論文 / 著書情報
Article / Book Information

論題	Incorporating Acoustic and Textual Information for Language Modeling in Code-switching Speech Recognition
Title	Incorporating Acoustic and Textual Information for Language Modeling in Code-switching Speech Recognition
著者	Hartanto Roland, 宇都 有昭, 篠田 浩一
Authors	Roland HARTANTO, Kuniaki UTO, Koichi SHINODA
出典	, vol. 121, no. 385, pp. 56-63
Citation	IEICE Technical Report, vol. 121, no. 385, pp. 56-63
発行日 / Pub. date	2022, 3
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright(c) 2022 IEICE

一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
EA2021-73,SIP2021-100,SP2021-58(2022-03)

Incorporating Acoustic and Textual Information for Language Modeling in Code-switching Speech Recognition

Roland HARTANTO　Kuniaki UTO　and　Koichi SHINODA

Tokyo Institute of Technology　2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan
E-mail:　{roland, uto}@ks.c.titech.ac.jp,　shinoda@c.titech.ac.jp

Abstract　People who speak two or more languages tend to alternate the language when they are speaking. This particular phenomenon is called code-switching, and it frequently occurs in multicultural society. Automatic speech recognition (ASR) for code-switching speech is a challenging task acoustically and linguistically because of the lack of code-switching data. This work aims to improve code-switching ASR system by improving the language model. We explore the code-switching data augmentation for language modeling by utilizing the ASR decoding lattice to tackle the pronunciation variation and data scarcity problems. We incorporate both acoustic and textual information by pretraining GPT2, a transformer-based language model, with the code-switching ASR decoding lattice. Our work achieves around 2 point absolute word error rate reduction from the baseline *n*-gram language model, and 0.33 point absolute reduction from the lattice-rescored baseline word error rate.

Keywords　code-switching, speech recognition, language model, attention mechanism

1. Introduction

The worldwide use of English requires people to learn English either as their mother tongue or secondary language. It contributes to the increase of multilingual speaker nowadays. When making a conversation, bilingual speakers may alternate between their mother tongue and their other language to smoothen the communication flow. This phenomenon is called code-switching. Intra-sentential switching [22], which is code-switching that occurs within a sentence, prevalently appears in conversations. The example of code-switching is Mandarin-English code-switching within Chinese community in Singapore and Malaysia [1]. The following sentence is an example of Mandarin-English code-switched sentence.

我四年完全没有 speak in Chinese	(I haven't spoken Chinese at all in four years)

Note that code-switching is neither creole, nor pidgin languages since creole and pidgins are stable languages [21]. Loanwords are also not categorized as code-switching since they have already adopted and become a part of a language.

Code-switching poses some challenges acoustically and linguistically to automatic speech recognition (ASR) system. Two main challenges of code-switching are the accented speech and data scarcity. The accented speech leads to pronunciation similarity of words or phrases from both languages. For example, "the" and "的" (de), "you" and "有", "water" and "我的" (wo de), etc. In addition, there are many monolingual resources, but there are not many code-switching resources available.

To deal with the pronunciation variation problem, [26] utilizes a bilingual phone set containing the words from both languages. Another approach is to add some possible pronunciation variations of some words to the pronunciation dictionary. These two methods are commonly applied to build a code-switching speech recognition system [23-26].

Overcoming the data scarcity problem, previous studies have attempted to improve the code-switching language modeling. Expanding the vocabulary [3] is the simplest one used to avoid out of vocabulary. [2, 4, 19] has tried to include linguistic information such as word class as a supporting feature in addition to words. [5] has attempted to leverage the monolingual corpora to pretrain the language model. Recently, [6, 7] have tried to perform data augmentation by generating artificial code-switching sentences by utilizing parallel monolingual corpora. A machine translation method can be used to align words and phrases in monolingual parallel text [6]. [7] employs a sequence-to-sequence model to generate the code-switching text.

