INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Data Augmentation for Code-switch Language Modeling by Fusing Multiple Text Generation Methods

Xinhui Hu[1]*, Qi Zhang*[1]*, Lei Yang*[1]*, Binbin Gu*[1]*, Xinkang Xu*[1]

[1]Hithink RoyalFlush AI Research Institute, Zhejiang, China

{huxinhui,zhangqi,yanglei2,gubinbin,xuxinkang}@myhexin.com

Abstract

To deal with the problem of data scarce in training language model (LM) for code-switching (CS) speech recognition, we proposed an approach to obtain augmentation texts from three different viewpoints. The first one is to enhance monolingual LM by selecting corresponding sentences for existing conversational corpora; The second one is based on replacements using syntactic constraint for a monolingual Chinese corpus, with the helps of an aligned word list obtained from a pseudo-parallel corpus, and part-of-speech (POS) of words; The third one is to use text generation based on a pointer-generator network with copy mechanism, using a real CS text data for training. All sentences from these approaches show improvement for CS LMs, and they are finally fused into an LM for CS ASR tasks.

Evaluations on LMs built by the above augmented data were conducted on two Mandarin-English CS speech sets D-TANG, and SEAME. The perplexities were greatly reduced with all kinds of augmented texts, and speech recognition performances were steadily improved. The mixed word error rate (MER) of DTANG and SEAME evaluation dataset got relative reduction by 9.10% and 29.73%, respectively.

Index Terms: code-switching, synthetic sentence, language model, speech recognition, pointer-generator network.

1. Introduction

With the rise of globalization, CS speech, which is defined as more than one language are contained inside an utterance, is becoming a common phenomenon in current societies, in particular in multilingual communities. The basic characteristic of CS speech is that speakers often embed words and phrases of a non-native language into the syntax of a native language in their conversations. Because of this characteristic, CS speech recognition is much more difficult than monolingual speech recognition. One of the reasons for that is the problem of training data scarce for both the acoustic model (AM) and the language model. In this study, we focus on the aspect of LM.

Here, we define the native language as the matrix language, while the non-native language as the embedded language. Different from in a monolingual language, there are following additional problems to be solved in CS LMs: (1) predicting words from both languages, (2) knowing when to pick words from each language, and (3) knowing when to code-switch of languages. The synthetic data have been verified useful for improving CS language modeling from the viewpoints of theory [1]. There are already theory researches leading the way for CS language modeling in automatic speech recognition (ASR) and natural language processing (NLP) systems, such as Equivalence Constraint (EC) theory[2][3][4], Functional Head Constraint (FHC) theory [5][6], and Matrix Language Frame (MLF) theory [7][8]. The synthetic CS sentences obtained by using these theories were all found indeed to be able to improve the perplexity of the trained LM over baseline models which were built by using only monolingual and real CS texts, the word error rate (WER) of related ASR systems were also reduced considerably.

Recent years, the neural network (NN) technologies demonstrate their strengths in the field of text generation, and have been actively applied to the domains of CS text processing [9][10][11][12]. For example, in [12], a seq-to-seq pointer network model with a copy mechanism was proposed to generate CS text by leveraging parallel monolingual translations from a limited source of CS sentences. The model learns how to combine words from parallel sentences and identifies when to switch one language to the other. Moreover, it captures CS constraints by attending and aligning the word inputs, without requiring any external knowledge. Based on their experiments, the LM trained by the generated sentences achieved the state-of-the-art performances.

Although these studies examined the generation methods of CS synthetic text from different angles and achieved good findings for language modeling, they mainly focused on a particular perspective to investigate the effect of an approach, either linguistic constraints or NN-based generation, few of them looked at the combination of them. As the results, the robustness of the LM built by them is poor, and the recognition performance of the ASR systems often gets worse even the PPLs of their LMs are decreased. On the other hand, these methods have strong dependencies on data and the performance variations of such LMs are large with different data sets.

