 years [1]. These changes affect how people communicate. As a result, the phenomenon of Japanese-English code-switching is becoming more frequent. Fotos et al. investigated four hours of conversation of four bilingual children in Japan who had either one or two American parents and observed 153 code-switchings [2]. Their reports revealed that some people actually use Japanese-English CS in their everyday lives. This phenomenon is challenging for spoken language technologies, i.e., automatic speech recognition (ASR), since such systems need to be able to handle the input in a multilingual setting.

Several works have constructed a CS ASR on many different language pairs. White et al. [3] investigated alternatives to model the acoustics for multilingual code-switching, and Imseng et al. [4] proposed an approach to estimate universal phoneme posterior probabilities for mixed-language speech recognition. Vu et al. [5] focused on addressing speech recognition of Chinese and English code-switching and proposed approaches for phoneme merging in combination with discriminative training as well as the integration of language identification systems into decoding processes. Recently, Yilmaz et al. [6] investigated the impact of bilingual deep neural networks in the contexts of Frisian and Dutch CS. But the common aim of developing a CS ASR is merely for transcribing CS-speech utterances into CS-text sentences within the speech of a single individual.

In contrast, in our study, we address the situational context during dialogs between CS and non-CS (monolingual) speakers to support monolingual speakers who are trying to understand CS speakers. We construct a system that can recognize code-switching speech and translate to monolingual texts. CS translation is difficult since systems must detect unpredictable switching positions and translate the broken context as monolingual language. To address the problems, we investigate several approaches, including a cascade of ASR and neural machine translation (NMT), a cascade of ASR and a deep bidirectional language model (BERT), an ASR that directly outputs monolingual transcriptions from CS speech, and multi-task learning. We evaluate and discuss these four ways on a Japanese-English CS to English monolingual task.

978-1-7281-2449-0/19/$31.00 ©2019 IEEE

TABLE I
EXAMPLE OF MONOLINGUAL TEXT RECOVERY USING BERT-MASKED LM

Source CS	あの ね , charles may be a bad husband , but he ' s とても 心 の 温か い 人 な の よ .
Masked text	[MASK] [MASK] [MASK] [MASK] [MASK] [MASK] , charles may be a bad husband , but he ' s [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] .
Label	you know [PAD] [PAD] [PAD] [PAD], charles may be a bad husband, but he ' s a very warm - hearted person .
Target English	you know , charles may be a bad husband , but he ' s a very warm - hearted person .

II. PROPOSED APPROACHES ON CODE-SWITCHING-TO-MONOLINGUAL TASK

As described above, we investigated several approaches to address communication problems across languages. First, we offer solutions for the transformation of CS text to monolingual text. After that, we transform CS speech to monolingual text. The details are described below.

A. Code-switching Text to Monolingual Text

1) Mono-Recovery BERT:

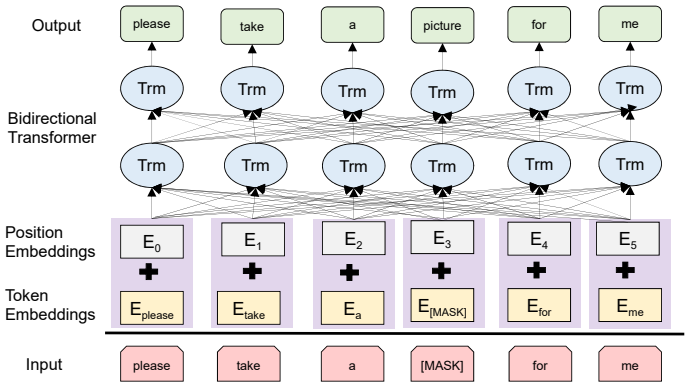

Fig. 1. Model architecture of BERT for code-switching-to-monolingual recovery.

Traditional language models (LMs) are based on a single-directional (left-to-right) approach that predicts the next word given a sequence. Unfortunately, such an approach limits the learning of context. BERT [7], which stands for Bidirectional Encoder Representations from Transformers, is a language understanding model that is bidirectionally (left-to-right and right-to-left) trained on a massive text corpus. In contrast with a traditional LM, BERT [7] has a deeper sense of language context.