However, even though these attempts are able to improve the code-switching speech recognition performance, they do not cover the pronunciation variation when transitioning

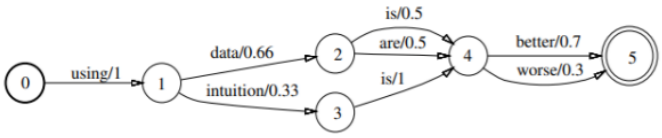


Figure 1. Word lattice example with word probability scores on each graph edge [8]

from one word to another. Consequently, they do not take into account the surrounding acoustic context of words with similar pronunciations. This is because increasing the amount of code-switching text only increases the variation of words. Increasing the pronunciation variation in the pronunciation dictionary only handles the pronunciation problem within a word.

To overcome the pronunciation variation in word transition, we propose to add acoustic information in addition to textual information in language modeling. More specifically, we introduce ASR decoding lattice or ASR hypotheses graph which includes both acoustic and textual information for word transitions as data augmentation in code-switching language modeling. We utilize a transformer-based language model, GPT2 which successfully achieves the state-of-the-art performance as our baseline. We train and evaluate our proposed model using the SEAME Mandarin-English code-switching speech corpora.

2. Previous Studies

2.1. Code-switching Speech Recognition

A conventional automatic speech recognition (ASR) system consists of three main components, a pronunciation dictionary, an acoustic model, and a language model. In a code-switching speech recognition system, the pronunciation dictionary or lexicon contains a list of words from both languages with their pronunciations. Therefore, the acoustic units modeled by the acoustic model are the combination of those for both languages.

Related to the acoustic units for code-switching speech recognition, [27, 28] create a phoneme mapping rules from one language to another. The pronunciation dictionary in [2] uses a merged phone set from both languages in code-switching. [26] builds a bilingual phone set containing phone sets from both languages, and this approach is commonly applied for code-switching speech recognition. DNN architecture can jointly learn the phone set from each language, thus cross-lingual phone merging is not a mandatory [26].

Dealing with data scarcity problem in acoustic model training, [29, 30] perform data augmentation by employing monolingual speech corpora for training. This approach may be beneficial for recognizing monolingual speech segments in code-switching speech. However, code-switching data is still needed to model acoustic change in word transition. [2, 3] introduce language identifier (LID) to help code-switching acoustic modeling tasks. Although including LID improves the ASR performance, we need to optimize the LID model which is also challenging.

The availability of code-switching text resources is also less than monolingual ones since code-switching occurs mostly in verbal communication. Previous studies have strived to enhance the code-switching language modeling. Expanding the vocabulary [3] and including linguistic information as additional features for language modeling have been attempted in [2, 4, 19]. The linguistic information can be part-of-speech tags [2, 19], trigger words [2], or general word classes [4]. [5] has tried to make use of monolingual corpora for language model pretraining. [2,6,7] generates artificial code-switching sentences for data augmentation. [6] employs machine translation on monolingual parallel text for sentence generation, while [2] leverages part-of-speech tags and trigger words to support code-switching sentence generation. [7] utilizes sequence-to-sequence model trained on parallel corpora to generate the sentences.

These methods increase the ability of the language model to predict more word variations. However, linguistic information depends on the language and may be only useful for languages with many available resources. Machine translations may translate words not at the proper places according to some constraints [20]. The sentences generated by a trained model may contains some repetitive words [7], which may be harmful for the language modeling.

2.2. Automatic Speech Recognition Decoding Lattice

The text output of a speech recognition system is decoded by combining the acoustic unit sequences in the dictionary, the acoustic model output, and the language model [8]. By combining all of them, the ASR system has many alternative word sequences as its hypotheses. This ASR hypotheses can also be expressed as a directed acyclic graph whose paths from the first to the last node represent word sequences. This graph is called decoding lattice. The example in Figure 1 shows a word lattice with six nodes and eight arcs that includes the words with their probability score. The probability score is the sum of acoustic model score, language model score, and pronunciation score.

2.3. GPT2 Language Model

Recent works show that transformer-based architecture [9] achieves the state-of-the-art performance in many tasks. GPT2 [10] is a transformer-based language model developed by OpenAI and it outperforms many language models on many natural language processing tasks for English. The attention mechanism is the key component that leads to their success.