In this study, we explored CS data augmentations from several perspectives, including monolingual sentences, syntactic constraints-based and NN-based sentence generation. The contributions of this paper are as follows: (1) emphasizing monolingual sentences to broaden the coverage of each language; (2) combining syntactic constraints-based approach method to enhance CS sentences in generality. (3) fusing NN-based generation method to enhance CS sentences in particularity.

2. System description

2.1. LM and synthetic sentences

The configuration of our CS LM system, together with data sources used for it, is shown in Figure 1. In this system, there are a baseline LM [13]($baselm$ [1]), 4 sub-LMs respectively trained by data sets D_{trns} (a transcript of a CS speech training data), D_{mono} (a monolingual sentence set), D_{POS} (a synthetic sentence set obtained from a syntactic constraints-based generation), and D_{pntn} (a synthetic sentence set obtained from a pointer-generator network-based generation).

[1]It was provided by the ASRU2019 Code-switching speech recognition challenge, http://asru2019.org/wp/?page_id=1881.

http://dx.doi.org/10.21437/Interspeech.2020-2219

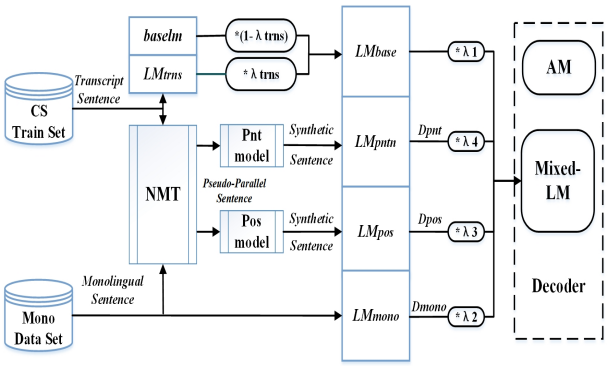

Figure 1: *Language model configuration for CS ASR system. Here, the AM refers to acoustic model.*

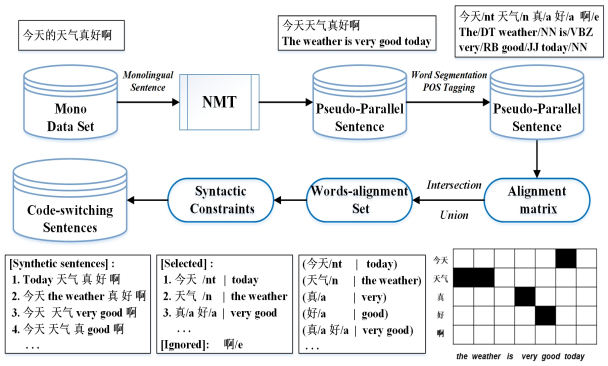

Figure 2: *Work flow of generation of D_{POS} based on syntactic constraints (POS).*

2.2. LM construction

In this study, n-gram LMs (n=3) were used for ASR tasks, and they were constructed by using the SRILM Toolkit[14]. With the $baselm$ and the augmented texts, the LM for decoding was built with Equation (1).

$$LM = \lambda_1 LM_{base} + \lambda_2 LM_{mono} + \lambda_3 LM_{POS} + \lambda_4 LM_{pntn} \quad (1)$$

where $LM_{base} = (1 - \lambda_{trns})baselm + \lambda_{trns} LM_{trns}$, and $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1.0$. Here, the LM_{trns} is the LM trained by a transcript of CS speech. The LM_{mono} is the LM trained by a selected monolingual sentence set, LM_{POS} is the LM trained by using synthetic sentences from a syntactic constraints-based generation, and LM_{pntn} is the LM trained by the synthetic sentences from a pointer-generator network-based generation. The weight values λ_* were obtained by tuning perplexity of the LM with respect to a development set.

3. Sentence generation procedures

3.1. D_{mono} - selections of monolingual sentence

Compared with CS text, monolingual texts are abundantly available. Due to this reason, how to utilize monolingual data in C-S modelings attract a lot of interests [15][16][17]. In [17], a variety of training schemes were explored and verified the effectiveness of training with large amounts of monolingual data followed by fine-tuning with small amounts of CS data. These facts revealed that monolingual data plays an important role in the CS language modelings. From the global points of view, monolingual data have abilities to improve predicting words within a language so that it can enhance the coverage and robustness of LM for that individual language.

