



Improving Code-switching Language Modeling with Artificially Generated Texts using Cycle-consistent Adversarial Networks

Chia-Yu Li, Ngoc Thang Vu

Institute for Natural Language Processing (IMS), University of Stuttgart, Germany

{licu, thangvu}@ims.uni-stuttgart.de

Abstract

This paper presents our latest effort on improving Code-switching language models that suffer from data scarcity. We investigate methods to augment Code-switching training text data by artificially generating them. Concretely, we propose a cycle-consistent adversarial networks based framework to transfer monolingual text into Code-switching text, considering Code-switching as a speaking style. Our experimental results on the SEAME corpus show that utilizing artificially generated Code-switching text data improves consistently the language model as well as the automatic speech recognition performance.

Index Terms: Code-switching speech, CycleGAN, Data augmentation

1. Introduction

Code-switching (CS) speech is a common phenomenon in multilingual countries and defined as speech that contains more than one language [1]. Recently, CS speech has received great attention in speech communities and is identified as one of the most challenging tasks for automatic speech recognition (ASR) systems, either for hybrid systems (e.g.[2]) or end-to-end systems (e.g.[3]). One of the main challenges is the data scarcity issue, especially in the context of language modeling.

Data augmentation provides a potential solution for data scarcity because it is less time-consuming than collecting and transcribing real speech data and people have shown in many contexts [4, 5, 6, 7, 8] that it improves results. Text generation - a data augmentation method - has been proposed in [9, 10, 11, 12, 13, 14, 15] with the aim of improving language models and therefore automatic speech recognition performance for CS speech. The first approach [9, 13, 14, 15] breaks down the main task in two consecutive steps: first to predict CS start and end points and then to replace a certain number of words or phrases in monolingual text with translated words or phrases in another language using either rules or machine learning approaches. This approach suffers from the error propagation problem. The second approach [10, 11] leverages recurrent neural network language models trained with CS transcriptions. Because of the small amount of CS transcriptions, the generated texts are often meaningless [10]. The third approach [12] formulates the CS text generation task as a sequence to sequence problem, mapping monolingual text to CS text and solves the task in an end-to-end fashion using sequence-to-sequence model [16, 17]. This model, however, requires a large amount of parallel training data. Moreover in other contexts of text generation, it tends to generate ill-formed text [18, 19].

Recently, CycleGAN - a variation of generative adversarial networks (GANs) [20] for style transfer - has been proposed for translating images [21] and then applied in other domains [22, 8]. The main idea of this model is to transfer one style of the source domain to another style of the target domain by

learning a mapping between them without any parallel training data. The first attempt on text was done in [23] in which the authors interpreted text styles in a common sense and proposed model to modify sentiment of the text and to transform the text title between paper and news styles.

In this work, we propose the novel idea of considering Code-switching as a speaking style and therefore to use CycleGANs [21] to generate artificially text data for two reasons: first, CycleGAN transfers styles without any parallel training data; second by doing so, CS text generation can be solved in an end-to-end fashion, meaning that the two tasks - CS points prediction and short text translation - will be jointly optimized to mimic the CS distributions in the real CS data and to maintain the same meaning when translating from one language to another. In order to successfully generate artificially CS text, we explore tricks how to train this complex model by using sequence-to-sequence model and by leveraging monolingual data, and investigate the impact of the cycle losses in CycleGAN on the language model and the automatic speech recognition performance.

In sum, our contributions are as follows: 1) To the best of our knowledge, we are the first to propose a novel framework based on the CycleGAN architecture for generating artificially CS texts from monolingual texts for language modeling; 2) We show that our CS generated texts contribute to improve not only the language modeling performance but also the ASR system on the SEAME corpus.

