INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Linguistically Motivated Parallel Data Augmentation for Code-switch Language Modeling

Grandee Lee [1], *Xianghu Yue* [1], *Haizhou Li* [1,2]

[1]National University of Singapore, Singapore
[2]University of Bremen, Germany
{grandee.lee, xianghu.yue}@u.nus.edu, haizhou.li@nus.edu.sg

Abstract

Code-switch language modeling is challenging due to data scarcity as well as expanded vocabulary that involves two languages. We present a novel computational method to generate synthetic code-switch data using the Matrix Language Frame theory to alleviate the issue of data scarcity. The proposed method makes use of augmented parallel data to supplement the real code-switch data. We use the synthetic data to pre-train the language model. We show that the pre-trained language model can match the performance of vanilla models when it is fine-tuned with 2.5 times less real code-switch data. We also show that the perplexity of a RNN based language model pre-trained on synthetic code-switch data and fine-tuned with real code-switch data is significantly lower than that of the model trained on real code-switch data alone and the reduction in perplexity translates into 1.45% absolute reduction in WER in a speech recognition experiment.

Index Terms: code-switch, language modeling, synthetic data generation

1. Introduction

Code-switching is commonly practiced by multilingual speakers and it is defined by the mixing of two or more languages within a sentence or between sentences [1]. In multilingual societies like Singapore and Hong Kong, such linguistic phenomenon pervades in both spoken and written communication. Code-switch poses many challenges to human language technologies, especially in the area of language modeling [2]. For instance, a code-switch language model will not match its monolingual counterparts due to data scarcity as well as expanded vocabulary that involves two languages. Since many downstream tasks such as automatic speech recognition (ASR) and machine translation depend on a language model, code-switch language modeling is a challenging imperative.

Code-switch occurs within a sentence sparingly and it occurs according to the speaker's preference [3]. In order to learn the sparse code-switch pattern, we argue that much more code-switch data are required for code-switch language modeling than monolingual ones. However, in reality, there is far less data available to us. This is because code-switch primarily exists in spoken form, and it would be very difficult to document a large amount of it enough for code-switch language modeling.

Existing solutions leverage on using linguistic information to generalize word lexicon. Many incorporate class [4, 5, 6, 7], Part-of-Speech [8, 9] or language ID [5] together with word input to improve generalization of the language model to the unseen test sequence. In [10, 11, 12] code-switch permission constraints are used to provide a code-switch probability for the language model. All the above-mentioned methods have contributed to different aspects. However, as the training data is under-resourced, we would expect further improvement when more training data is available. In this paper, we wish to generate synthetic data to tackle data scarcity problem directly.

A related work [13] proposes the use of Equivalence Constraint theory [14, 15] to generate synthetic code-switch data for English and Spanish. In [16, 17], word embedding is used to synthesize bigrams. However, the linguistic study regarding the syntactic structure of code-switch is still an active research area and there is no consensus on the theory underlying the nature of code-switching discourse. Among others, there are three dominant theories in explaining the formation of code-switch, they are the Matrix Language Frame (MLF) theory [18], the Equivalence Constraint theory [14, 15] and the Functional Head Constraint theory [19, 20]. We adopt MLF which assumes that a code-switch sentence will have a dominant language (matrix language) and inserted language (embedded language). The insertions could be words or larger constituents and they will conform to the grammatical frame of the matrix language. We choose MLF because, to our knowledge, no work has been conducted on using MLF to generate synthetic data for code-switch language modeling. We study English and Chinese code-switch in Southeast Asian countries, that is a more distant language pair than other common pairs such as English and Spanish. We think that MLF describes the language well in reality.

This work is also motivated by language model fine-tuning [21], where a pre-trained language model is later fine-tuned for downstream tasks much like the case of pre-training on ImageNet in vision. Although we do not follow strictly the proposed method such as using the slanted triangular learning rates and adding a new task-specific layer, we are motivated by the idea of proposing a good initial prior so that subsequent task can be improved or achieve comparable performance with much less data. Such pre-training and fine-tuning technique also have the additional advantage of faster convergence.