Since CS phenomenon mainly happens in daily conversations, we accordingly selected monolingual sentences D_{mono} from an existing text corpus in which domain was daily conversation. In this study, 20G of monolingual Chinese texts (148.5M sentences) and 8G English texts (99M sentences) were selected.

3.2. D_{POS} - generation with syntactic constraints

The sentence D_{POS} set was generated by the workflow shown in the Figure 2. The inputs of the whole procedure was a Chinese monolingual sentence set in the domain of daily conversation. With the help of a neural machine translation (NMT) system[18], the input sentence set was converted into a Chinese-English pseudo-parallel sentence set. Then, taking the pseudo-parallel sentence set as a target, CS synthetic sentences were obtained by performing the following steps.

- Word segmentation and POS tagging for both Chinese and English sentences using methods in [19] and [20];
- Using the EM algorithm to find alignment nodes, adding bidirectional alignment nodes to the alignment matrix;
- Using the operations of intersection and union, getting all word alignment list (word-pair);
- Based on syntactic constraints as did in [21], selecting tags of verb, noun, pronoun, adjective, and adverb as target POS set;
- For a tagged Chinese sentence, some words that belongs to the above target POS set were randomly substituted with their aligned English counterparts, forming a fixed number of CS sentences.
- Repeating the above processing for all sentences in the input corpus.

In this procedure, the synthetic sentences were obtained by substituting words or phrases in sentences of the matrix language (Chinese) with their aligned counterparts of the embedded language (English). Since the substitution was among the aligned words, it did not alter the syntactic and semantic integrity of the sentence. We chose phrases and words with different POS tags evenly and limited the proportion of replacements within 20% of a whole sentence.

In this study, we used a mono Chinese conversational sentence set containing 48.3M sentences (3.9G in size) as the source data, and finally obtained an CS sentence set of 155.5M sentences (20G in size) by using the above procedures.

3.3. D_{pnt} - generation using pointer-generator network

3.3.1. Characteristics of pointer-generator network

In order to fix the problem of unnatural sentences appeared in the syntactic constraints-based method, the pointer-generator network[22] based generation was supposed to be a reasonable solution. The pointer-generator network is a variation of the sequence-to-sequence model with attention. Instead of translating one sequence into another, it yields a succession of pointers to the elements of the input series. It uses attention as a pointer to select a member of the input sequence as the output, that is to copy words from the input to the output using an attention mechanism and generate the output sequence using its decoder.

Table 1: *Statistics of CS data $DTANG$ and $SEAME$.*

Set	Item	Train	Dev	Test
DTANG	Utterance	183,479	16,123	16,152
	$Tokens_E$	301,477	22,402	22,133
	$Tokens_C$	1,571,972	180,939	181,745
	Dur.(H)	200.0	20.6	20.4
SEAME	Utterance	74,927	9,301	9,552
	$Tokens_E$	316,726	30,154	50,537
	$Tokens_C$	661,025	101,076	64,009
	Dur.(H)	97.0	8.0	6.0

In the decoding step, word generation probability Pgen [0,1] is calculated by summing weights on word distributions and attention distributions. It can either copy words through pointers or generate words from a predefined vocabulary.

In text generation, the pointer-generator network has been verified superior to other methods, including the seq2seq model with attention, linguistic constraint-based methods [12][21]. Different from the D_{POS}, the sentences D_{pnt} generated by the pointer-generator network have high naturalness and similar expression styles as the input sentences. So, such kinds of generated sentences are regarded as suitable to be used as in-domain data, and generally applied as an extension of real data.

3.3.2. Training pointer-generator network

The pointer-generator network was trained by a concatenated sequence of parallel sentences, constrained by code-switching texts. The attention mechanism helped the decoder to generate meaningful and grammatical sentences.

To obtain training data in the form of (parallel sentence, CS sentence) for training the pointer-generator network, we selected a Mandarin-English CS text set DTANG (8.5M, including training and dev sets) as the source data. With the help of a neural machine translation (NMT) system [18], the source data set was converted into a Chinese-English pseudo-parallel sentence set. In this study, we selected 3-best of the decoding results as output, and finally obtained 26M (in size) generated sentences.