2. Proposed Method

2.1. Sequence-to-sequence (S2S) Model

The sequence-to-sequence model [16] proposed an end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. In the CS context, the S2S model takes a Chinese sentence as input and generates the Chinese-English CS sentence. For example, given a sequence of inputs (Chinese sentence) (x_1, x_2, \dots, x_T) , the goal of the S2S models is to estimate the conditional probability:

$$P(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{i=1}^{T'} P(y_i | v, y_1, \dots, y_{i-1}) \quad (1)$$

where $(y_1, \dots, y_{T'})$ is the corresponding output sequence (Chinese-English CS sentence) whose length T' may differ from T . v is the representation of the input sequence (x_1, \dots, x_T) given by the last hidden state of the encoder. The model is trained in a supervised manner. In CS, the trick is to generate paired training data by utilizing Chinese texts and their partial dictionary-based translations that form artificially generated CS texts [9]. The idea of this simple method is to replace a certain number of words or phrases in monolingual texts with

their translations based on a dictionary, or, as in this work, using the Google Translation API. Our intuition is that the S2S model will learn more generalized patterns in CS behaviours by taking advantage of the continuous latent semantic space and thereby generate more CS variations than the dictionary based method.

2.2. CycleGAN-based Model

The main disadvantage of the previous proposed method is that it does not make use of any prior knowledge encoded in an existing CS corpus. Therefore, we propose to employ the pre-trained S2S models, which are $G : X \rightarrow Y$ and $F : Y \rightarrow X$ in Figure 1, in the CycleGAN architecture to take advantage of both methods: the S2S generation framework and the CS prior knowledge encoded in a 'Discriminator' without any real paired data (monolingual text and its corresponding CS text).

CycleGAN is a currently popular technique to solve image to image translation problems where the goal is to learn a mapping G between an input image from a source domain X and an output image from a target domain Y without using paired training data [21]. The mapping $G : X \rightarrow Y$ is learnt such that the distribution of images from $G(X)$ is indistinguishable from the distribution Y using an adversarial loss. Because this mapping is highly under-constrained, [21] coupled it with an inverse mapping $F : Y \rightarrow X$ and introduce a cycle consistency loss to push $F(G(X)) = X$ (and vice versa). The full objective is

$$\begin{aligned} \mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{GAN}(G, D_Y, X, Y) \\ & + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \end{aligned} \quad (2)$$

where G tries to generate images $G(x)$ that look similar to images from domain Y , while D_Y aims to distinguish between translated samples $G(x)$ and real samples y . The same holds for the Discriminator D_X which tries to distinguish between $F(Y)$ and real x . λ controls the relative importance of cycle consistency loss:

$$\begin{aligned} \mathcal{L}_{cyc}(G, F) = & \mathbf{E}_{x \sim p_{data}(x)} [\| F(G(x)) - x \|_1] \\ & + \mathbf{E}_{y \sim p_{data}(y)} [\| G(F(y)) - y \|_1] \end{aligned} \quad (3)$$

In the context of CS, X refers to monolingual text (e.g. Chinese text) and Y refers to CS text. We define that

$$\begin{aligned} \mathcal{L}_{cyc}(F(G(X))) = & \mathbf{E}_{x \sim p_{data}(x)} [\| F(G(x)) - x \|_1] \\ \mathcal{L}_{cyc}(G(F(Y))) = & \mathbf{E}_{y \sim p_{data}(y)} [\| G(F(y)) - y \|_1] \end{aligned} \quad (4)$$

Furthermore, we adapt the objective to the following equation:

$$\begin{aligned} \mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{GAN}(G, D_Y, X, Y) \\ & + \mathcal{L}_{GAN}(F, D_X, Y, X) \\ & + \lambda_1 \mathcal{L}_{cyc}(F(G(X))) \\ & + \lambda_2 \mathcal{L}_{cyc}(G(F(Y))) \end{aligned} \quad (5)$$

where λ_1 and λ_2 are tunable parameters to control the importance of the two cycle consistency losses. Moreover, we can tune the balance between them.

Figure 1 shows the network architecture of CycleGAN for generation of CS text. There are two Generators G and F : one transforms a Chinese sentence to a CS one, another converts a CS sentence to a Chinese one. The goal of the two Discriminators (D_X and D_Y) is to predict whether the sample is from the actual distribution ('real') or produced by the Generator ('fake'). The constraint of cycle consistency enforces the network to use shared latent features that can be used to map this output back to the original input. It could prevent the Generator

