論文 / 著書情報
Article / Book Information

Title	Lattice-Based Data Augmentation for Code-Switching Speech Recognition
Author	Roland Hartanto, Kuniaki Uto, Koichi Shinoda
Journal/Book name	Proceedings of 2022 APSIPA Annual Summit and Conference, , , pp. 1667-1672
Pub. date	2022, 11
DOI	https://doi.org/10.23919/APSIPAASC55919.2022.9980277
Copyright	(c)2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Note	This file is author (final) version.

Lattice-based Data Augmentation for Code-switching Speech Recognition

Roland Hartanto, Kuniaki Uto, Koichi Shinoda
Tokyo Institute of Technology, Tokyo, Japan
E-mail: roland@ks.c.titech.ac.jp, uto@ks.c.titech.ac.jp, shinoda@c.titech.ac.jp

***Abstract*— Code-switching is a common phenomenon that occurs within conversations among multilingual speakers. The limited availability of code-switching resources poses some challenges to code-switching speech recognition. Our work addresses both data scarcity and pronunciation variations in word transitions by introducing speech recognition decoding lattice for data augmentation in code-switching speech recognition, specifically in language modeling. Decoding lattices contain both acoustic and textual information that help solve the pronunciation variations problem. We pretrain GPT2, a transformer-based language model, with lattices obtained from the first-pass decoding of code-switching training data. The first-pass decoding is performed by using the baseline speech recognition system with n-gram language model. We successfully reduce around 2 point of word error rate from the previously mentioned baseline and 0.33 point from the baseline that utilizes GPT2 language model. Ablation study also shows an improvement when including acoustic information for code-switching language model pretraining. In addition, we show that despite having a limited amount of word switching variations information, our proposed method achieves a comparable result with previous studies that employ artificial code-switching sentences.**

I. INTRODUCTION

Code switching is a linguistic phenomenon that occurs when multilingual speakers use two or more languages alternately in conversations. The speakers frequently mix the usage of multiple languages in a sentence (intra-sentential [1]). The use of Mandarin and English alternately when conversing which prevalently occurs in Singapore and Malaysia [2] is one of the code-switching examples.

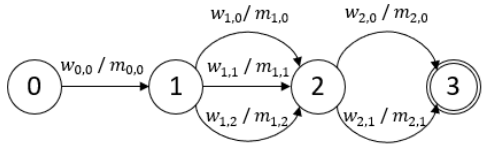
Developing an automatic speech recognition (ASR) for code-switched speech is challenging due to the existence of several challenges. The main challenges are accented speech and data scarcity. Bilingual speakers may have accents when they are pronouncing words that are not their mother tongue. Moreover, it causes some pronunciation variations for some words. Some words may be pronounced similarly with other words from all the languages used. When switching from one language to another, the pronunciation of some words is affected by the previous word spoken. In addition, there exists a limited amount of code-switching resources.

There are several approaches that deal with the pronunciation variations problem. A bilingual phone set from both languages in code-switching is created in [3]. Recently, it has become possible to jointly learn the phone sets of the two languages by using DNN architecture, thus it enables us to utilize a bilingual phone set. Another common approach is to make a pronunciation dictionary with possible pronunciation variations for some words [4-6].

There are also previous studies that have attempted to overcome the data scarcity problem by improving the code-switching acoustic modeling. [7, 8] perform data augmentation by incorporating monolingual speech resources in training. This method may be useful to improve the recognition of monolingual speech segments. However, code-switching corpus is necessary to model the acoustic transition in word switching. [9, 10] train a language identification model to support code-switching acoustic modeling. The attempts successfully improve the ASR performance, but optimizing the language identification model is challenging.