4. Experiments

4.1. Corpora

One CS speech corpus, referred as DTANG, was provided by the ASRU 2019 Code-Switching Challenge [13][23], its contents were shown in Table 1. The AM for experiments was trained by using its training speech data sets that included 200 hours of CS speech, 500 hours of monolingual Mandarin speech, and 200 hours of monolingual English speech. To verify the effect and robustness of our proposed method, we also used another popular CS corpus SEAME for evaluations[24][25][2]. However, the SEAME train set was not utilized except for particular notes.

Besides CS corpora, the other text sets were also used as shown in Table 2. Three types of sentences were selected from our existing monolingual text corpora including English and Chinese, and DTANG's Chinese-English CS texts originated from its speech transcripts. As for the data source, these data can be divided into speech transcripts and self-collected texts.

[2]The SEAME test set was obtained from https://github.com/zengzp0912/SEAME-dev-set.

Table 2: *Details of different data resources.*

Type	Chinese	English	CS
Speech trns.	20M	49M	7.7M
General text	20G	8G	

The domains of these data were all in daily conversations. The data briefs are shown in Table 2.

4.1.1. CS Data statistics and analyses

Figure 3 shows the Chinese character's occurrence distributions in the CS data sets of DTANG and SEAME. Although the Chinese takes the majority of a sentence in both sets, its concentration degrees in two sets are clearly different. The curve of the SEAME is relatively flat, that means no big difference among all percentage interval. On the other hand, the curve of the DTANG is obviously concentrated in the left half, where the occupied percent is larger than 50%, of which, the largest one is the interval containing over 80% Chinese characters.

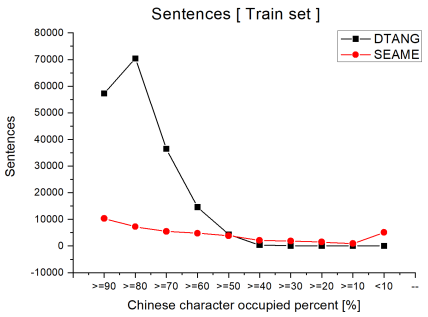

Figure 3: *Distributions of Chinese character in two CS sets.*

Such phenomenon can be attributed to the fact that they were collected from different places, the DTANG was built in mainland China, and the SEAME was built in Singapore. This reveals a fact that a place with stronger multilingual atmosphere prefers language mixtures. In DTANG, word embeddings are preferred, while in SEAME relatively long phrases are often embedded.

4.2. Performance measures

Two performance measures, perplexity (PPL) and mixed error rate (MER) were used for LM evaluations. During the calculation of MER, each single Chinese character was regarded as an unit, while an English word was regarded as an unit.

4.3. Other NN-based generation methods for comparisons

We also compared the pointer-generator network with the other two NN-based generation methods including transformer-based [18] and BERT-based [26] models. These CS text generation methods were realized by applying the neural network as a masked language model to predict the mixed-in embedded words [11], taking the real CS texts as training data.

4.4. Experimental results

Table 3 shows experimental results with respect to different LMs. These LMs were trained by using different synthetic sen-