2. Synthetic Data Generation

The insertional assumption in MLF strongly motivates the use of aligned parallel data. Given a pair of aligned sentences as shown in Fig. 1, we can randomly select a few words in Chinese and substitute them with their aligned counterparts to synthesize a Chinese matrix and English embedded code-switch sentence, and vice versa for English.

In this naive approach, we have assumed that insertions happen at word level and also they occur everywhere without regard to syntactic boundaries which deviates from the MLF theory. Phrases such as "this is" which would normally switch together will be broken up and switched separately. Intuitively the next step is to align the phrases rather than the words in a sentence. In this way, we can impose some constraints on the code-switch points using phrase based alignment.

http://dx.doi.org/10.21437/Interspeech.2019-1382

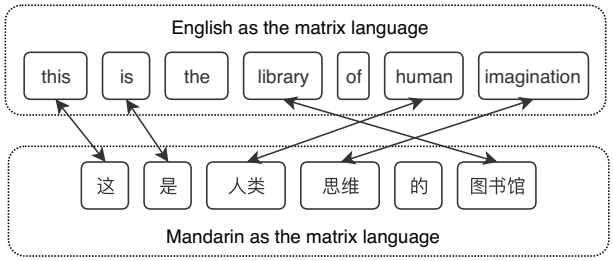

Figure 1: *An example of aligned parallel sentences. In the naive approach, the aligned words are randomly inserted into each of the matrix language.*

2.1. Phrase Based Alignment

The phrase based alignment reported in this paper is different in concept to the one in machine translation, whereby the later is optimized for the translation probability of source to target sentences. In this work, we are concerned with grouping words which are more likely to be together in our vernacular. At the same time, in the phrase extraction phase, we want to extract verbal phrases which do not violate the alignment. In the example in Fig. 1, "human imagination" can be regarded as a possible phrase whereas "library of human" should not be a phrase since the alignment is violated in the Chinese matrix language. Thus, we should only explore around the alignment points for the possible phrases. The phrase alignment and phrase extraction can be achieved using a statistical machine translation system, such as Moses [22]. The resultant phrase table also contains phrase entries that are long and infrequent, which should not be considered as verbal phrases.

It is observed that the length of embedded language is usually short, therefore grouping of "library of human imagination" altogether should be discouraged. Also, we find that a simple frequency threshold cannot capture most of the verbal phrases based on our human evaluation, whereby a low frequency threshold tend to include longer phrases which contain valid code-switch points, thus should be broken into smaller phrases, and a high threshold will exclude certain rare valid verbal phrases. A good balance is to use f_r together with low absolute frequency threshold to filter out extreme cases. f_r of the phrase w is defined as,

$$f_r(w) = \frac{Count(w)}{\sum_{i \in P} Count(w_i)}, \tag{1}$$

where P is the set of phrases that contain w at the beginning. Therefore, phrase length l and relative frequency f_r are used as criteria to extract the verbal phrases from the phrase table. The extracted verbal phrases combined with word alignment are used to generate synthetic code-switch data.

2.2. Sampling Based on Code-switch Probability

Using the phrase based alignment in the previous section, we could exhaustively generate all the possible combination of synthetic code-switch text, in which case the final generated corpus will be large and the original unigram distribution will be skewed. Certain word will have its original distribution multiplied more than the rest due to an exponential increase of code-switch combinations with increasing sentence length and alignment points. Skewing of unigram distribution introduce bias in the data and ultimately will affect the language model that is trained on the skewed data.

Thus, we use a sampling approach whereby each aligned word or phrase pair is assigned with a code-switch probability p_{cs}, and the synthetic sentences are generated on-the-fly during the language model training process. In each iteration of training, sentences are generated following the syntactic structure of the Chinese text and code-switch English words or phrases according to p_{cs} and vice versa. Given that there are enough iterations, the model is able to observe all code-switch combinations including the original monolingual sentences. Furthermore, unigram count of words or phrases will increase proportionally to the number of iterations.