The limited availability of code-switching text resources also poses a challenge in code-switching speech recognition. Language model is one of the important components that leads to the successful code-switching speech recognition. Thus, many previous studies have also attempted to enhance the data scarcity problem for code-switching language modeling. Some researchers have attempted to perform vocabulary expansion [10] and make use of additional linguistic features as additional inputs for language modeling [9, 11, 12]. [13] has attempted to utilize monolingual corpora for model pretraining. [14, 15] perform data augmentation by employing generated artificial code-switching sentences. [14] makes use of parallel text corpora to train a sequence-to-sequence model to generate code-switching text. [15] employs machine translation to align words and phrases in parallel texts to make new artificial code-switching sentences. The previous attempts indeed increase the code-switching ASR performance. However, they still have some limitations. Linguistic information is language dependent and the use of it may be useful only for languages with large resources. Artificial code-switching sentences may contain some repetitive words [14], which may degrade language models.

Related to language modeling, [16] investigates the benefit of utilizing acoustic units for language modeling for ASR. It finds that multilingual phoneme-level language modeling generalizes the model better over multiple languages than character-based language modeling for low resource language. Hence, acoustic information may be useful to learn cross-lingual representation for code-switching ASR and help solve the pronunciation variations problem. Another study [17] utilizes lattice structures for speech translation tasks. A lattice contains multiple alternative sequences, although it is noisy. It



$$W = w_{0,0}, w_{1,0}, w_{1,1}, w_{1,2}, w_{2,0}, w_{2,1}$$
$$T = 0, 1, 1, 1, 2, 2$$
$$M = m_{0,0}, m_{1,0}, m_{1,1}, m_{1,2}, m_{2,0}, m_{2,1}$$

Fig. 1 *An example of word confusion network for model input.*

is useful to alleviate the data scarcity problem, and it is not language dependent. If we use lattices decoded by an ASR system, then the lattices contain acoustic information in their word transition scores. Therefore, ASR lattices can be useful to alleviate both data scarcity and pronunciation variations problems for code-switching ASR.

In this work, we attempt to include acoustic information of word transitions in language modeling to deal with the variations of pronunciation in word transition. We utilize ASR decoding lattices for data augmentation in code-switching language modeling. We employ GPT2 [18], a transformer-based causal language model, as our baseline language model as it achieves the state-of-the-art performance in language modeling. We pretrain GPT2 with decoding lattices before training it with code-switching sentences. We conduct our experiments on the South-East Asia Mandarin-English (SEAME) code-switching speech corpus [2]. We, as far as we know, are the first who attempt to use decoding lattices to support code-switching language modeling.

II. Lattice-based Data Augmentation

A. Transformer-based language model

A transformer-based language model created by OpenAI, called GPT2 [18], outperforms many models on various natural language processing tasks. Previous studies have shown the effectiveness of transformer architecture [19] in achieving the state-of-the-art performance in various tasks. The key to their success is the attention mechanism. Attention mechanism calculates the importance score of a word query (Q) with respect to all words in the input sequence indexed by key vectors (K). This score is also called attention score. The attention score is then multiplied by each word representation vectors (V) of the input sequence. The attention mechanism can be expressed as follows, with d as the model embedding size.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right)V. \qquad (1)$$

GPT2 is a causal language model. It means that the architecture works in one direction, and the attention mechanism considers only the previous words within a context window with the length of c for each word. Word position information is usually added together for each word since attention mechanism does not consider the sequence order. The objective function of causal language modeling is to maximize the likelihood of the probability of input sequence $(w_1, \dots, w_m)$. The objective function can be written as follows:

$$L(w_1, \dots, w_m) = \sum_{i=1}^{m} log\ P(w_i | w_{i-c}, \dots, w_{i-1}). \qquad (2)$$

B. Acoustic and Textual Information from Lattices for Code-switching Language Modeling

A conventional speech recognition system decodes audio to text by employing the acoustic model output, the language model, and the acoustic unit sequences from the pronunciation dictionary. Consequently, a speech recognition system has many possible hypotheses which are represented as a directed acyclic graph called a decoding lattice. A lattice has many possible paths that contain words and their transition probability. The transition probability is calculated from the acoustic model, the pronunciation dictionary, and the language model scores. Thus, it contains acoustic information from the acoustic model score and textual information from the language model score.

However, a lattice has a complex structure because each node may have multiple destination nodes, and thus it is difficult to define the position of each word. Since the position information is needed in language modeling using a transformer-based language model, we convert a lattice to a word confusion network. The structure of a word confusion network is simpler than a lattice because each node has arcs that end at only one destination node. Hence, we can define the position of each word in the graph.