The transformer is made of a stack of transformer blocks. Each block consists of an attention block that compute the attention scores of an input item against all input items and a feed forward layer. In language modeling, the input for the transformer is a sentence. In other words, the attention mechanism calculates the score of a word against all other words in a sentence. The attention block calculates the score of a word or a query Q against all words symbolized as keys K. The score is obtained by multiplying vector Q by each vector in K (divided by the model embedding size d) followed by softmax operation. Afterwards, the score is multiplied by each word representation vector V in the input sentence. The following are definition of components in a transformer block [9].

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right)V \qquad 2.1$$

$$\text{FeedForward}(x) = \max(0, xW_1 + b_1)\,W_2 + b_2 \qquad 2.2$$

$$\text{softmax}(x_{\text{class}}) = \frac{\exp(x_{\text{class}})}{\sum_c^{C_t}\exp(x_c)} \qquad 2.3$$

In the definition of feed forward layer, x is the output from the attention layer. W_1, W_2, b_1, and b_2 are the weight and the bias parameters of the feed forward layer.

GPT2 adopts the multi-head attention mechanism which divides the attention into several attention heads. It enables the use of information from the subspace representation of each word.

Attention mechanism also does not consider the sequence ordering [9]. So, word position information is included in addition to words as the input of GPT2. In addition, GPT2 is a causal language model. Thus, it calculates the attention scores of a word against its predecessor words within a context window with size of c. The causal language modeling objective is to maximize the log likelihood of the input sequence probability. The objective function for a word sequence $(t_1, \dots, t_m)$ is shown in equation 2.4.

$$\text{L}(t_1, \dots, t_m) = \sum_{\text{i=1}}^{\text{m}} \log \text{P}(t_i | t_{i-c}, \dots, t_{i-1}) \qquad 2.4$$

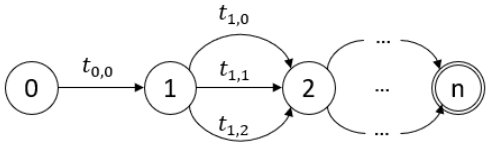


Figure 2. Word confusion network example

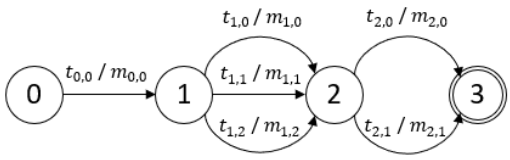
3. Including Acoustic and Textual Information in Language Modeling

The lack of code-switching text resources poses a challenge to code-switching language modeling. Meanwhile, language model is one of the important aspects that leads to the successful code-switching speech recognition since code-switching is a linguistic phenomenon, and the word switch follows some constraints [22]. Therefore, we focus on improving the code-switching language modeling. We utilize GPT2 as our baseline due to the state-of-the-art performance achieved in language modeling.

In this section, we elaborate our proposed method of including acoustic and textual information in language modeling. We propose to include these both information by using ASR decoding lattices as data augmentation for language model pretraining. We hypothesize that including both acoustic and textual information can help solving the pronunciation variation problem in word transitions, and therefore it leads to the improvement of the ASR performance. As far as we know, this is the first time ASR decoding lattice is used in code-switching language modeling. The language model architecture is based on GPT2.

The position of each word is needed in causal language modeling using transformer-based architecture. However, the lattice structure is complex because a node in the graph can have more than one arc with different destination nodes. In Figure 1, State 1 have two arcs that point to States 2 and 3. Therefore, it is hard to determine the position of a word in a sequence for the language modeling since there are many possible paths, and consequently, each word may have more than one possible position.

To solve the problem, we simplify the lattice structure into a word confusion network structure. Word confusion network is also a graph that has nodes and arcs, but the arcs from each node pointing to only one destination node. Thus, a word confusion network has a sausage-like shape such as shown in Figure 2. By converting the lattice into word



$T = t_{0,0}, t_{1,0}, t_{1,1}, t_{1,2}, t_{2,0}, t_{2,1}$
$P = 0, 1, 1, 1, 2, 2$
$M = m_{0,0}, m_{1,0}, m_{1,1}, m_{1,2}, m_{2,0}, m_{2,1}$

Figure 3. Model input example representing a word confusion network

confusion network, we can determine the position of each word for the causal language modeling. We convert the lattice into word confusion network using the same algorithm as in [11].