Table 3: *Performance (PPL and MER) changes with training data*

ID	Model (+training data)	+Size	DTANG Dev.		DTANG Eval		SEAME Eval	
			PPL	MER	PPL	MER	PPL	MER
B_0	$baselm$		1198.00	6.58	947.85	6.61	2634.68	67.69
B_1	$B_0 + TRN_{DTANG}$	7.7M	770.56	6.60	780.09	7.23	2148.71	67.90
B_{1m}	$B_1 + mono$	28G	800.27	6.50	834.57	7.15	1085.29	64.63
B_{1p}	$B_1 + POS$	12G	641.32	6.17	680.08	6.99	1115.10	65.13
B_{1B}	$B_1 + BERT$	400M	250.98	5.76	620.39	6.83	853.87	61.03
B_{1T}	$B_1 + Transformer$	400M	210.86	6.02	668.29	6.88	834.19	60.98
B_{1N}	$B_1 + PointNet$	26M	295.08	5.32	635.69	6.80	716.01	58.32
BMP	$B_{1p} + mono$		536.53	5.84	606.82	6.84	424.21	57.80
BPN	$B_{1p} + PointNet$		119.35	2.96	550.51	6.51	528.87	58.02
BMP_B	$BMP + BERT$		124.57	3.83	477.25	6.12	406.24	57.36
BMP_T	$BMP + Transformer$		122.78	3.82	490.48	6.09	404.93	57.11
BMP$_N$	***BMP +PointNet***		***109.32***	***2.71***	***467.00***	***6.01***	***389.12***	***57.52***
$BMPN_S$	$BMPN + TRN_{SEMAE}$	4.2M					152.76	47.56
BMPN$_{SA}$	***BMPN$_S$, AM$_{SEAME}$***							***23.74***

tences, together with a baseline LM. From the results, we can see that the LM performances were smoothly improved with additions of these synthetics sentences, such as the selected monolingual sentences ($mono$), synthetic sentences by syntactic constraints based substitution POS and synthetic sentences by the NN-based text generators.

BMP_N gets the best performance. In this case, sentence sets (TRN_{DTANG}, mono, POS, PointNet) were used. Compared with baseline model B_0, the PPL of the test set with respect to BMP_N was dropped from 947.85 to 467.00 (with 50.7% relatively reduction), and the MER was reduced from 6.61% to 6.01% (with 9.1% relative reduction).

5. Discussions

Through our experiments and analyses, we have some clear findings on following points.

Fusions of different sentence types: for CS LMs, such fusions were demonstrated superior to individual sentence set;

Monolingual sentences (mono): compared B_{1m} with B_1, and BMP with B_{1p}, the MER steadily decreased with the addition of monolingual sentences ($mono$) although PPL was increased in some cases. It can be explained that the monolingual sentences were selected from a conversational corpus in which domain was consistent with the CS speech, the prediction ability for each individual language was improved.

Syntactic constraint-based generation (POS): compared B_{1p} with B_1, and BMP with B_{1m}, both PPL and MER were decreased with a large scale when sentence set POS was added. This addition was assumed to broaden the coverage of the CS sentences. Moreover, the improvement by it was found larger than by monolingual data $mono$;

NN-based generation methods (BERT, Transformer, PointNet): it is noted that all the 3 NN-based sentence generation methods have shown to be effective for CS LMs. Of the 3 NN-based generation methods, the PointNet was the most effective. It can be assumed that generated sentences with its copy mechanism have a strong similarity to the in-domain data.

Data dependency: The performances on data set $SEAME$ were also improved with the above augmentation data, although it was not used for training. So, the problem of data dependency for CS LM was alleviated.

Achievements in the ASRU 2019 CS speech recognition challenge: in the track 2 of the challenge in which the participant's LM was the object of evaluation, our system was ranked the fourth place [13]. However, this was the result based on the whole system including AM and LM. By our analyses on the effect of our LM, the MER of our system was reduced from 6.61% (in the track1 where an assigned LM was used) to 5.88% (in the track2 where the same AM as the one in the track1 was used), and 11.04% of relative reduction. This reduction was the largest one among all the participants, and was clearly ahead of the team ranked in the 2nd best reduction in MER (with 4.4% relative reduction).

6. Conclusions

In this study, we proposed a data augmentation approach based on multiple sentence generation methods for the CS language models. These generation methods were considered from different aspects, such as the robustness of monolingual language, the generality of CS sentence by syntactic constraints, and the particularity of an in-domain CS sentence. The fusion of these methods showed that they have strong complementarities.

We conducted evaluations of the LM in the contexts of C-S ASR experiments. Compared with a baseline LM, for the test set of DTANG, the PPL was reduced by relative 50.7%, the MER was decreased relatively 9.1% when our augmented sentences were used. For the test set of SEAME, which data were even not used for training, the PPL and MER were relatively reduced by 85.2% and 15.0%, respectively. Moreover, when the SEAME's training transcripts were used for LM, the relative reductions were 94.2% and 29.7%, respectively.