BERT [7] exploits the Transformer [8], an attention mechanism that bidirectionally learns the contextual relations between words (or sub-words) in a text. It has two training phases: (1) pre-training with a generic dataset for language representation and (2) fine-tuning on a specific task, such as sentiment analysis [9], question answering [10], name entity recognition [11], which is trained with a domain-specific dataset. Ghazvininejad et al. also utilized conditional masked language models like BERT for translation tasks by introducing a new mask-prediction algorithm [12] that repeatedly selects the new positions of the mask tokens and predicts them at each iteration.

Since we only need a language understanding model, we just utilized a pre-trained BERT that leverages a masked language model (Masked LM). By randomly masking some tokens, we used other tokens to predict the masked tokens to learn the representations. Unlike other approaches, BERT predicted masked tokens instead of the entire input. Thus, in our case, given a CS text that mixed words from the 1st and 2nd languages, we masked unwanted words from the 2nd language and used BERT to recover complete sentences in the monolingual text of the 1st language. Since we do not know exactly how many words should be replaced, we put several [MASK] tokens in the positions of unwanted words. Then the model is filled with tokens [PAD] if the original target token size is smaller than the number of [MASK] tokens. Table I shows an example of monolingual text recovery using a BERT-masked LM.

The architecture uses a multi-layer bidirectional Transformer encoder [8]. We followed the hyperparameters and the weight initialization scheme to the BERT$_{Base}$ model, which is a publicly available BERT English model with 12 layers, 768 hidden sizes, 12 self-attention heads, and 110-M parameters, and 30522 words [7]. The model architecture is depicted in Fig. 1. It is simpler than the original BERT model [7]. We did not have to use the special classification embeddings (`[CLS]`). We also did not have to use the segmentation embeddings for next sentence predictions. We also did not use a special token (`[SEP]`) for separating sentences, and instead of that, we separated sentences for each input.

2) CS2Mono NMT:

Here we perform neural machine translation (NMT) from code-switching to a monolingual text. Despite extensive research on MT, few works address the problem of CS translation. Sinha et al. [13] translated Hindi-English CS by isolating each language. Since they used a traditional approach instead of a neural MT, the context between language switchings is unlikely to be considered. Recently, Google's multilingual neural machine translation system [14] produced CS examples from monolingual sentences by weighting language selection in a linear combination of embedding vectors. The system simply outputs words randomly from different languages.

In contrast, we trained NMT to learn CS to monolingual text translation on synthetic and natural data. Our NMT system is a standard attention-based encoder-decoder model [15], [16]. In the encoder, we fed the input text into a fully connected layer

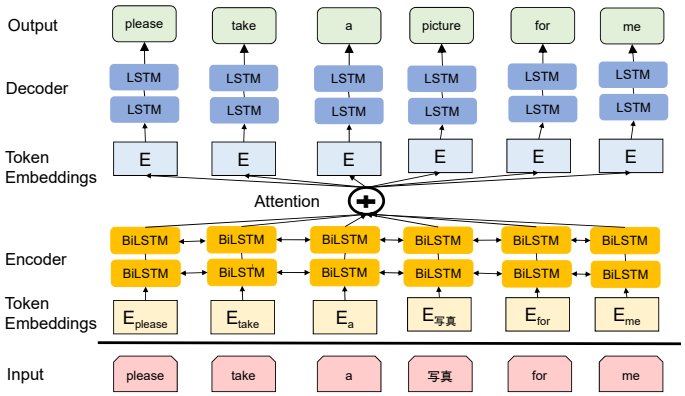

Fig. 2. Model architecture of NMT.

and transformed it by a LeakyReLU ($l = 1e{-}2$) [17] activation function. The output goes through two stacked BiLSTM layers with 256 hidden units for each direction (512 hidden units in both directions). In the decoder, the characters were projected by two LSTM layers with 512 hidden units. The decoder used the attention module with scores calculated by a multilayer perceptron [18]. The model architecture is depicted in Fig. 2.