Since a state transition in word confusion network may have multiple alternatives, we design the input for the GPT2 model. Using the same word sequence notation as before, the input sequence becomes $T = t_{0,0}, \dots t_{0,K_0-1}, \dots, t_{j,k_j}, \dots, t_{n-1,K_{n-1}-1}$, where n is the number of states, $j \in [0, \mathrm{n}-1]$ is the state number, K_j is the number of word alternatives in state j, and $k_j \in [0, K_j]$. Next, the word position identifications P is the state numbers of the input sequence T. Finally, we define an attention mask M using the word transition score to include the acoustic and textual information in language modeling. Figure 3 shows the example of T, P, and M for a word confusion network. By applying attention mask M, we add the mask M before applying the softmax operation in equation 2.1. Therefore, the new formula can be written as follows.

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\top} + \boldsymbol{M}}{\sqrt{d}}\right)V \qquad 3.1$$

During training, the language model objective is modified from equation 2.4 corresponding to the change of model input. The training optimizes the probability of a word with the highest transition score in the word confusion network, given all words in all predecessor states. The following is the definition of the new objective function.

$$\mathrm{L}\left(t_{0,0}, \dots, t_{n-1,K_{n-1}-1}\right) = \sum_{\mathrm{j}=1}^{\mathrm{n}-1} \log \mathrm{P}\left(t_{j,k_{best}} | t_{j-c}, \dots, t_{j-1}\right) \qquad 3.2$$

$$\mathrm{k_{best}} = \underset{k_j \in \{0,\dots,K_j\}}{\arg\max} \left(m_{j,k_j}\right) \qquad 3.3$$

4. Experiments

4.1. Datasets

The speech corpus used in this work is the South-East Asian Mandarin English (SEAME) code-switching corpus [1, 12]. This corpus contains Mandarin and English code-switched speech from Singapore and Malaysia. It represents the Mandarin-English code-switching that commonly occurs in conversations among the Chinese community in Singapore and Malaysia. The data is 112.6 hours of recorded spontaneous speech including the transcription, and the characteristic of code-switching in this dataset is Mandarin dominant. It means that there are more sentences with more Mandarin words than English words in this corpus.

Table 1. SEAME dataset statistics [12]

Split name	# Speakers	Duration	Duration ratio (%)		
			Man	Eng	CS
train	134	101.13	16	16	68
dev man	10	7.49	14	7	79
dev sge	10	3.93	6	41	53

Table 2. Monolingual text corpora statistics

Dataset name	# utterances	# words
AISHELL2	1M	6.8M
TEDLIUM-3	268K	5M

For training and evaluation, we use the same data division as in [3]. The data are divided into a training set and two dev sets. The training set is used to train the acoustic model and language model in the ASR, while the other two dev sets are used for evaluation. The dev sets are the Mandarin dominant dev set (dev man) and the English dominant dev set (dev sge). The Mandarin dominant set contains more Mandarin words, and the English dominant set contains more English words. The statistics of SEAME dataset are presented in Table 1.

In addition to the transcription of SEAME corpus, we also use two other monolingual text corpora to for language model pretraining. The monolingual Mandarin text corpus is the transcription of AISHELL-2 [13] and the monolingual English text corpus is the transcription of TEDLIUM-3 [14]. AISHELL-2 is originally a speech corpus designed for speech recognition in many domains such as sports, music, science and technology, etc. TEDLIUM-3 is also a speech corpus that is built from TED conference videos. The statistics of these two monolingual datasets are shown in Table 2.

4.2. Settings

We build the classical ASR utilizing Kaldi toolkit [15]. This toolkit is used for the speech feature extraction, acoustic modeling, and speech decoding. The acoustic model trained in this work is based on the Hidden-Markov-Model (HMM) and Time-delay neural network (TDNN). The TDNN inputs are 40-dimensional MFCC and 100-dimensional iVector.

