Improving code-switching speech recognition with data augmentation and system combination

Duo Ma*, Haihua Xu†, Guanyu Li* and Eng Siong Chng‡
* Northwest Minzu University, Lanzhou, China
† Temasek Laboratories, Nanyang Technological University,singapore
‡ School of Computer Science and Engineering, Nanyang Technological University, Singapore
maduo25@163.com guanyu-li@163.com

***Abstract*—We focused on a study of comprehensive approaches to an improved code-switching speech recognition, using data augmentation and system combination methods. For data augmentation, we not only use speech speed perturbation based method, but we also attempt to add diversified room impulse response based reverberate noise, as well as music, babble, and white noise based additive noise. It is found we still can achieve significant performance improvement with such noise-corrupted data augmentation methods, though our SEAME code-switching data belongs to a clean corpus. In addition to data augmentation methods, we also adopt Minimum Bayesian risk-based lattice combination method to further improve our recognition results. We achieve significant word error rate (WER) reduction on lattice combination with/without recurrent neural network language model based lattice rescoring. Compared with our previous efforts [6], we achieve up to 2.29% and 5.61% absolute WER reduction on the two *dev* sets respectively, while 4.83% and 8.04% absolute WER reduction after system combination.**

I. Introduction

Code-switching speech recognition [21] is gradually drawing more and more attention in recent years as cross-lingual contact between people from around the world increases. It is also challenging because it contains cross-lingual transfer within or between utterances. Consequently, it is hard to learn a robust model due to data sparsity issue, compared with monolingual speech recognition case.

To alleviate data sparsity issue, data augmentation [3], [16], [8], [2], [24] is widely employed in diversified deep neural network-based machine learning areas. By data augmentation, we mean the original training data size is artificially expanded by adding some modified data versions to the training data. Taking speech recognition, for instance, one can be expanded the training data size either by perturbing the original vocal track length [7] or simply by perturbing the speed of the original speech [11]. The latter makes the data three times the original.it also shown consistent recognition performance improvement is achieved. This is especially effective for a small training data set.

In addition to data sparsity alleviation, one of another function of data augmentation is to improve the robustness of the training models. Aside from the approaches as mentioned, one can also obtain robust speech recognition with data augmentation by adding diversified noise to the original data. One of the simpler kind of noise is additive noise, such as music noise, babble noise, and white noise. Another kind of noise is reverberant noise, which is produced by convolving the original data with the room impulse response (RIR) signal.

In this paper, we are meant to improve our code-switching speech recognition performance with SEAME data [6], using the data augmentation methods as above mentioned. Besides, we also attempt to boost the results by lattice-based system combination method [22]. We are motivated by the following facts. 1) SEAME data is not a completely clean corpus. It has background babble noise. Besides, it is recorded in diversified meeting rooms. This suggests it should contain reverberant noise to some degree. As a result, it is worthwhile a study to see if the data augmentation utilizing adding noise corrupted data helps.2) It has long proved one can obtain significantly improved recognition results by combining diversified recognition systems. We employ a lattice-based system combination method hopefully to achieve further improvement.

This is natural since diversified data augmentation methods yield diversified speech recognition systems.This paper is organized as follows. In Section II,we introduce our method to improve speech recognition performance. Section III describes the technical details of our data augmentation methods, and in Section IV we describe the data employed for the experiment. In Section V, we report the experimental setup and results, and finally, we conclude the work in Section VI.

II. Proposed approaches to an improved speech recognition

Compared with our previous work [6] for code-switching speech recognition on the SEAME corpus, we make several improvements in terms of the pipeline as follows. First, we are attempting to add both reverberant and white noise to the training data, such that we evaluate the effectiveness of the noise corrupted data augmentation method on the recognition results. Secondly, we build acoustic models using state-of-the-art work in [14]. Finally, we also perform comprehensive Minimum Bayes Risk-based lattice combination methods, which yield significant performance improvement. For clarity, the proposed methods are illustrated in Figure 1.