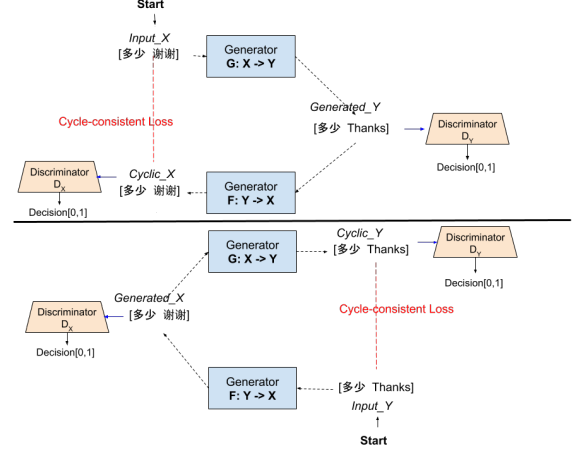


Figure 1: Network architecture of CycleGAN for CS text generation. X refers to monolingual text, e.g. Chinese and Y refers to CS text, e.g. Chinese English CS sentences.

from generating the same sentence in the target domain while the input is quite different. The upper part of Figure 1 describes how we compute the cycle consistency loss $\mathcal{L}_{cyc}(F(G(X)))$ and the lower part is for $\mathcal{L}_{cyc}(G(F(Y)))$.

3. Resources

3.1. SEAME Dataset

We use SEAME - the Mandarin-English Code-switching spontaneous speech corpus in South East Asia that contains 99 hours of spontaneous Chinese-English speech recorded from Singaporean and Malaysian. All recordings are done with close-talk microphones in a quiet room. The speakers are aged between 19 and 33 and almost balanced in gender (49.7% female and 50.3 % male). 16.96% of utterances are English, 15.54% are Chinese and the rest (67%) are CS utterances [24].

3.2. Code-mixing Index (CMI)

In order to compare the artificially generated CS sentences with the real CS corpus, we use Code-mixing Index (CMI) at the utterance level introduced in [25] as a measurement of the level of mixing between languages, that is

$$CMI = \begin{cases} 100 \times [1 - \frac{\max\{w_i\}}{(n-u)}], & n > u \\ 0, & n = u \end{cases} \quad (6)$$

where n is the total number of tokens, u is the number of other non-verbal tokens (e.g. people laughing) and $\max\{w_i\}$ is the number of tokens from the dominant language. If an utterance only contains non-verbal tokens (i.e., if $n = u$), it has an index of zero. The CMI ranges from $[0, 50]$. The lower the value, the lower the level of mixing between languages. Because this index does not indicate the dominant language in the utterances, we add the dominant language tag before the CMI and group them into ten groups presented in Table 1. It illustrates ten groups of different mixing levels and the proportion of each group in the SEAME data set. For example, EN-C2 presents a portion of data with English as the dominant language and the CMI ranges (0,15]. It is 18%, 16% and 15% of the SEAME train, dev and eval set, respectively.

Table 1: *Different levels of code-mixing in the SEAME corpus.*

Groups	CMI	train (%)	dev (%)	eval (%)
ZH-C1	0	1	1	3
ZH-C2	(0,15]	18	16	15
ZH-C3	(15,30]	25	26	24
ZH-C4	(30,45]	21	23	20
ZH-C5	(45,50]	8	9	9
EN-C1	0	0	0	0
EN-C2	(0,15]	4	3	3
EN-C3	(15,30]	9	9	11
EN-C4	(30,45]	13	11	14
EN-C5	(45,50]	1	1	1

4. Experimental setup

4.1. Monolingual Datasets

We utilize two Chinese datasets to generate Code-switching text which were used to train the S2S RNN model and the pretrained generators in the CycleGAN: 1) Aishell-1 contains 96,078 sentences contributed by 400 people in China and covers “Finance”, “Science and Technology”, “Sports”, “Entertainments” and “News” topics [26]. 2) The National University of Singapore SMS Corpus is a corpus of Short Message Service (SMS) messages collected for research at the Department of Computer Science at the National University of Singapore. It consists of 29,031 SMS messages in Chinese and the messages largely originate from Singaporeans and mostly from students attending the University [27].

4.2. Baseline Systems

The baseline language model is trained with SEAME training text containing 94,504 sentences that includes monolingual (English or Chinese) and CS sentences. The word-based language model is one layer of LSTM with 650 units and trained with a batch size of 20. It achieves perplexities of 84 and 75 on “dev” and “eval” set, respectively. The subword-based language model is a two layers RNN model with 650 units and trained with the batch size of 256 (Espnet default settings). It has a perplexity of 12.37 on “dev” and 10.90 on “eval” set. The baseline end-to-end (E2E) ASR is trained by Espnet [3] using our best recipe presented in [28]. The mixed error rate (MER), which is the error rate on Chinese characters and English words, on SEAME “dev” and “eval” set are 30.7% and 23.0% without fusing any external language model, respectively. Integrating the baseline language model to our E2E ASR using shallow fusion results in a MER of 30.4% on “dev” and 22.4% on “eval” set.