2.3. Adaptation

As [23] points out, any language model is very brittle to out of domain evaluation. Not only are we not able to obtain large real code-switch data, but we also cannot obtain large parallel data in the target domain. Therefore we implement an adaptation phase to alleviate the effect of domain mismatch of the pre-trained model to test data, and present the model with in-domain monolingual data in the last training step. In this way, the improvement in perplexity after it is fine-tuned on code-switch data will be largely due to observing real code-switch data and not due to observing in-domain data. The data preparation will be described in the next section. The training strategy is to pre-train the model on the synthetic mixed domain data, and adapt it to the target domain in the final adaptation phase.

3. Experiments

3.1. Corpus

SEAME (South East Asia Mandarin-English) corpus is a well-documented database for spontaneous conversational speech [24], recorded under the setting of casual conversation and interview. SEAME corpus comes with text transcription. For parallel corpora, we use OpenSubtitle [25] and TedTalk [26] because they are from spontaneous speech generally, well reported with quality sentence alignment and most importantly available in abundance. We use GIZA++ for the alignment [27].

The SEAME corpus is broken down into collections of monolingual and code-switch sentences, and code-switch sentences are further divided into Train, Dev, Test of approximately equal proportions, as shown in Table 1. The monolingual sentences of SEAME are used for the adaptation phase described in Section 2.3. In the pre-processing step, hesitation, paralinguistic markers and punctuations are removed and the Chinese text is segmented using the Standford Chinese segmenter [28]. The parallel corpora are referred to as *Parallel* in Table 1 and they are used to generate synthetic code-switch corpus, referred to as *Synthetic*, using the method described in Section 2. SEAME *Train*, subsequently referred to as *Train*, is the only source of real code-switch data and it is used to fine-tune the pre-trained model. The composition of the database is summarized below.

Vocabulary coverage is the percentage of entire SEAME corpus' vocabulary found in the parallel corpora, it is 100% for Chinese and 92% for English. The Switch Point Fraction (SPF) is the ratio of code-switch points to the number of word boundaries within a sentence. The average SPFs are reported for different SEAME subsets. A standard vocabulary of $69K$, including the vocabulary from the parallel corpora and *Train*, is used in all subsequent language models presented in Section 3 and 4.

Table 1: *Summary of datasets used. SEAME Phase II corpus is first divided into monolingual sentences and code-switch sentences. The code-switch sentences are further divided into Train, Dev and Test.*

Dataset	#Token	#Vocab	SPF
Parallel [26, 25]	$4.80M$	$68K$	0
SEAME *Monolingual*	$0.35M$	$12K$	0
SEAME *Train*	$0.29M$	$12K$	0.23
SEAME *Dev*	$0.29M$	$12K$	0.23
SEAME *Test*	$0.28M$	$12K$	0.23

3.2. Model

A standard LSTM-based language model [29, 30] is used for all experiments, because the main goal of this paper is to tackle data scarcity and we believe that further improvement in model architecture will also improve on top of this method. The language model is of 2 LSTM layers with 300 hidden unit and a drop-out rate of 0.3 in-between layers [31]. The word w_t is the input to the LSTM model.

$$y_{t+1} = LSTM(w_t) \tag{2}$$

$$p_i = \frac{e^{y_i}}{\sum_{j=1}^{V} e^{y_j}}. \tag{3}$$

$$Loss = -\frac{1}{N} \sum_{i=t+1}^{N} Y_i \ln(p_i), \tag{4}$$

The output y_{t+1} of LSTM model is normalized by softmax over the total vocabulary V, to obtain the predicted word's distribution p_i. Finally the loss function is cross-entropy and Y_i is the one-hot label for the correct prediction.

4. Results

All the perplexities are reported on *Test* set without exception. In the following experiments, we categorically compute two sets of perplexities for each test scenario, one for model pre-trained with synthetic code-switch data and another case for the model trained from scratch with *Train*. One thing to note is that the pre-trained model is already adapted with SEAME *monolingual* data to close the domain gap as outlined in the adaptation phase in Section 2.3. The baseline model is the one trained from scratch with SEAME *Monolingual* data first and then *Train* to ensure the only difference between the models is synthetic data pre-training.

Table 2: *Perplexity of the model under the various training scenarios. The synthetic corpus used for pre-training is phrase aligned with switch probability* $p_{cs} = 0.7$.