B. Code-switching Speech to Monolingual Text

1) Cascade CS2CS ASR + Mono-recovery BERT:

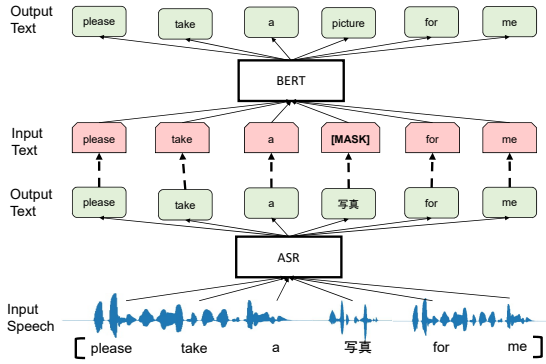

Fig. 3. Model architecture of Cascade of ASR+BERT

Given code-switching speech, we first performed a neural ASR and produced CS text. Then we utilized BERT to recover the monolingual text. The cascade model architecture is depicted in Fig. 3. The ASR system is an attention-based encoder-decoder model [15], [16]. In the encoder, the input features a log-scaled Mel-spectrogram fed into a fully connected layer and transformed by a LeakyReLU ($l = 1e{-}2$) [17] activation function. The output goes through three stacked BiLSTM layers that have 256 hidden units for each direction (512 hidden units in both directions). In the decoder, the characters were fed into a 128-dims embedding layer and projected by one LSTM layer with 512 hidden units. The decoder uses the attention module and calculates the score with a multilayer perceptron [18]. The ASR model architecture is depicted in Fig. 5, and BERT is the same model described in Section II-A1.

2) Cascade CS2CS ASR + CS2Mono NMT:

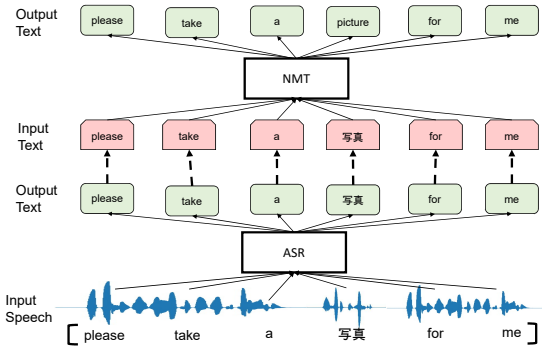

Fig. 4. Model architecture of Cascade of ASR+NMT

Given CS speech, we first performed a neural ASR and produced CS text. After that, we utilized NMT to translate from CS to monolingual text. Similar to the cascade ASR + BERT, we used our attention-based encoder-decoder model. The cascade model architecture is depicted in Fig. 4. The ASR system has the same architecture as above, and NMT is the same model described in Section II-A2.

3) Direct CS2Mono ASR with single-task learning:

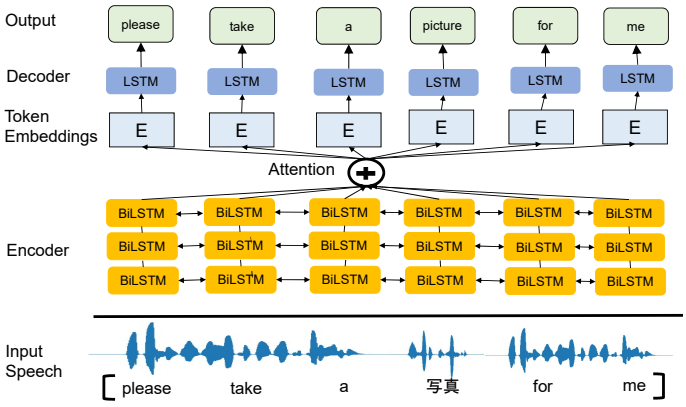

Fig. 5. Model architecture of ASR.