III. data augmentations

A. Noise data description

As mentioned earlier, we employ two kinds of noise data sources. One is reverberant noise data source from

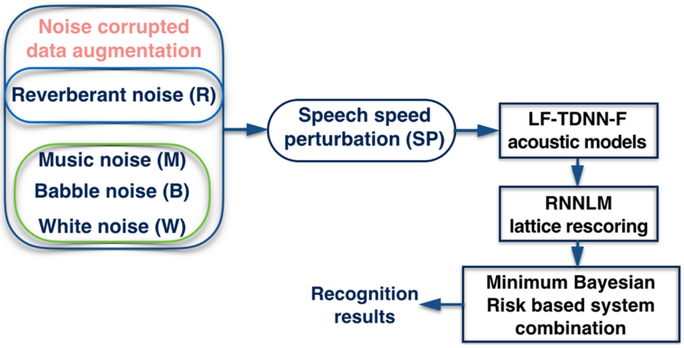
Noise corrupted data augmentation
Reverberant noise (R)
Music noise (M)
Babble noise (B)
White noise (W)
Speech speed perturbation (SP)
LF-TDNN-F acoustic models
RNNLM lattice rescoring
Minimum Bayesian Risk based system combination
Recognition results

Fig. 1: Illustration of the proposed approaches to an improved code-switching speech recognition

http://www.openslr.org/28. It is an entire bunch of Room Impulse Response (RIR) data sets, that are to be convolved with the training audio, generating reverberant noise effect. The RIRs correspond to three categories, i.e., small-room, medium-room, and large-room. Here, based on the actual SEAME data recording environment consideration, we ignore the large-room case for all the following experiments. Another noise category is additive noise, that is obtained from MUSAN data set [18], and is downloaded from *http://www.openslr.org/17*. Some details of the noise data are reported in Table I.

Noise Type	Description		
	Sub-Type	#Waves	length (hrs)
Reverberant	Samll room Medium room Large room	20K 20K 20K	-
Additive	Music Babble White	645 426 930	∼41 ∼60 ∼6

TABLE I: Noise data description

B. Data augmentation methods

As shown in Figure 1, we first perform noise corruption based data augmentation and then the speech speed perturbation based data augmentation [11]. For the latter, refer to [12] for more details. We are here to detail how we conduct noise corruption based data augmentation methods. Broadly speaking, we conducted three kinds of methods adding noise. They are outlined in Table II.

For easy notation, we denote reverberant noise as r, and $\mathrm{r_s}$, $\mathrm{r_m}$, and $\mathrm{r_{mix}}$ represent small, medium, and mixing category reverberant noise respectively. The $\mathrm{r_{mix}}$ means we randomly select small and medium room type during the process of adding noise. Similarly, we denote music, babble, and white noise respectively as m, b, and w. Tabel method A in Table II as example, $\mathrm{A_{r_s mw}}$ means we separately add reverberant noise $\mathrm{r_s}$, music noise, and white noise to the training data, such that the final data size is increased to three times of the original data (i.e., x3). As a result, if the following speech speed perturbation method that triple the data is considered, we can produce up to 12 times of the original training data after overall data augmentation operation, i.e., ∼1200 hours of training data in our case. Besides, there are many ways to combine. In practice, we just conducted a small part of the combination for method A. We do the method B in Table II similarly but merge the clean data before we do speed perturbation. This means method B plus speed perturbation can yield 15 times of the original training data. However, method C is different though clean data is also considered. For method C, whatever kind of noise data is obtained, we only randomly extract part of the data that are close to the original SEAME data in size. This is inspired by [19]. Consequently, it only produces double of the original training data.

Method	Remark
A	Without original clean SEAME data included Maximum data increase size is x4
B	With original clean SEAME data included, Maximum data increase size x5
C	With original clean SEAME data included, Maximum data increase size x2

TABLE II: Noise corruption based data augmentation methods

We use KALDI[1] script to add noise. For reverberant noise data generation, we randomly extract the RIR files from a given set to convolve with the data. For additive noise data generation, we control the signal-to-noise ratio (SNR) range 5-15dB, 13-20dB, and 0-15dB for the music, babble, and white noise respectively.

IV. Data Description

All experiments are conducted on the SEAME data [6]. It is an English-mandarin code-switching corpus, mostly uttered by young college students in Malaysia and Singapore, with a spontaneous close-talk attribute. During recording, all participants are asked to have a free conversation with an anchor, and only the speech of the participants are recorded[2]. All the training and testing data used in this work are completely the same with [6], where two `dev` sets, i.e., $\mathrm{dev_{man}}$ (about 7 hours) and $\mathrm{dev_{sge}}$ (about 4 hours), are separately defined. Both two `dev` sets contain code-switching utterances which means there are cross-lingual transfer in utterances, but the $\mathrm{dev_{sge}}$ is biased to English content, while the $\mathrm{dev_{man}}$ is biased to Mandarin content. Readers can refer to [6] for more details.