To utilize the transformer-based language model, we perform two-pass decoding. We employ n-gram language model with 4 as the n value for the first pass decoding followed by lattice rescoring with the new language model. The n-gram is computed using SRILM [16], and the new language model is implemented based on OpenAI GPT2 publicly available in the Huggingface library [17]. We utilize Pytorch machine learning framework [18] for the language model training. The configurations of the language model are set to be the same as the small-sized GPT2 [10]. The number of both hidden layers and attention heads is 12, the size of both embedding and hidden layer is 768, and the causal language model context size is 1024.

We train the language model in four scenarios. The first two scenarios are the baselines trained using only code-switching dataset. The first one (SEAME 4-gram) uses 4-gram language model. The second one (SEAME GPT2) uses GPT2 language model to rescore the first pass decoding lattice. The third scenario (+Mono) employs the monolingual datasets for GPT2 language model pretraining before training it with the code-switching data. The last scenario (+Mono+WCN) uses monolingual datasets and word confusion networks for pretraining. The word confusion networks are obtained by decoding the code-switching training set using the ASR with the first baseline language model. For the evaluation, we calculate the perplexity to measure the language model performance and the word error rate (WER) for the ASR performance. For both metrics, lower perplexity and word error rate values indicate better language model and ASR performances, respectively.

4.3. Results

The experiment results are shown in Table 3. The results exhibit improvements for both language model and ASR performances after introducing monolingual corpora for the language model pretraining. Moreover, the best overall performances for both language model and ASR are achieved after introducing word confusion networks. Our proposed method successfully improves the word error rate of evaluating the ASR on the Mandarin dominant set by the absolute reduction of 1.94 point from the first baseline and 0.33 point from the second baseline. The word error rate evaluated on the English dominant set is not improved after including the word confusion network compared to the third scenario, although it still performs better than the second baseline and is comparable to the third scenario.

We also observe the word substitution error for ASR after rescoring with the GPT2 language model. The results presented in Table 4 (dev man) and Table 5 (dev sge) exhibit the best overall substitution error for our proposed method. Introducing acoustic and textual information through word confusion network reduces both cross-

Table 3. ASR and language model experiment results

Model	Perplexity		WER (%)	
	dev man	dev sge	dev man	dev sge
SEAME 4-gram	285.74	233.62	24.28	32.97
SEAME GPT2	183.43	133.68	22.67	31.18
+Mono	183.19	132.96	22.47	30.87
+Mono+WCN	181.89	130.79	22.34	30.88

Table 4. Word substitution errors (dev man)

Model	M→E (%)	E→M (%)	M→M (%)	E→E (%)	#Subs.
SEAME GPT2	2.81	7.42	10.08	13.30	14457
+Mono	2.73	7.46	10.05	13.17	14353
+Mono+WCN	2.71	7.34	9.89	13.02	14159

M→E: Mandarin to English substitution; E→M: English to Mandarin substitutions; M→M: monolingual Mandarin substitutions; E→E: monolingual English substitutions

Table 5. Word substitution errors (dev sge)

Model	M→E (%)	E→M (%)	M→M (%)	E→E (%)	#Subs.
SEAME GPT2	4.73	4.53	10.26	18.26	11572
+Mono	4.66	4.60	10.41	17.96	11509
+Mono+WCN	4.71	4.51	10.28	18.04	11484

M→E: Mandarin to English substitution; E→M: English to Mandarin substitutions; M→M: monolingual Mandarin substitutions; E→E: monolingual English substitutions

Table 6. Absolute WER reduction comparison with other works

Model	WER (SEAME only) (%)	WER (after rescoring) (%)	Absolute reduction (%)
Word LM + Class LM [4]	25.74	25.65	0.09
Vocab expansion [3]	25.10	25.00	0.10
+Mono [5]	22.67	22.47	0.20
Synthetic CS [6]	24.11	23.80	0.31
S2S [7]	22.40	22.10	0.30
Ours (+Mono+WCN)	22.67	22.34	0.33

lingual and monolingual substitutions, although we can see some slightly worse results compared to the model pretrained only using monolingual dataset in dev sge. Moreover, the word confusion network is shown to reduce the Mandarin monolingual substitutions significantly after the monolingual pretraining so that it achieves comparable result with the baseline. The worst monolingual Mandarin substitutions by the monolingual pretrained model might be caused by some bias towards the monolingual dataset learned during the monolingual pretraining.