Model	Pre-training	Training	Perplexity
Baseline	No	SEAME *Train*	219
*Pre*CS1	*Synthetic*	No	359
*Pre*CS2	*Synthetic*	SEAME *Train*	**173**
*No*CS	*Parallel*	SEAME *Train*	223

The perplexity reduction of the model *Pre*CS2, which is pre-trained on the *Synthetic* code-switch corpus and fine-tuned on *Train*, over the Baseline model in Table 2 is 21%. This significant improvement in perplexity is a positive indication of the effectiveness of the proposed synthetic code-switch pre-training framework. Additionally, we tested a model pre-trained with the original parallel corpora, *No*CS, adapted with SEAME *Monolingual* and then fine-tuned with *Train*, which only differs from the proposed fine-tuned model by the data augmentation process. Its perplexity is 223, that is a bit worse than the baseline. This shows that data augmentation is necessary and without it, the mixed-domain data will hurt the target domain model. Furthermore, the pre-trained model without fine-tuning, i.e. *Pre*CS1, can still give a perplexity of 359, indicating it to be a good prior.

The pre-trained model also offers the advantage of faster convergence. Under the same learning rate condition, a model trained from scratch will require more than 20 epochs to converge while fine-tuning the pre-trained model will require no more than 7 epochs.

4.1. Effect on the Code-switching Word Sequence

To show analytically that the improvements come from code-switch segments rather than from the monolingual segments of the sentences, we tabulate the average perplexity of the words immediately after switch points, since such words are affected most by data scarcity due to code-switch and we want to improve their prediction. At the same time, we do not want the monolingual segment to suffer from pre-training with synthetic code-switch data, so we also show the perplexity of words after non-switch points. The results in Table 3 affirm our claim.

Table 3: *Average perplexity of words immediately after a switch or non-switch point in SEAME Test.*

Baseline		Pre-train with fine-tune	
Non-switch	Switch	Non-switch	Switch
183	332	160	207

The synthetic data pre-training also improves the monolingual segments, it is not surprising since the monolingual segments exist in a code-switch context. The LSTM model will predict the non-switching word based on the past context which may contain code-switch words, thus better modeling of the code-switch segments will also improve the monolingual segments.

4.2. Effect of Fine-tuning Data Ratio

We fine-tune the model with different proportions of *Train*, shown in Table 4. This simulates the practical situation in which we could obtain a limited amount of in-domain data for training and much larger parallel data for pre-training.

Table 4: *The perplexity of the pre-trained model fine-tuned with different percentage (*λ*) of real code-switch data.*

λ	20%	40%	60%	80%	100%
Pre-training+λ*Train*	238	213	196	182	173

We observe that when training with 40% of the original SEAME *Train* we can already achieve comparable perplexity with a model trained from scratch on the full *Train* set. And as we include more *Train* data, we are able to reduce the perplexity to 173 which is a 21% improvement over the case in Baseline in Table 2. This result supports our claim that generated synthetic data contribute significantly to code-switch language modeling. Also, synthetic data is generated using a small parallel set, and results are expected to improve with an increasing amount of synthetic data.

4.3. Effect of Different Switch Probability

Table 5 explains our choice of $p_{cs} = 0.7$. Without the adaptation phase, $SPF = 0.30$ gives the best perplexity because

it is still close to the *Test* domain SPF of 0.23 as reported in Table 1. With the adaptation phase, contrary to our expectation that the synthetic data with the closest SPF to *Test* domain will perform better, higher SPF which represents over code-switched corpus gives the best perplexity of 359. This could have helped the model to learn more possible code-switch combinations than training on data with lower SPF and the adaptation phase, which consists of in domain monolingual data, will fine tune the model.