Here we trained an attention-based encoder-decoder ASR to produce monolingual text given the CS speech. Although this model uses the same architecture as the model described in Fig. 5, it directly generates English transcriptions from CS speech.

4) Direct CS2Mono ASR with multi-task learning:

Multi-task learning for speech translation has variations. Typical multi-task learning [19] shares an encoder. Triangle multi-task learning [20] provides information from a decoder as well as a shared encoder. We adopted the typical multi-task learning that has two decoders with shared an encoder. The first decoder outputs CS text, and the second outputs monolingual text. The shared encoder and ASR decoder have

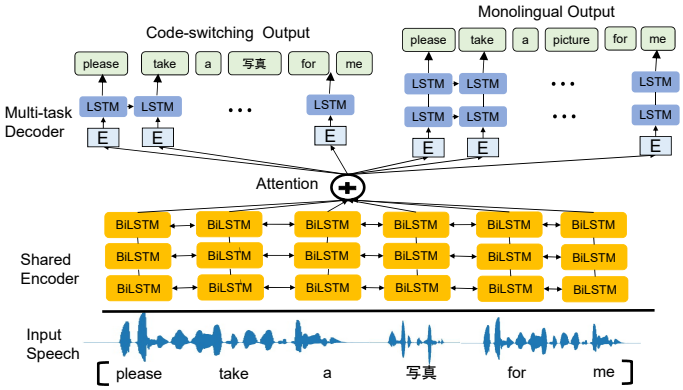

Fig. 6. Model architecture of multi-task learning.

the same hyperparameters as the ASR model. The translation decoder has the same hyperparameters as the decoder of the NMT model. This model architecture is depicted in Fig. 6.

III. Code-switching Corpora

We constructed artificial code-switching and natural code-switching. For those corpora, we utilized the monolingual Japanese and English ATR Basic Travel Expression Corpus (BTEC) [21], [22].

According to Bautista [23], code-switching is caused by either proficiency-driven or deficiency-driven. The proficiency-driven code-switching occurs when a speaker is competent with both languages and easily able to switch from one language to another. The deficiency-driven code-switching occurs when a speaker is lack of competency of one language and therefore has to go back to another language. As the data of deficiency-driven code-switching has not been obtained yet, we only handle the proficiency-driven code-switching in this work.

Loanwords and quotations are not theoretically code-switching, but we also handle loanwords and quotations within a CS framework because we aim to recognize every word in Japanese-English conversations.

All of the constructed sentences were tokenized. We applied a morphological analyzer, Mecab [24], on the Japanese sentences and WordPiece tokenization [25] on the English sentences.

In the following paragraphs, We explain the details about how artificial code-switching and natural code-switching were constructed.

A. Artificial code-switching

First, we chose the switching positions based on the result of TreeTagger, a part-of-speech tagging tool [26].

- Word-level CS:
  - One Word-Insertion (Insertion 1): Since the number of target tokens in the switching position is 1, the number of [MASK] tokens is also 1.
  - Two Word Insertion: Since the number of target tokens in the switching position is 2, the number of [MASK] tokens is also 2.
- Phrase-level CS:
  This corpus has artificial phrase-level code-switching that is longer than the word-level CS. The number of target tokens in a switching position is not determined. We used four [MASK] tokens to predict the target tokens in each switching position. If the length of the target tokens did not reach 4, we used [PAD] tokens to fill in the remaining labels

We chose a noun as insertion 1 in word-level code-switching, and insertion 2 contains the tokens that come before insertion 1, which are mostly determiners. The phrase-level CS position was chosen after the prepositions. Given the switching positions, we translated them by machine translation and used the neural machine translation model trained with the English BTEC - Japanese BTEC. We only used the Google translation API when the neural machine translation did not generate the translation result well.

That corpus was divided 50-K sentences for a training set, 500 for a development set, and 500 for a test set.

All the text was synthesized using Google TTS.