4.3. Implementation & Hyperparameters

The S2S and CycleGAN models are implemented using Python3 and Pytorch. Both S2S and the two Generators in CycleGAN use RNN encoder decoder that consists of two RNNs with 650 LSTM units. The models are trained with 96,078 Chinese sentences from Aishell-1 and their corresponding CS artificially generated sentences. The two Discriminators in CycleGAN are trained with SEAME training data containing 94,504 sentences and 96,078 Chinese sentences from the Aishell-1 dataset. The dimension of word embeddings is 300 and the batch size is 64. The optimizer is Adam [29]. The best tunable parameters (λ_1 and λ_2) in the objective of CycleGAN on the development set are 0.3 and 0.8, respectively.

5. Results

5.1. Language Model Evaluation

Table 2 shows the perplexity of two different language models (word-based LM and subword-based LM) on SEAME “dev” and “eval” set. We use the subword-based LM because our best E2E ASR system [28] outputs subwords. Both are trained with three different texts: SEAME training text, SEAME text adding the text from S2S model, and SEAME text adding the text from CycleGAN. The results show that the language model trained with SEAME and the generated text from CycleGAN outperforms the baseline LM and the one trained with SEAME and the generated text from S2S model on the “eval” set.

Table 2: *The PPL on SEAME “dev” and “eval” set*

Word-based LM	dev	eval
SEAME	84.00	75.00
+text from S2S	83.03	74.67
+text from CycleGAN	83.06	72.59
Subword-based LM	dev	eval
SEAME	12.37	10.90
+text from S2S	12.13	10.80
+text from CycleGAN	10.39	9.59

5.2. End-to-End Speech Recognition Evaluation

The subword-based LM is integrated to the E2E speech recognition system using shallow fusion [28], and the MERs on SEAME “dev” and “eval” set are presented in Table 3 and 4. ‘Overall’ means the entire “dev” or “eval” set, and ‘EN’, ‘ZH’ and ‘CS’ are denoted as the monolingual English sentences, monolingual Chinese sentences and Chinese-English CS sentences in “dev” or “eval” set, respectively. The results on both sets show that the MER is improved most not only on CS utterances but also on monolingual ones when the language model trained with the combination of SEAME text and the generated CS text from CycleGAN.

Table 3: *The MER on SEAME “dev” set*

Subword-based LM	Overall	EN	ZH	CS
SEAME	30.4	43.4	23.8	30.3
+text from S2S	30.3	42.8	23.4	30.2
+text from CycleGAN	30.1	42.6	23.2	30.0

Table 4: *The MER on SEAME “eval” set*

Subword-based LM	Overall	EN	ZH	CS
SEAME	22.4	32.4	22.7	21.5
+text from S2S	22.1	33.0	23.3	21.2
+text from CycleGAN	21.9	32.5	22.1	21.0

6. Quantitative & Qualitative Analysis

6.1. Impact of λ_1 and λ_2

Table 5 shows the perplexity of subword-based language model on SEAME dev sets with different values of λ_1 and λ_2 . $\lambda_1 = 0.3$ seems to deliver the best performance in most cases. The E2E systems that integrate language models trained with text

generated using CycleGAN framework either with $\lambda_1 = 0.3$ and $\lambda_2 = 0.8$ or with $\lambda_1 = 0$ and $\lambda_2 = 0.6$ achieve the best MER on both SEAME “dev” and “eval” set.

Table 5: The PPL of subword-based LM on SEAME “dev” set

$\lambda_2 \setminus \lambda_1$	0	0.1	0.2	0.3	0.4
0.5	11.56	12.09	12.18	11.11	10.62
0.6	11.02	11.14	11.66	10.81	11.64
0.7	11.05	12.22	10.57	10.56	11.07
0.8	11.94	11.82	11.87	10.39	11.62
0.9	10.43	11.23	11.12	11.96	10.77

Table 6: Different levels of code-mixing in the SEAME text and, in the texts generated from S2S and CycleGAN.