Table 5: *The effect of different p_{cs} and SPF on perplexity is reported for pre-trained model without fine-tuning.*

p_{cs}	0.2	0.3	0.4	0.5	0.7
SPF	0.22	0.30	0.37	0.42	0.47
Pre-train	476	449	455	391	359

4.4. Effect of Phrase Based Alignment

Lastly, we present the phrase based alignment using verbal phrases extracted from the phrase table based on phrase length and relative frequency. While the effect of phrase length is empirically shown in Table 6, the frequency threshold is selected based on a subjective evaluation of a sample of extracted phrases. We align the parallel corpora based on the extracted phrase and generate the synthetic corpus. $p_{cs} = 0.7$ is used since it gives the best perplexity as reported in Table 5. The best phrase length according to the experiment is 2 and 3 as they both give the same perplexity. Phrase based alignment is consistently better than word based approach due to the effect of switching verbal phrases as discussed in Section 2. Furthermore, we note that longer phrases give lower SPF, which effectively prohibit certain plausible code-switch combinations. However, the negative effect of longer phrases is not revealed in the test result, possibly due to the limited test domain. Overall, we achieve a perplexity of 173 using the proposed method which is 21% improvement over model trained from scratch using only *Train*.

Table 6: *The entire pre-training and fine-tuning process is repeated with different synthetic corpora that are aligned using different phrase length. Phrase length of 1 indicates word aligned synthetic corpus.*

Phrase Length	1	2	3	4	5
Perplexity	183	**173**	**173**	174	174

5. Benchmark against the State-of-the-Art

To put our result into context, we compare the state-of-the-art language models on SEAME dataset. SEAME dataset has two releases: 1) SEAME Phase I [32], and 2) SEAME Phase II [24]. We note that SEAME Phase I approximates to 60% of SEAME Phase II in terms of total tokens as reported in Table 7. We compare our proposed model, denoted as *Synthetic CS*, with the state-of-the-art language models in Table 8. The *Synthetic CS* is pre-trained with *Synthetic* corpus without the adaptation phase, since the monolingual adaptation data is from SEAME. The pre-trained model is then fine-tuned on Phase II Train set and tested on Phase II Eval set. To be consistent with the state-of-the-art language models, we re-train the model using the same $25K$ vocabulary for training and testing. We choose phrase length as 2 based on the results of Table 6. The perplexity of our model is 142.53 which outperforms RNNLM [8], FL+OF [8], LSTM [9] and FLM [4]. We note that our model achieves competitive performance against Multi-task [9] using standard LSTM network. Comparing with LSTM [9], we observe a 6.9% perplexity reduction and further improvement in the network structure will likely improve the perplexity on top of this.

Table 7: *SEAME Phase I & II Database.*

Phase I	Train set	Dev set	Eval set
Tokens	$0.76M$	$31K$	$17K$
Vocabulary (CN+EN)	$(7+10)K$	–	–
Phase II			
Tokens	$1.2M$	$65K$	$60K$
Vocabulary (CN+EN)	$(8+17)K$	–	–

Table 8: *Perplexity of the state-of-the-art language models evaluated on SEAME test set. Models marked with † indicate that training and testing are done on SEAME Phase I. Models marked with * indicate training and testing phases are done on SEAME Phase II.*

Model	**PPL Dev**	**PPL Eval**
RNNLM† [8]	246.60	287.88
FL + OF† [8]	219.85	239.21
FLM† [4]	177.79	192.08
LSTM* [9]	150.65	153.06
Multi-task* [9]	141.86	141.71
Synthetic CS	142.41	142.53

We also conduct ASR experiment on the SEAME database with $101.1hrs$ of training and $11.5hrs$ of evaluation. The ASR system is set up according to [33], whereby the acoustic model is based on time-delay neural network and the language model is trigram. The best WER for the system is 25.25%. To show that the reduction in perplexity also translates to reduction in WER, we perform lattice rescoring using the Synthetic CS model. Our pre-trained language model, without the adaptation phase, is fine-tuned on the Train transcription used in the ASR. The WER dropped from 25.25% to 23.80%, an absolute improvement of 1.45%. To take away the improvement due to RNN language model, we also perform lattice rescoring using a RNN language model without pre-training. Its best WER is 24.11% which is higher than the WER using Synthetic CS model and this shows that the proposed method has practical benefit to the downstream tasks such as ASR.

6. Conclusion

The results support the proposed computation method to generate synthetic code-switch data using the Matrix Language Frame theory. We show that using the synthetic data as a supplement to real code-switch data reduces perplexity by 21% over model trained without synthetic data. To achieve this result we experiment with different phrase length and code-switch probability. The result is comparable to the state of the art models using standard LSTM layer. The Synthetic CS model improves 1.45% in WER when used for lattice rescoring.