We note that all the above-mentioned data augmentation methods are only conducted on the training data part, and we fix the `dev` sets. As a result, most of our results can be compared with those in [6].

V. Experimental Setup and Results

All our experiments are conducted with KALDI. In this section, we are detailing the experimental setups and the corresponding results respectively.

[1] https://github.com/kaldi-asr/kaldi

[2] However, we can still hear the anchor's voice for some utterances in both training and testing data sets.

A. Experimental Setup

1) Acoustic modeling: The acoustic models are trained with the lattice-free maximum mutual information criterion over the factorized time-delay neural network (LF-MMI-TDNN-f) [15], [14]. The front-end is the concatenation of 40-dimensional MFCC features and 100-dimensional i-vectors[17], [9]. They are transformed before fed into the convolutional neural network (CNN). The neural networks are composed of two parts. the bottom layers are for the 6-layer CNN [20], [13], [4], which is targeted for better feature learning. Upon the CNN, they are 9-layer for the factorized time-delay neural network (TDNN-f), which shows consistently improved results and faster decoding in [14]. For the TDNN-f configuration, the TDNN layers have 1536 neurons, and the bottleneck layers have 160 neurons.

2) Lexicon and language modeling: The lexicon is mixed with ∼33K English words and ∼6K Mandarin characters. The phone set has 252 phonemes, of which 213 phonemes are Mandarin initials and finals, and the remaining 39 are English phonemes. For a better recognition, we adopt data-driven based lexicon learning to deal with the word pronunciation probability modeling and the silence probability modeling methods proposed in [1].

We use 3-gram language models boosted with maximum entropy method [5], [10] to build the grammar `G` for the first-pass decoding. We found it is yielding slightly better results compared with the conventional Kneser-Ney method.

We also employ recurrent neural network language models (RNNLMs) to rescore lattice. The RNNLM has 800-dimensional word embedding, upon which there are a TDNN and an LSTM-p layer. Particularly, the LSTM-p layer is composed of an LSTM layer with 200 cells and a projection layer with 200 neurons.

B. Results

We present results of the data augmentation, recurrent neural network language models (RNNLM) based lattice rescoring, and system combination respectively.

1) Data augmentation by noise corruption: Table III reports our baseline results. Compared with our previous results in [6], we achieve tremendous performance improvement thanks to the much more deeper and factorized neural network employed [14]. We note that the speech speed perturbation based data augmentation method [11] consistently achieves improvements on the two `dev` sets. As a result, it is always adopted in the following experiments, which means whatever training data is used, the final training data size is a 3-time duplication of the previous data size. Besides, we always use the results in the last row in Table III as the "baseline" to which the other data augmentation methods are compared.

Table IV reports word error rate results with noise corruption method A as indicated in Table II. From Table IV, we attempt various kinds of noise corruption methods, however, most of systems yield degraded results. Though method $\mathrm{A}_{\mathrm{rmixbmw}}$ has obtained marginal 0.36% WER reduction on the $\mathrm{dev}_{\mathrm{sge}}$ set, it similarly gets a worse result on the $\mathrm{dev}_{\mathrm{man}}$. One of the disadvantages of method A is that it ignores the fact that SEAME data is a clean data set. It uses noise-contaminated training data to recognize the clean `dev` sets. In the following recipes, we still employ noise-contaminated data as method A, but they are merged with the original clean data to train the speech recognition system. Due to the time limitation, we only consider a small part of combinations in this work. Table V shows the detail results. We can see from Table V that we have obtained improved recognition results in method $\mathrm{C}_{\mathrm{r}_{\mathrm{mix}}\mathrm{mwb}}$, with the clean data included to train the speech recognition system. Though the improvement is marginal but consistent on the two `dev` sets. This is crucial since adding diversified noise to the clean data (x2) means the improvement of the robustness of the system but still no performance degradation on the clean test data.