We also show the comparison of word error rate improvement of ASR from different previous studies in Table 6. The results are obtained from the evaluation with the Mandarin dominant set. Our proposed method achieves the largest word error rate absolute reduction among the works that we compare. Compared to data augmentation by using generated code-switching sentences [6,7], our approach achieves a slightly higher but still comparable word error rate absolute reduction. For [6] and [7], the data splits are different but they have a similar distribution to the Mandarin dominant set. [6] and [7] utilize parallel monolingual data to generate new code-switching sentences. Meanwhile, our method only uses additional code-switching data for training from the word confusion networks which are originated from the SEAME training set. It means that including acoustic and textual information in language modeling is beneficial, even with the limitation of word switching variations.

Additionally, we also present some qualitative results in Table 7. The results show the improvement of recognizing

Table 7. Examples of speech recognition output
*) Ref is the reference sentence.

ref	(呃) 不 是 locals 很 多 foreigners 我 的 level
SEAME 4-gram	(呃) 不 是 locals 很 多 foreigners '***' water level
SEAME GPT2	(呃) 不 是 locals 很 多 foreigners for the level
+Mono	(呃) 不 是 locals 很 多 foreigners pro level
+Mono +WCN	(呃) 不 是 locals 很 多 foreigners 我 的 level
ref	curry veggie and then 一 个 好 像 more on (er) 就 是 几 个 veggie 这 样 你 才 一 起 两 块 多 一 餐 这 样 [咯]
SEAME 4-gram	curry wednesday and then 有 一 个 好 像 more on a 几 时 几 个 veggie '***' there 你 在 一 起 两 块 多 一 餐 '***' '***' 痒
SEAME GPT2	curry weijie and then 有 一 个 好 像 more on (er) 几 十 几 个 veggie '***' then 你 在 一 起 两 块 多 一 餐 这 样 '***'
+Mono	curry weijie and then 一 个 好 像 more on (er) 几 十 几 个 veggie '***' then 你 在 一 起 两 块 多 一 餐 '***' 样 '***'
+Mono +WCN	curry veggie and then 一 个 好 像 more on (er) '***' '***' 几 个 veggie '***' then 你 在 一 起 两 块 多 一 餐 这 样 '***'

several Mandarin and English words. Our method successfully recognizes the word “我的” and “veggie”, while the others recognize it as other words. These results exhibit the effectiveness of our proposed method for solving the pronunciation variation problem by taking into account the acoustic and textual information in word transitions.

5. Conclusion

We propose to incorporate ASR decoding lattice in code switching language modeling to alleviate the pronunciation variation problem in word transitions. The ASR decoding

lattice contains both acoustic and textual information represented by word probabilities. The experiment results show that our method outperforms the GPT2 baseline and achieve the largest word error rate absolute reduction compared to several previous researches. Supporting the quantitative results, the qualitative results shows that our method can recognize some words that are wrongly predicted by other models. With the limitation of word switching variation, introducing both acoustic and text information in language modeling is shown to be effective.

Apart from the unseen code-switching variation, our approach does not consider the previous acoustic context of a word. Therefore, investigating the effectiveness of utilizing bidirectional model and artificial code-switching sentences to extend our method is left for future work.

6. Acknowledgement

This work was supported by JST CREST JPMJCR1687, JSPS KAKEN 16H02845.

References

[1] D. Lyu, T. Tan, Chng Eng Siong, and Haizhou Li. Seame: a mandarin-english code-switching speech corpus in south-east asia. In INTERSPEECH, 2010.

[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 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4889–4892, 2012.

[3] Zhiping Zeng, Yerbolat Khassanov, V. Pham, Haihua Xu, Chng Eng Siong, and Haizhou Li. On the end-to-end solution to mandarin-english code-switching speech recognition. In INTERSPEECH, 2019.

[4] Zhiping Zeng, Haihua Xu, Tze Yuang Chong, Eng-Siong Chng, and Haizhou Li. Improving n-gram language modeling for code-switching speech recognition. In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pages 1596–1601, 2017.