7. Acknowledgments

We thank the reviewers for their helpful comments. This research is supported by the project Human-Robot Collaborative AI for AME under RIE2020 Advanced Manufacturing and Engineering Programmatic Grant A18A2b0046. Grandee Lees research is also supported by the NUS Research Scholarship.

8. References

[1] P. Auer, *Code-switching in conversation: Language, interaction and identity*. Routledge, 2013.

[2] Ö. Çetinoğlu, S. Schulz, and N. T. Vu, "Challenges of computational processing of code-switching," in *Proceedings of the Second Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, 2016, pp. 1–11. [Online]. Available: http://www.aclweb.org/anthology/W16-5801

[3] P. Auer, "From codeswitching via language mixing to fused lects: Toward a dynamic typology of bilingual speech," *International journal of bilingualism*, vol. 3, no. 4, pp. 309–332, 1999.

[4] H. Adel, N. T. Vu, and T. Schultz, "Combination of recurrent neural networks and factored language models for code-switching language modeling," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2013, pp. 206–211.

[5] N. T. Vu and T. Schultz, "Exploration of the impact of maximum entropy in recurrent neural network language models for code-switching speech," in *Proceedings of The First Workshop on Computational Approaches to Code Switching*, 2014, pp. 34–41.

[6] H. Adel, N. T. Vu, K. Kirchhoff, D. Telaar, and T. Schultz, "Syntactic and semantic features for code-switching factored language models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 431–440, 2015.

[7] Z. Zeng, H. Xu, T. Y. Chong, E.-S. Chng, and 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)*. IEEE, 2017, pp. 1596–1601.

[8] H. Adel, N. T. Vu, F. Kraus, T. Schlippe, H. Li, and T. Schultz, "Recurrent neural network language modeling for code switching conversational speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8411–8415.

[9] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switching language modeling using syntax-aware multi-task learning," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, 2018, pp. 62–67. [Online]. Available: http://aclweb.org/anthology/W18-3207

[10] Y. Li and P. Fung, "Code-switch language model with inversion constraints for mixed language speech recognition," *Proceedings of COLING 2012*, pp. 1671–1680, 2012.

[11] ——, "Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7368–7372.

[12] ——, "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.

[13] 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)*, vol. 1, 2018, pp. 1543–1553.

[14] S. Poplack, "Sometimes ill start a sentence in spanish y termino en espanol: Toward a typology of code-switching," *The bilingualism reader*, vol. 18, no. 2, pp. 221–256, 2000.

[15] D. Sankoff, "The production of code-mixed discourse," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 8–21.

[16] E. van der Westhuizen and T. Niesler, "Synthesising isiZulu-English code-switch bigrams using word embeddings," *Proc. Interspeech 2017*, pp. 72–76, 2017.

[17] E. van der Westhuizen and T. R. Niesler, "Synthesised bigrams using word embeddings for code-switched ASR of four south african language pairs," *Computer Speech & Language*, vol. 54, pp. 151–175, 2019.

[18] C. Myers-Scotton, *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press, 1997.

[19] A.-M. Di Sciullo, P. Muysken, and R. Singh, "Government and code-mixing," *Journal of linguistics*, vol. 22, no. 1, pp. 1–24, 1986.

[20] 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.

[21] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 328–339.

[22] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[23] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, no. 4, pp. 403–434, 2001.

[24] G. Lee, T.-N. Ho, E.-S. Chng, and H. Li, "A review of the Mandarin-English code-switching corpus: SEAME," in *Asian Language Processing (IALP), 2017 International Conference on*. IEEE, 2017, pp. 210–213.

[25] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, 2016, pp. 923–929.

[26] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, 2012, pp. 2214–2218.

[27] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[28] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing chinese word segmentation for machine translation performance," in *Proceedings of the third workshop on statistical machine translation*. Association for Computational Linguistics, 2008, pp. 224–232.

[29] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010, pp. 2877–2880.

[30] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012, pp. 194–197.

[31] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[32] 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.

[33] P. Guo, H. Xu, L. Xie, and E. S. Chng, "Study of Semi-supervised Approaches to Improving English-Mandarin Code-Switching Speech Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1928–1932. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1974