B. Natural code-switching

A Japanese-English bilingual speaker made the CS text. Although he lives in an English-speaking country, his parents are Japanese. He also studied in Japan for one year. Therefore, he often uses code-switching in his daily life. We gave him 1000 pairs of Japanese-English sentences from the BTEC from which he made phrase-level CS sentences from pairs.

This corpus includes 0.9-K natural CS sentences for a training set and 0.1K for a test set among all the artificial CSs. The number of target tokens in a switching position was not determined. We used six [MASK] tokens to predict the target tokens in each switching position. If the length of the target tokens did not reach 6, we used [PAD] tokens to fill in the remaining labels.

We then asked a bilingual speaker to read and record the constructed natural CS text. He recorded at his own residence, but he did so in a quiet room.

We sampled all the speech waveforms at a sampling rate of 16 kHz. For the speech features, we used a log magnitude spectrogram extracted by short-time Fourier transform (STFT) from the Librosa library [27]. First, we applied wave-normalization (scaling 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 frameshift, and a 2048-point FFT. After we got the spectrogram, we took the squared magnitude and extracted the Mel-spectrogram with a Mel-filterbank with 40 filters.

IV. Experiments

Table II shows the WordPiece error rate between the CS text as the source text and the English text as the target text. Our aim is to reduce the errors and produce language that more closely resembles the monolingual English text.

TABLE II
WordPiece error rate (%) between CS text as source text and English text as target text

Word-level CS	
Artificial (Insertion 1)	6.73
Artificial (Insertion 2)	13.93
Phrase-level CS	
Artificial	27.20
Mix Natural	34.96

A. Code-switching Text to Monolingual Text

First, we performed a CS text to a monolingual text. Table III shows the WordPiece error rate of the word-level CS on the BERT model and Table IV shows the WordPiece error rate of the word-level CS on the NMT model. Regarding the training data, BERT required less data than NMT. Because, BERT needed only monolingual data with MASK, while NMT needed parallel of CS and monolingual data. In the word-level CS task, surprisingly BERT could perform better than NMT in mismatched cases, while NMT performed better in the matched case.

Next, Table V and Table VI show the WordPiece error rate of phrase-level CS on the BERT model and the NMT model, respectively. In the phrase-level CS task, the NMT model translated all artificial CS tests well, but the BERT model translated the natural CS test better. Natural CS generally has more complex Japanese phrases as its source text, which makes the translation task by NMT became complicated. However, as the BERT model did not have any constrained on the CS source text, its prediction could outperform the NMT model.

TABLE III
WordPiece error rate (%) of word-level CS on BERT model

	Insertion 1 Test	**Insertion 2 Test**
Insertion 1	3.31	11.20
Insertion 2	3.54	7.98

TABLE IV
WordPiece error rate (%) of word-level CS on NMT model

	Insertion 1 Test	**Insertion 2 Test**
Insertion 1	2.02	13.76
Insertion 2	4.08	4.96

TABLE V
WordPiece error rate (%) of phrase-level CS on BERT model

	Artificial CS Test	**Mix Natural CS Test**
All Artificial CS	9.75	14.96
Mix Natural CS	10.37	13.21

TABLE VI
WordPiece error rate (%) of phrase-level CS on NMT model

	Artificial CS Test	**Mix Natural CS Test**
All Artificial CS	7.56	16.34
Mix Natural CS	6.34	19.16

B. Code-switching Speech to Monolingual Text

Table VII and Table VIII show the WordPiece error rate of the phrase-level CS on the cascade of the ASR+BERT model and the ASR+NMT model, respectively. Table IX and Table X show the WordPiece error rate of the phrase-level CS on the direct ASR model using single-task and multi-task learning, respectively.

Among these models, the cascade models learned a much simpler task than direct recognition. Furthermore, ASR+BERT used less data than ASR+NMT. In this condition, it seems the task became too hard for BERT, as the ASR error increased the number of [MASK] tokens. Nevertheless, BERT still got a close value to that of other models. In summary, it seems that the more powerful model, the better the performance.