[5] E. Yilmaz, et. al. (2019), Multi-Graph Decoding for Code-Switching ASR, INTERSPEECH 2019

[6] Grandee Lee, Xianghu Yue, and Haizhou Li. Linguistically motivated parallel data augmentation for code-switch language modeling. In INTERSPEECH, 2019.

[7] Chia-Yu Li and Ngoc Thang Vu. Improving Code-Switching Language Modeling with Artificially Generated Texts Using Cycle-Consistent Adversarial Networks. In Proc. Interspeech 2020, pages 1057–1061, 2020

[8] Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech Recognition with Weighted Finite-State Transducers, pages 559–584. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[9] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. ArXiv, abs/1706.03762, 2017.

[10] Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[11] Lidia Mangu, Eric Brill, Andreas Stolcke, Finding consensus in speech recognition: word error minimization and other applications of confusion networks, Computer Speech & Language, Volume 14, Issue 4, 2000, Pages 373-400,

[12] Grandee Lee, Thi-Nga Ho, Eng-Siong Chng, and Haizhou Li. A review of the mandarin-english code-switching corpus: Seame. In 2017 International Conference on Asian Language Processing (IALP), pages 210–213, 2017.

[13] Jiayu Du, X. Na, Xuechen Liu, and Hui Bu. Aishell-2: Transforming mandarin asr research into industrial scale. ArXiv, abs/1808.10583, 2018.

[14] François Hernandez, Vincent Nguyen, Sahar Ghannay, N. Tomashenko, and Y. Estève. Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation. ArXiv, abs/1805.04699, 2018.

[15] Daniel Povey, A. Ghoshal, Gilles Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlícek, Y. Qian, Petr Schwarz, J. Silovský, G. Stemmer, and Karel Veselý. The kaldi speech recognition toolkit. 2011.

[16] A. Stolcke. Srilm - an extensible language modeling toolkit. In INTERSPEECH, 2002.

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771, 2019.

[18] 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.

[19] Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Codeswitching language modeling using syntax-aware multi-task learning. In Proceedings of the Third Workshop on Computational Approaches to Linguistic CodeSwitching, pages 62–67, Melbourne, Australia, July 2018. Association for Computational Linguistics

[20] Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1543– 1553, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[21] J. Arends, P. Muysken, and N. Smith. Pidgins and Creoles: An Introduction. Creole language library. J. Benjamins, 1995.

[22] Shana Poplack. "sometimes i'll start a sentence in spanish y termino en español" : Toward a typology of code-switching. Linguistics, 51:11 – 14, 1979.

[23] S. Yu, S. Hu, S. Zhang, B. Xu, Chinese-English bilingual speech recognition, in: Proceedings of International Conference on Natural Language

Processing and Knowledge Engineering, IEEE, 2003.

[24] J. Y. Chan, H. Cao, P. Ching, T. Lee, Automatic recognition of CantoneseEnglish code-mixing speech, International Journal of Computational Linguistics & Chinese Language Processing, Volume 14, Number 3, September 2009.

[25] S. Sivasankaran, B. M. L. Srivastava, S. Sitaram, K. Bali, M. Choudhury, Phone Merging for Code-Switched Speech Recognition, in: Proceedings of the Third Workshop on Computational Approaches to Linguistic CodeSwitching, 2018, pp. 11–19.

[26] Pengcheng Guo, Haihua Xu, Lei Xie, Eng Siong Chng, Study of Semi-supervised Approaches to Improving English-Mandarin Code-Switching Speech Recognition. INTERSPEECH 2018, pp. 1928-1932

[27] Basem Ahmed and Tien-Ping Tan. Non-native accent pronunciation modeling in automatic speech recognition. 11 2011.

[28] B. M. L. Srivastava and Sunayana Sitaram. Homophone identification and merging for code-switched speech recognition. In INTERSPEECH, 2018.

[29] Ayushi Pandey, Brij Mohan Lai Srivastava, and Suryakanth V Gangashetty. Adapting monolingual resources for code-mixed hindi-english speech recognition. In 2017 International Conference on Asian Language Processing (IALP), pages 218– 221, 2017.

[30] Emre Yilmaz, H. V. D. Heuvel, and D. V. Leeuwen. Acoustic and textual data augmentation for improved asr of code-switching speech. In INTERSPEECH, 2018.