Groups	CMI	SEAME (%)	S2S (%)	CycleGAN (%)
ZH-C1	0	1	0	0
ZH-C2	(0,15]	17	13	12
ZH-C3	(15,30]	25	12	20
ZH-C4	(30,45]	21	12	22
ZH-C5	(45,50]	9	13	5
EN-C1	0	0	0	2
EN-C2	(0,15]	4	13	10
EN-C3	(15,30]	9	12	11
EN-C4	(30,45]	13	13	18
EN-C5	(45,50]	1	12	0

6.2. Generated Text

We analyzed 148,853 generated CS sentences from S2S and CycleGAN and show some interesting examples in this subsection. Table 6 shows that when comparing the distribution of CMI groups, texts generated by CycleGAN have a more similar distribution to the ones in SEAME than texts generated by S2S across all different levels of code-mixing. This fact suggests that CycleGAN successfully enforces the generated texts to mimic CS behaviour in the original corpus.

Table 7: The output from S2S and CycleGAN models

Monolingual	提高 铁路 在物流 市场 中的竞争力
S2S	improve 铁路 在 logistics market in 的竞争力
CycleGAN	improve railway in 2016 market in 的竞争力

While we aim at generating texts with the same meaning of the inputs, we observed an interesting effect as shown in an example in Table 7. While the output of S2S contains Chinese words being replaced by their translations, CycleGAN produces words with different meanings (“2016” marked in bold instead of the English noun “logistics”). In this case, CycleGAN transforms the input sentence into a different meaningful CS one instead of just replacing Chinese words by their translations.

Furthermore, we observe that text generated using CycleGAN framework might simulate speaking styles used in SEAME. Table 8 presents two examples in which CycleGAN generates text with some repetitive words, mimicking real speakers in SEAME data who tends to stutter.

Table 8: CycleGAN simulates stuttering speaking style

Reference (NI01MAX0101)
take take 多一个 then 就 then 就 这 but 那 边 的 东 西 最 好 了 我 觉 得 in terms of of 那
CycleGAN
and 如 果 take take out of ten 根据 one series of of of 问题

6.3. Analysis of Recognition Results

The top of Table 9 presents the recognition performance comparison in terms of average substitution, deletion, insertion and MER between the baseline and the improved ASR systems. The baseline system has less deletion, while adding artificially generated CS text from CycleGAN results in better substitution and insertion rates and the overall MER. We show some cherry picked ASR output examples in the bottom of Table 9 that show the positive impact of using artificially generated CS text data on improving the ASR performance in both monolingual (English and Chinese) and CS speech utterances.

Table 9: ASR performance comparison in terms of substitution, deletion, insertion and MER on the eval set and some examples.

System	Sub.	Del.	Ins.	MER
Baseline	13.6	4.8	4.0	22.4
+text from S2S	13.1	5.1	3.9	22.1
+text from CycleGAN	13.1	5.1	3.7	21.9

Reference (EN example)	SHARED MICROSCOPE
System	
Baseline	SHARED MY CROSCORD 吗
+text from S2S	SHARED MY CROSCORD 吗
+text from CycleGAN	SHARED MICROSCOPE
Reference (CS example)	不 ^a concern about this
System	
Baseline	book concern about this
+text from S2S	book concern about this
+text from CycleGAN	不 concern about this
Reference (ZH example)	不厉害打架的
System	
Baseline	不厉害大家的
+text from S2S	不厉害大家的
+text from CycleGAN	不厉害大家的

^aThe pronunciation (IPA) of Mandarin character “不” is [bu]

7. Conclusion

We explored a data augmentation method based on CycleGAN to generate CS text with the goal of improving language modeling and further ASR performance. We considered CS as a speaking style and extended the CycleGAN framework to convert monolingual text to CS text. Our experiments showed that we could generate CS text that not only mimics the CS behaviours in terms of different levels of mixing languages but also the speaking styles in the SEAME corpus. Furthermore, our experiments on SEAME show promising results with consistent improvements in perplexities for language modeling as well as MERs for ASR. Future work could explore in depth differences between the real CS corpus and the artificially generated one with respect to linguistic properties.