To utilize the word confusion network, we define the input for GPT2 language model. An input example is shown in Figure 1. For an n-state word confusion network, we symbolize a word sequence as $W = w_{0,0}, \dots w_{0,K-1}, \dots, w_{i,k}, \dots, w_{n-1,K-1}$, where $i \in [0, n-1]$ is the state index, K is the number of state transitions from each state, and $k \in [0, K]$is the state transition index. We define the word position indices as T, and a word position index is the same as the origin state index of a word. Then, we use the transition probabilities as attention mask M, and the mask is added during the attention score calculation. Accordingly, Equation (1) is rewritten as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top} + \boldsymbol{M}}{\sqrt{d}}\right)V. \qquad (3)$$

Corresponding to the new input design, we also define a new objective function to train the language model using word confusion networks. The function maximizes the likelihood of a word sequence by computing the probability of a word with the highest transition score ($w_{i,k_{\text{best}}}$) for each state given all words from all previous states. The new objective function can be expressed as follows:

$$L(w_{0,0}, \dots, w_{n-1,K-1}) = \sum_{i=1}^{n-1} \log P(w_{i,k_{\text{best}}} | w_{i-c}, \dots, w_{i-1}) \quad (4)$$

$$k_{\text{best}} = \arg\max_{k \in \{0,\dots,K\}} (m_{i,k}). \quad (5)$$

III. EXPERIMENTS

A. Datasets

We utilize SEAME (South-East Asia Mandarin-English) code switching speech corpus [2] for our experiments. This dataset contains Mandarin and English code-switched spontaneous speech recorded in conversational and interview scenarios. The code-switching in this dataset is Mandarin dominated [20]. In other words, the sentences in this corpus contain more Mandarin words than English words. The total duration of the recordings is 112.6 hours. The transcription is also included in this dataset.

For ASR and language model training and evaluation, the data splits from [10] are employed. The splits consist of a training set and two evaluation sets. We use the training set to train both the acoustic and language models of the ASR. The evaluation sets are a Mandarin dominated set, referred to as dev man in Table 1, and an English dominated set, referred to as dev sge. We show the statistics of all data splits in Table 1. The columns #Man, #Eng, and #CS exhibit the proportion of the number of monolingual Mandarin, monolingual English, and code-switching utterances.

For training the transformer-based language model, we employ Mandarin and English monolingual text datasets in addition to SEAME. We use the transcription of AISHELL-2 speech corpus [21] for Mandarin monolingual text and the transcription of TEDLIUM-3 speech corpus [22] for English monolingual text. The statistics of these datasets are shown in Table 2. These monolingual datasets are employed for language model pretraining. We pretrain the language model before training it with the word confusion networks extracted from ASR decoding lattices.

B. Setup

We employ Kaldi toolkit [23] for feature extraction, acoustic modeling, and building the deep neural network (DNN) - Hidden-Markov-Model (HMM) ASR system. The acoustic model is time delay neural network (TDNN) [24]. The input features for the acoustic model are 40-dimensional MFCC and 100-dimensional i-vectors. The number of hidden layers of the TDNN is 6. For the pronunciation dictionary, we use a bilingual phone set that consists of Mandarin initial-final tonal phones from AISHELL2 corpus [19] and English phones from the CMU Pronouncing Dictionary.

We perform two-pass decoding. The first-pass decoding uses *n*-gram language model with n equals to 4. The *n*-grams are computed using SRILM toolkit [25]. Afterwards, lattice rescoring is performed by using the transformer-based language models. The implementation of the transformer-based language model is based on GPT2 architecture available in the Huggingface library [26]. The language model configuration used is the same as the small GPT2 model [18], with 12 hidden layers and attention heads. Both embeddings and hidden layer have the size of 768, and the context size for the causal language model is 1024.