TABLE VII
WordPiece error rate (%) of phrase-level CS on cascade of ASR+BERT

	Artificial CS Test	**Mix Natural CS Test**
All artificial CS	17.80	47.98
Mix Natural CS	18.62	29.22

TABLE VIII
WordPiece error rate (%) of phrase-level CS on cascade of ASR+NMT

	Artificial CS Test	**Mix Natural CS Test**
All artificial CS	11.07	43.52
Mix Natural CS	11.32	25.29

TABLE IX
WordPiece error rate (%) of phrase-level CS on direct ASR

	Artificial CS Test	**Mix Natural CS Test**
All artificial CS	11.54	38.35
Mix Natural CS	10.89	26.10

TABLE X
WORDPIECE ERROR RATE (%) OF PHRASE-LEVEL CS ON MULTI-TASK LEARNING

	Artificial CS Test	**Mix Natural CS Test**
All artificial CS	10.55	36.49
Mix Natural CS	10.70	25.75

V. CONCLUSION

We compared four ways to achieve code-switching translation: a cascade of ASR and neural machine translation (NMT), a cascade of ASR and BERT, an ASR that directly outputs monolingual transcription from CS speech, and multi-task learning. The results reveal that the more powerful model, the better the performance.

In the future, we will further investigate how natural the translation results are by conducting an evaluation by bilingual people. As this paper is targeted only to proficiency-driven code-switching, we will handle deficiency-driven code-switching in the future. Furthermore, we will investigate the possible combinations of BERT with other models.

VI. ACKNOWLEDGEMENT

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

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] Sandra S. Fotos, "Japanese-English code switching in bilingual children," in *Proc. of JALT Journal*, 1990, vol. 12, pp. 75–98.

[3] 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.

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

[5] 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.

[6] Emre Yilmaz, Henkvan den Heuvel, and David van Leeuwen, "Investigating bilingual deep neural networks for automatic recognition of code-switching Frisian speech," in *Proc. of SLTU*, Yogyakarta, Indonesia, 2016, vol. 81, pp. 159 – 166.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL-HLT*, 2019, pp. 4171–4186.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. of NIPS*, 2017, pp. 5998–6008.

[9] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of EMNLP*, 2013, pp. 1631–1642.

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proc. of EMNLP*, 2016, pp. 2383–2392.

[11] Erik F. Tjong Kim Sang and Fien De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. of HLT-NAACL*, 2003, pp. 142–147.

[12] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer, "Constant-time machine translation with conditional masked language models," *arXiv preprint arXiv:1904.09324*, 2019.

[13] R Mahesh K Sinha and Anil Thakur, "An investigation of acoustic models for multilingual code switching," in *Proc. of the 10th conference on machine translation*, Phuket, Thailand, 2005, pp. 149–156.

[14] 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," in *Proc. of Transactions of the Association for Computational Linguistics*, 2017, vol. 5, pp. 339–351.

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

[16] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. of ICASSP*, 2016.

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

[18] Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. of EMNLP*, 2015, pp. 1412–1421.

[19] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Proc. of Interspeech*, Stockholm, Sweden, 2017, pp. 2625–2629.

[20] Antonios Anastasopoulos and David Chiang, "Tied multitask learning for neural speech translation," in *Proc. of NAACL-HLT*, 2018, pp. 82–91.

[21] Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita, "Multilingual spoken language corpus development for communication research," in *Proc. of The Association for Computational Linguistics and Chinese Language Processing*, 2007, vol. 12, pp. 303–324.

[22] 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.

[23] Maria Lourdes S Bautista, "Tagalog-english code switching as a mode of discourse," in *Proc. of Asia Pacific Education Review*, 2004, vol. 5, pp. 226–233.

[24] Taku Kudo, "Mecab: Yet another part-of-speech and morphological analyzer," *http://taku910.github.io/mecab*, 2006.

[25] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[26] Helmut Schmid, "Treetagger-a language independent part-of-speech tagger," *http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/*, 1994.

[27] 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.