8. References

- [1] P. Auer, *Code-switching in conversation: Language, interaction and identity*. Routledge, 2013.
- [2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The Kaldi Speech Recognition Toolkit,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.
- [3] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “Espnet: End-to-end speech processing toolkit,” in *Proceedings of Interspeech*, 2018, pp. 2207–2211.
- [4] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in *Proceedings of Interspeech*, 2015, pp. 3586–3589.
- [5] A. Gorin, R. Lileikyte, G. Huang, L. Lamel, J.-L. Gauvain, and A. Laurent, “Language model data augmentation for keyword spotting in low-resourced training conditions,” in *Proceedings of Interspeech*, 2016, pp. 775–779.
- [6] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.
- [7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proceedings of Interspeech*, 2019, pp. 2613–2617.
- [8] F. Bao, M. Neumann, and N. T. Vu, “CycleGAN-based Emotion Style Transfer as Data Augmentation for Speech Emotion Recognition,” in *Proceedings of Interspeech*, 2019, pp. 2828–2832.
- [9] N. T. Vu, D. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E. Chng, T. Schultz, and H. Li, “A first speech recognition system for mandarin-english code-switch conversational speech,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4889–4892.
- [10] 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@EMNLP*, 2014, pp. 34–41.
- [11] E. Yilmaz, H. van den Heuvel, and D. A. van Leeuwen, “Acoustic and Textual Data Augmentation for Improved ASR of Code-Switching Speech,” in *Proceedings of Interspeech*, 2018, pp. 1933–1937.
- [12] G. I. Winata, A. Madotto, C. Wu, and P. Fung, “Learn to code-switch: Data augmentation using copy mechanism on language modeling,” *CoRR*, vol. abs/1810.10254, 2018.
- [13] C.-T. Chang, S.-P. Chuang, and H.-Y. Lee, “Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation,” in *Proceedings of Interspeech*, 2019, pp. 554–558.
- [14] Y. Gao, J. Feng, Y. Liu, L. Hou, X. Pan, and Y. Ma, “Code-Switching Sentence Generation by Bert and Generative Adversarial Networks,” in *Proceedings of Interspeech*, 2019, pp. 3525–3529.
- [15] G. Lee, X. Yue, and H. Li, “Linguistically Motivated Parallel Data Augmentation for Code-Switch Language Modeling,” in *Proceedings of Interspeech*, 2019, pp. 3730–3734.
- [16] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in *Proceedings of Neural Information Processing Systems (NIPS)*, 2014, pp. 3104–3112.
- [17] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.
- [18] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A Diversity-Promoting Objective Function for Neural Conversation Models,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2016, pp. 110–119.
- [19] L. Luo, J. Xu, J. Lin, Q. Zeng, and X. Sun, “An Auto-Encoder Matching Model for Learning Utterance-Level Semantic Dependency in Dialogue Generation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 702–707.
- [20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.
- [21] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.
- [22] T. Kaneko and H. Kameoka, “Parallel-data-free voice conversion using cycle-consistent adversarial networks,” *CoRR*, vol. abs/1711.11293, 2017.
- [23] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, “Style transfer in text: Exploration and evaluation,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018, pp. 663–670.
- [24] Nanyang Technological University, Universiti Sains Malaysia, “Mandarin-English Code-Switching in South-East Asia.” [Online]. Available: <https://catalog ldc.upenn.edu/LDC2015S04>
- [25] B. Gambäck and A. Das, “On Measuring the Complexity of Code-Mixing,” in *Proceedings of the 1st Workshop on Language Technologies for Indian Social Media (SOCIAL-INDIA)*, 2014.
- [26] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “AISHELL-1: An Open-Source Mandarin Speech Corpus and A Speech Recognition Baseline,” in *Proceedings of the International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (Oriental COCOSDA)*, 2017, pp. 1–5.
- [27] T. Chen and M.-Y. Kan, “Creating a Live, Public Short Message Service Corpus: The NUS SMS Corpus,” *Language Resources and Evaluation*, pp. 299–355, 2013.
- [28] C.-Y. Li and N. T. Vu, “Integrating Knowledge in End-to-End Automatic Speech Recognition for Mandarin-English Code-Switching,” in *Proceedings of International Conference on Asian Language Processing*, 2019, pp. 160–165.
- [29] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.