For evaluation, we compare five scenarios. We train two baselines as the first two scenarios. The first baseline is the ASR with n-gram language model before lattice rescoring, referred as SEAME 4-gram in Table 3. The second baseline is the ASR after lattice rescoring with the new language model, referred to as SEAME GPT2. Both baselines use only SEAME transcript for language model training. The next scenario uses the language model pretrained using monolingual text datasets, referred to as "+Mono". Our proposed method, referred to as "+Mono+WCN", employs the language model pretrained by monolingual datasets and word confusion networks obtained

Table 1. SEAME corpus statistics

Data	#utterances	#Man (%)	#Eng (%)	#CS (%)
train	97,293	20	25	55
dev man	6,531	19	14	67
dev sge	5,321	8	52	40

Table 2. Monolingual text corpus statistics

Dataset Name	.#utterances	#words
AISHELL2 [19]	1M	6.8M
TEDLIUM-3 [20]	268K	5M

Table 3. Language model and speech recognition 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 (w/o acoustic score)	181.51	130.90	22.46	30.81
+Mono+WCN	181.89	130.79	22.34	30.88

Table 4. Word substitution error rate on 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 (w/o acoustic score)	2.75	7.38	9.96	13.22	14298
+Mono+WCN	2.71	7.34	9.89	13.02	14159

Table 5. Word substitution error rate on 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 (w/o acoustic score)	4.74	4.63	10.24	17.92	11480
+Mono+WCN	4.71	4.51	10.28	18.04	11484

Table 6. A speech recognition example (Ref is the reference sentence. Bolded words are misrecognized words.)

Model/ref	Sentence
Ref	… 希望就是每个 weekend ah 都是能有 gathering ah 这样的东西 [啊] 到处跑到处去 enjoy 这样 [啦] but 都是要看我的那个 job 以未来的 future jobs scope …
SEAME GPT2	… **也**是每个 weekend [啊] 都是 **in nanyang** gathering **讲**的东西 [啊] 到处**讲出**去 enjoy **讲** but 都是要**靠**我的那个 job **因为 like the** future job scope …
+Mono	… **也**是每个 weekend [啊] 都是 **in nanyang rank** [啊] **讲**的东西 [啊] 到处**讲出**去 enjoy 这样 [啊] but 都是要**靠**我的那个 job 未来的 future job scope …
+Mono+WCN (w/o acoustic score)	… **也**是每个 weekend [啊] 都是 **in 南洋** gathering **讲**的东西 [啊] 到处**讲出**去 enjoy 这样 **[啊]** but 都是要**靠**我的那个 job 未来的 future job scope …
+Mono+WCN	… **也**是每个 weekend [啊] 都是 **in 南洋** gathering ah 这样的东西 [啊] 到处**讲出**去 enjoy 这样 **[啊]** but 都是要**靠**我的那个 job 以未来的 future job scope …

Table 7. The comparison of WER reduction with several previous studies

Model	WER (SEAME only) (%).	WER (after improvement) (%)	Absolute reduction (%)
Word LM + Class LM [11]	25.74	25.65	0.09
Vocab. Expansion [10]	25.10	25.00	0.10
Synthetic CS [15]	24.11	23.80	0.31
S2S [14]	22.40	22.10	0.30
Ours (+Mono+WCN)	22.67	22.34	0.33

from the first pass decoding lattices. The monolingual pretraining is performed due to the noisy ASR decoding lattice [17]. To demonstrate the effect of including acoustic information in language modeling, we conduct an experiment of using the word confusion networks without acoustic model scores, referred to as "+Mono+WCN (w/o acoustic score)". The ASR employed to decode these lattices is the same as the first baseline. In addition, the maximum number of word transitions from state to state in a word confusion network is 3 and is extracted by using Kaldi toolkit [1], where the implementation is based on [27]. The metrics used for the evaluation are perplexity for language modeling and word error rate (WER) for the ASR. Lower perplexity and WER exhibit better performances.

C. Results and Discussion

We present the experiment results in Table 3. From the results, our proposed method gives the best overall results. We achieve the lowest perplexity for both Mandarin and English dominated evaluation splits. Consistently, it also exhibits better ASR performances than both baselines for both data splits. The absolute WER reductions are 1.94 point from the first baseline and 0.33 point from the second baseline. The ASR with language model pretrained only using monolingual datasets (+Mono) also achieves better WER than both baselines, and shows a slightly lower WER for dev sge. However, our model still achieves a comparable WER for dev sge and reduces the WER for dev man even further.