7. Acknowledgements

We would like to thank Dr. Zhiping Zeng for providing us the SEAME test set. We also thank our colleagues of acoustic team to prepare acoustic models for experiments.

8. References

[1] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. 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)*, 2018, pp. 1543–1553.

[2] S. Poplack, "Sometimes i' ll start a sentence in spanish y termino en espanol: toward a typology of code-switching1," *Linguistics*, vol. 18, no. 7-8, pp. 581–618, 1980.

[3] D. Sankoff, "A formal production-based explanation of the facts of code-switching," in *Bilingualism: Language and Cognition*, 1998, pp. 39–50.

[4] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. 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)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1543–1553. [Online]. Available: https://www.aclweb.org/anthology/P18-1143

[5] H. M. Belazi, E. J. Rubin, and A. J. Toribio, "Code switching and x-bar theory: The functional head constraint," *Linguistic inquiry*, pp. 221–237, 1994.

[6] Y. Li and P. Fung, "Language modeling with functional head constraint for code switching speech recognition," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 907–916.

[7] D. Languages, "grammatical structure in codeswitching," 1993.

[8] G. Lee, X. Yue, and H. Li, "Linguistically motivated parallel data augmentation for code-switch language modeling," in *Twentieth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 3730–3734.

[9] S. Garg, T. Parekh, and P. Jyothi, "Code-switched language models using dual rnns and same-source pretraining," *arXiv preprint arXiv:1809.01962*, 2018.

[10] B. Samanta, S. Reddy, H. Jagirdar, N. Ganguly, and S. Chakrabarti, "A deep generative model for code-switched text," *arXiv preprint arXiv:1906.08972*, 2019.

[11] Y. Gao, J. Feng, Y. Liu, L. Hou, X. Pan, and Y. Ma, "Code-switching sentence generation by bert and generative adversarial networks," *Proc. Interspeech 2019*, pp. 3525–3529, 2019.

[12] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switched language models using neural based synthetic data from parallel sentences," *arXiv preprint arXiv:1909.08582*, 2019.

[13] X. Shi, Q. Feng, and L. Xie, "The asru 2019 mandarin-english code-switching speech recognition challenge: Open datasets, tracks, methods and results," *arXiv preprint arXiv:2007.05916*, 2020.

[14] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "Srilm at sixteen: Update and outlook," in *Proceedings of IEEE automatic speech recognition and understanding workshop*, vol. 5, 2011.

[15] S.-P. Chuang, T.-W. Sung, and H.-Y. Lee, "Training a code-switching language model with monolingual data," *arXiv preprint arXiv:1911.06003*, 2019.

[16] S.-P. Chuang, T.-W. Sung, and H.-y. Lee, "Training code-switching language model with monolingual data," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7949–7953.

[17] H. Gonen and Y. Goldberg, "Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training," *arXiv preprint arXiv:1810.11895*, 2018.

[18] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2tensor for neural machine translation," *CoRR*, vol. abs/1803.07416, 2018. [Online]. Available: http://arxiv.org/abs/1803.07416

[19] Z. Li, M. Zhang, W. Che, T. Liu, W. Chen, and H. Li, "Joint models for chinese pos tagging and dependency parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1180–1191.

[20] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*. Association for Computational Linguistics, 2003, pp. 173–180.

[21] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switching language modeling using syntax-aware multi-task learning," *arXiv preprint arXiv:1805.12070*, 2018.

[22] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[23] "Asru 2019 code-switching challenge: Data introduction," http://asru2019.org/wp?page_id=1881/, accessed Feburary 26, 2020.

[24] 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 *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[25] P. Guo, H. Xu, L. Xie, and E. S. Chng, "Study of semi-supervised approaches to improving english-mandarin code-switching speech recognition," *arXiv preprint arXiv:1806.06200*, 2018.

[26] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: The impact of student initialization on knowledge distillation," *arXiv preprint arXiv:1908.08962*, 2019.