System	$\mathrm{dev}_{\mathrm{sge}}$ (%WER)	$\mathrm{dev}_{\mathrm{man}}$ (%WER)
TDNN-LF-MMI in [6]	32.42	22.57
factorized-TDNN-LF-MMI	26.55	19.62
+ speed perturbation	25.36	18.64

TABLE III: Baseline word error rate results

Method	Overall data duplication	$\mathrm{dev}_{\mathrm{sge}}$(%WER)	$\mathrm{dev}_{\mathrm{man}}$(%WER)
Baseline	x3	25.36	18.64
$\mathrm{A}_{\mathrm{r}_{\mathrm{mix}}}$	x3	26.72	19.92
$\mathrm{A}_{\mathrm{r}_{\mathrm{s}}}$	x3	26.85	19.64
$\mathrm{A}_{\mathrm{r}_{\mathrm{m}}}$	x3	26.90	19.95
A_{b}	x3	26.64	19.70
$\mathrm{A}_{\mathrm{r}_{\mathrm{mix}}\mathrm{b}}$	x6	26.76	19.90
A_{mw}	x6	25.74	18.81
$\mathrm{A}_{\mathrm{r}_{\mathrm{mix}}\mathrm{bw}}$	x9	25.83	19.11
$\mathrm{A}_{\mathrm{r}_{\mathrm{mix}}\mathrm{bm}}$	x9	25.67	18.98
$\mathrm{A}_{\mathrm{bmw}}$	x9	25.79	19.15
$\mathrm{A}_{\mathrm{r}_{\mathrm{mix}}\mathrm{bmw}}$	x12	**25.00**	**18.72**

TABLE IV: Word error rate results with overall noise corrupted training data, i.e., method A in Table II.

Method	Overall data duplication	$\mathrm{dev}_{\mathrm{sge}}$(%WER)	$\mathrm{dev}_{\mathrm{man}}$(%WER)
Baseline	x3	25.36	18.64
$\mathrm{Br}_{\mathrm{mix}}$	x6	25.31	19.08
$\mathrm{B}_{\mathrm{r}_{\mathrm{s}}}$	x6	25.65	18.98
B_{mw}	x9	25.26	18.72
C_{mw}	x6	25.41	18.32
$\mathrm{C}_{\mathrm{r}_{\mathrm{mix}}\mathrm{mwb}}$	x6	**25.27**	**18.39**

TABLE V: Word error rate results by merging the noise corrupted and the clean training data.

2) RNNLM-based lattice rescoring: As mentioned earlier, we also use the training transcripts to train recurrent neural network language models to rescore the lattice output by the different speech recognition systems. We present the RNNLM-based lattice rescoring results in Table VI. It is seen from Table VI we gain significant word error rate reductions in every situation with the RNNLM-based lattice rescoring method. Besides, we get better results where noise corruption based data augmentation method is employed. We guess this might

System	dev$_{\text{sge}}$ (%WER)	dev$_{\text{man}}$ (%WER)
[6]	29.56	20.54
Our baseline+ rescoring	24.28	17.71
$\text{B}_{\text{r}_{\text{mix}}}$	**23.9**	**17.71**
$\text{B}_{\text{r}_{\text{s}}}$	24.27	17.83
$\text{A}_{\text{r}_{\text{mix}}\text{mwb}}$	**23.87**	**17.70**
B_{mw}	**23.96**	**17.53**
C_{mw}	**23.96**	**17.24**
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$	**23.85**	**17.25**

TABLE VI: Word error rate results from RNNLM-based lattice rescoring.

System (Baseline+)	dev$_{\text{sge}}$ (%WER)	dev$_{\text{man}}$ (%WER)
Single best No system combination	23.85	17.25
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$	22.31	16.21
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$+$\text{B}_{\text{mw}}$	21.80	15.89
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$+$\text{B}_{\text{mw}}$+$\text{B}_{\text{r}_{\text{mix}}}$	21.61	15.80
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$+$\text{B}_{\text{mw}}$+$\text{B}_{\text{r}_{\text{mix}}}$+$\text{A}_{\text{r}_{\text{mix}}\text{mwb}}$	**21.52**	**15.69**

TABLE VIII: Word error rate results by system combination, with the lattice to which RNNLM-based lattice rescoring is applied.

be due to we have denser lattice for those ASR systems trained with noise augmented data.

3) Lattice-based system combination: So far, we have obtained diversified speech recognition systems due to different data augmentation methods being employed. As a result, it is interesting to examine how system combination with such diversified systems performs. Table VII reports the system combination results where no RNNLM-based lattice rescoring is applied to the lattice before or after the system combination. To show the effectiveness of the system combination method, we also present the corresponding results from the single best speech recognition system in Table VII. It can be seen the improvement from the 5 system combination (last row in Table VII) is remarkable.