To analyze the results further, we show the word substitution error rates in Table 4 (dev man) and Table 5 (dev sge). The columns M→E and E→M represent the Mandarin to English and English to Mandarin cross-lingual substitution errors, respectively. The columns M→M and E→E represent the monolingual Mandarin and English substitutions, respectively. We can see that the total number of substitutions is the lowest after introducing both acoustic and textual information in language modeling for dev man. Our model reduces not only cross-lingual substitutions but also monolingual substitutions consistently, while +Mono shows unstable results, especially for substitutions towards Mandarin words. This may be caused by the bias introduced by the monolingual datasets during pretraining. However, our model has higher error rates for some monolingual cases in dev sge due to the noise contained in the ASR lattices.

The scenario "+Mono+WCN (w/o acoustic score)" shows a significant higher WER than +Mono+WCN for dev man. Moreover, all substitutions error rates of +Mono+WCN for dev man are significantly lower than the model without acoustic scores. However, for dev sge, the WER and the word substitution error rates of +Mono+WCN are slightly higher than the model with no acoustic scores. In Table 5, we can see that the errors shift from cross-lingual to monolingual substitutions. This ablation study demonstrates that the model pretrained with acoustic information has the ability to differentiate the two languages better than the model pretrained without acoustic information. We also find that the substitution errors are mostly originated from the English monolingual utterances. Nevertheless, the improvements are more significant than the deterioration. Thus, introducing acoustic information contained in decoding lattice to language modeling is effective in improving the code-switching ASR performance, especially for code-switching utterances. We present a speech recognition example in Table 6. Our model successfully predicts "这样" in a switching point from English to Mandarin, even though its pronunciation is similar to "讲".

We also compare several previous studies on improving code-switching ASR performance in Table 7. While [11] improves the n-gram language model and [14] integrates the sequence-to-sequence language model in the ASR, [10, 15]

[1] Kaldi provides a command "lattice-mbr-decode" that outputs a confusion network.

performs lattice rescoring by using their improved language model. The results obtained are from the ASR evaluation on dev man. The data splits for [14] and [15] are different but the distributions are similar to dev man. Our proposed method achieves the largest absolute WER reduction, although the results are considered comparable to [14] and [15]. Both [14] and [15] employ parallel monolingual text to generate some novel code-switching sentences, but such parallel monolingual text cannot be available in many languages. On the other hand, our method only employs the word confusion networks originated from the code-switching speech corpus itself. It means that we can achieve better performance by introducing acoustic and textual information from decoding lattices in language modeling, even with a limited word switching knowledge. Therefore, our proposed method is effective in helping solve the data scarcity problem.

IV. Conclusions

We propose to solve the pronunciation variations problem in word transitions by introducing acoustic and textual information extracted from ASR decoding lattices. This information is contained in lattices' word probabilities. The experiment results exhibit improvements in code-switching speech recognition. Furthermore, the ablation study shows that introducing acoustic information in code-switching language modeling is useful to improve the code-switching ASR performance. Compared with previous studies, our proposed model achieves the largest WER reduction. Taking into account acoustic and textual information in language modeling is shown to be effective despite of its word switching variations limitation. For future work, we plan to investigate the noise reduction for code-switching ASR lattices and extend our approach by incorporating bidirectional information and generated code-switching sentences.

Acknowledgment

This research is supported by JST CREST JPMJCR1687, JSPS KAKEN 16H02845.

References

[1] S. Poplack, "Sometimes I'll start a sentence in Spanish Y TERMINO EN ESPAÑOL: Toward a typology of code-switching," *Linguistics*, vol. 18, pp. 581-618, 1980.

[2] D.-C. Lyu, T. P. Tan, E. S. Chng, and H. Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia," in *Proc. of INTERSPEECH*, 2010, pp. 1986-1989.

[3] P. Guo, H. Xu, L. Xie, E. S. Chng, "Study of Semi-supervised Approaches to Improving English-Mandarin Code-Switching Speech Recognition," in *Proc. of INTERSPEECH*, 2018, pp. 1928-1932.