System (Baseline+)	dev$_{\text{sge}}$ (%WER)	dev$_{\text{man}}$ (%WER)
Single best No system combination	25.27	18.39
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$	23.33	17.07
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$+$\text{B}_{\text{mw}}$	22.95	16.85
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$+$\text{B}_{\text{mw}}$+$\text{B}_{\text{r}_{\text{mix}}}$	22.79	16.78
$\text{C}_{\text{r}_{\text{mix}}\text{mwb}}$+$\text{B}_{\text{mw}}$+$\text{B}_{\text{r}_{\text{mix}}}$+$\text{A}_{\text{r}_{\text{mix}}\text{mwb}}$	**22.27**	**16.41**

TABLE VII: Word error rate results by system combination, where no RNNLM-base lattice rescoring is applied on the resulting lattice.

One of the advantages from KALDI is the resulting output of the RNNLM-based lattice rescoring is not N-best utterances, but it is a heavy pruning lattice [23]. This motivates us to examine the effectiveness of the system combination method on those output lattices by the RNN-based lattice rescoring method. Table VIII shows the word error rate results by system combinations over the lattices that are rescored with the RNNLMs. Here, the best single system refers to the system with RNNLM-based lattice rescoring operation in Table VIII. Results in Table VIII clearly shows the effectiveness of the system combination method over the lattice that is RNNLM rescored.

VI. Conclusions

As a continuation of our previous work, in this paper, we further made a series of efforts to achieve better code-switching speech recognition results on the SEAME data, ranged from the front-end data augmentation by adding reverberant and additive noise, to the lattice-based system combination in post-processing. We found through the SEAME data is a clean data set, adding diversified noise still can make improved speech recognition systems, not only yielding better recognition results, but also improving the robustness and diversity. As a result,we find that whether rnnlm lattice rescoring or not, the system combination method based on lattice after using our data augmentation method has a significant reduction in word error rate.This explains from the side that our data augmentation method is a substantial help to improve single speech recognition performance.

Acknowledgment

Chng Eng Siong is supported by the Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University, Singapore. Xu Haihua is supported by TL@NTU, Nanyang Technological University, Singapore. This research is also supported by Research Funds for the Central Universities (31920170145) and the Natural Science Foundation of China under Project (Grant No. 61633013) and Gansu Provincial First-class Discipline Program of Northwest Minzu University (NO.11080305).

References

[1] Guoguo Chen, Hainan Xu, Minhua Wu, Daniel Povey, and Sanjeev Khudanpur. Pronunciation and silence probability modeling for asr. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[3] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(9):1469–1477, 2015.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[5] Joshua T Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001.

[6] 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.

[7] Navdeep Jaitly and Geoffrey E Hinton. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, 2013.

[8] Naoyuki Kanda, Ryu Takeda, and Yasunari Obuchi. Elastic spectral distortion for low resource speech recognition with deep neural networks. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 309–314. IEEE, 2013.

[9] Martin Karafiát, Lukáš Burget, Pavel Matějka, Ondřej Glembek, and Jan Černockỳ. ivector-based discriminative adaptation for automatic speech recognition. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 152–157. IEEE, 2011.

[10] Sanjeev Khudanpur and Jun Wu. A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 1, pages 553–556. IEEE, 1999.

[11] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[12] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224. IEEE, 2017.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proceedings of INTERSPEECH*, 2018.

[15] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. Purely sequence-trained neural networks for asr based on lattice-free mmi. In *Interspeech*, pages 2751–2755, 2016.

[16] Anton Ragni, Katherine Mary Knill, Shakti P Rath, and Mark John Gales. Data augmentation for low resource languages. 2014.

[17] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59. IEEE, 2013.

[18] David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

[19] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[21] 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. IEEE, 2012.

[22] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu. Minimum bayes risk decoding and system combination based on a recursion for edit distance. *Computer Speech & Language*, 25(4):802–828, 2011.

[23] Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5929–5933. IEEE, 2018.

[24] Emre Yılmaz, Henk van den Heuvel, and David A van Leeuwen. Acoustic and textual data augmentation for improved asr of code-switching speech. *arXiv preprint arXiv:1807.10945*, 2018.