[4] S. Yu, S. Hu, S. Zhang, B. Xu, "Chinese-English bilingual speech recognition," in *Proc. of International Conference on Natural Language Processing and Knowledge Engineering*, 2003, pp. 603-609.

[5] J. Y. Chan, H. Cao, P. Ching, T. Lee, "Automatic recognition of Cantonese-English code-mixing speech," *International Journal of Computational Linguistics & Chinese Language Processing*, vol. 14, no. 3, pp. 281-304, 2009.

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

[7] A. Pandey, B. M. L. Srivastava, S. V. Gangashetty, "Adapting monolingual resources for code-mixed hindi-english speech recognition," in *Int. Conf. on Asian Language Processing* (*IALP*) , 2017, pp. 218– 221.

[8] E. Yilmaz, H. V. D. Heuvel, and D. V. Leeuwen, "Acoustic and textual data augmentation for improved ASR of code-switching speech," in *Proc. of INTERSPEECH*, 2018, pp. 1933-1937.

[9] N. T. Vu, D.-C Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E. S. Chng, T. Schultz, and H. Li, "A first speech recognition system for mandarin-english code-switch conversational speech,"in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (*ICASSP*), 2012, pp. 4889–4892.

[10] Z. Zeng, Y. Khassanov, V. Pham, H. Xu, E. S. Chng, H. Li, "On the end-to-end solution to mandarin-english code-switching speech recognition," in *Proc. of INTERSPEECH*, 2019, pp. 2165-2169.

[11] Z. Zeng, H. Xu, T. Y. Chong, E. S. Chng, H. Li, "Improving n-gram language modeling for code-switching speech recognition" in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (*APSIPA ASC*), 2017, pp. 1596–1601.

[12] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switching language modeling using syntax-aware multi-task learning," in *Proc. of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, 2018, pp. 62–67.

[13] E. Yilmaz, S. Cohen, X. Yue, D. V. Leeuwen, H. Li, "Multi-Graph Decoding for Code-Switching ASR," in Proc. of *INTERSPEECH*, 2019, pp. 3750-3754.

[14] C. Y. Li, N. T. Vu, "Improving Code-Switching Language Modeling with Artificially Generated Texts Using Cycle-Consistent Adversarial Networks," in *Proc. of INTERSPEECH,* 2020, pp. 1057–1061.

[15] G. Lee, X. Yue, and H. Li, "Linguistically motivated parallel data augmentation for code-switch language modeling," in *Proc. of INTERSPEECH*, 2019, pp. 3730-3734.

[16] S. Dalmia, X. Li, A. W Black, F. Metze, "Phoneme Level Language Models for Sequence Based Low Resource ASR," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (*ICASSP*)*,* 2019, pp. 6091-6095.

[17] M. Sperber, G. Neubig, N. Q. Pham, A. Waibel, "Self-Attentional Models for Lattice Inputs," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1185–1197.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, 2019.

[19] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *31st Conf. on Neural Information Processing Systems*, 2017, pp. 5998-6008.

[20] G. Lee, T. N. Ho, E. S. Chng, H. Li, "A review of the mandarin-english code-switching corpus: Seame," in *Int. Conf. on Asian Language Processing* (*IALP*), 2017, pp. 210–213.

[21] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin ASR research into industrial scale," 2018, [Online] Available: https://arxiv.org/abs/1808.10583.

[22] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *20th Int. Conf. on Speech and Computer* (*SPECOM*), 2018, pp. 198-208.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlícek, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," in *Proc. of ASRU*, 2011, pp. 1-4.

[24] V. Peddinti, D. Povey, S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. of INTERSPEECH*, 2015, pp. 3214-3218.

[25] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proc. of INTERSPEECH*, 2002.

[26] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," 2019, [Online] Available: https://arxiv.org/abs/1910.03771.

[27] H. Xu, D. Povey, L. Mangu, J. Zhu, "Minimum Bayes Risk Decoding and System Combination Based on a Recursion for Edit Distance," *Computer Speech & Language*, vol. 25, pp. 802-828, 2011.