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

Code-Switching Sentence Generation by Bert and Generative Adversarial Networks

Yingying Gao, Junlan Feng, Ying Liu, Leijing Hou, Xin Pan, Yong Ma

China Mobile Research

{gaoyingying, fengjunlan, liuyingzn, houleijing, panxinyjy, mayongyjy}@chinamobile.com

Abstract

Code-switching has become a common linguistic phenomenon. Comparing to monolingual ASR tasks, insufficient data is a major challenge for code-switching speech recognition. In this paper, we propose an approach to compositionally employ the Bidirectional Encoder Representations from Transformers (Bert) model and Generative Adversarial Net (GAN) model for code-switching text data generation. It improves upon previous work by (1) applying Bert as a masked language model to predict the mixed-in foreign words and (2) basing on the GAN framework with Bert for both the generator and discriminator to further assure the generated sentences similar enough to the natural examples. We evaluate the effectiveness of the generated data by its contribution to ASR. Experiments show our approach can reduce the English word error rate by 1.5% with the Mandarin-English code-switching spontaneous speech corpus OC16-CE80.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics, code-switching

1. Introduction

With rapid developing of global business and culture merging, the code-switching speaking style has become increasingly common in our daily life from casual chats to formal meetings. It is a specially popular phenomena in multi-cultural areas, or countries where there are more than one official language or the official language is different from the native language. Table 1 lists a few examples how English words are code-switched in Mandarin sentences. Compared to the monolingual scenario, the code-switching utterances pose unique challenges for ASR. One is phoneme confusion from more than one language. Second is inadequate data for training code-switching acoustic and language models. In this paper we focus on handling the second challenge. As [1] pointed out, inadequate data is partially caused by the inconsistency between spoken language and written language. People tend to mix foreign words in spoken language more often than in written language. Hence, it is difficult to collect large amount of code-switching text data for training effective language models.

Table 1: *Code-Switching Sentence Examples*

一位(One) 朋友(Friend) 在(On) facebook 上面 说(Said)
siri 真(Really) 什么 都 知道(Know anything) 啊
他(He) 让(Let) 我(Me) 在 google 里 百度(Baidu) 一下
又做(Again) 夜猫子(Stay late) 好 吧(Well) good night
今天(Today) 晚上(Evening) 去(Go) happy 一下

To address this issue, researchers and practitioners have proposed several lines of approaches. Earlier efforts are mainly based on rules such as the minimum frequency. According to these rules, [2] selects 102 Chinese sentences in which 311 Chinese words are translated into English to generate 821 code-switching sentences. [3] generates code-switching text under a set of frequency-based, lexeme-based and POS-tag-based rules. It works as follows. English segments are first found in transcriptions of a Mandarin-English speech corpus (SEAME) [4] and then translated to Chinese segments with a translation model. The obtained Chinese segments are searched in a large Chinese text corpus and replaced with their English counterparts. This search and replace (S&R) approach is further fine tuned with a set rules such as (1) limiting the replacements to segments which occur at least twice in the training text, (2) replacing the found segments if the word preceding the segment is a trigger word or a trigger part-of-speech tag , which are selected based on dictionaries manually built upon observations of the SEAME data, etc.

These rules are clever and help ease the problem at a certain degree. However, they are far from being a functional solution in real applications. Code switching is a complicated linguistic behavior heavily relying on multi-factors such as context, culture and individual preference. This underlying complexity is unlikely to be captured by a set of hard rules. Recently researchers have appealed to learning statistical models to generate code-switching data from monolingual text resources. [1] reported a GAN based model to generate Chinese-English text data with Chinese as the host language and English as the guest language. The generator takes as input a Chinese sentence such as “好(Okay) 我(I)知道(Know)” and generates a sequence of probabilities which respectively corresponds to each Chinese word and represents whether it is proper to be replaced with its English counterparts. Then the synthesized output might be like “Okay 我(I)知道(Know) ”. The discriminator in GAN takes this output as input and predicts if the sentence is natural or artificial. The generator is a 4-layer neural network with the first layer as the word embedding layer, the second layer as a bidirectional LSTM, the third layer being a fully connected layer, and the output layer as a simple sigmoid probability. The discriminator shares the embedding layer and bidirectional LSTM with the generator. This approach proves to be better than rule-based generation.

These proposed techniques successfully push up the accuracy of code-switching speech recognition, however the gap is still evident from the monolingual tasks. We argue there are two major reasons. One is the size of training data is not big enough to cover the variations. Second, the experimented models so far are relatively simple comparing to the complex nature of the code-switching problem. In this paper, we approach these challenges by the following: (1) We analyze our observation of the Mandarin-English code-switching data to reveal the complexity and the gap between the nature of the problem and the current proposed approaches. (2)Inspired by our observation

http://dx.doi.org/10.21437/Interspeech.2019-2501

and the powerfulness of Bert [5], we propose to apply the Bert pre-trained model and the deep self-attention architecture of it to generate code-switching data from Chinese text resources. (3) We experiment with the GAN framework to have the Bert masked LM as the base for the generator and the discriminator to assure the closeness of generated code-switching data and the natural ones. The theoretical and experimental details are reported in the rest of this paper.

2. Our Approach

2.1. Code-switching Data Analysis

After manually checking many examples, we categorize the data we observe into the following four categories according to the underlying reasons they are code-switched:

1. Vocabulary extension: Typical examples are popular brand names, product names, or technology terms in English such as *wifi, imax, ipad, hotdog, Gucci, latex, Hotmail, moto* etc. These words augment the standard Chinese vocabulary and often don't have compact and widely accepted counterparts in Mandarin. They are more likely nouns and barely influence the Chinese grammar.

2. Word translation: Sentences are natural before and after the Chinese words are substituted with English words. "对我(To me) 说再见(Say bye)" is such an example. "对我(To me) say bye" is equally natural.

3. Phrase or sentence transformation: Some English words, after being frequently code-switched into Mandarin, maintain their core meaning in English and at same time transform its context including the surrounding words and the host language grammar. A sentence including these words wouldn't be natural if the Mandarin counterparts are used to replace them. For instance, the word *hold* in "两个(Two) 宝贝(Babies) 一起(Together) cute 我(I) hold 不住(Can't) 了" can't be directly replaced by 坚持. At the phrase level, a good example "江南(Jiangnan) style", a popular song name, mixes a Chinese noun 江南*(Jiangnan)* and the English word *style*.

4. Spoken citation: such as "那书*(The book) CoCo Chanel* 特别 *(So)* 棒*(Great)*".

It is not hard to see that word-level translation-based approaches that we summarized in Introduction won't be able to generate data for the 1st, 3rd and 4th cases. State-of-art model-based language generation techniques at same time are not mature enough to generate open-domain informative text [6].

2.2. Bert-based Data Generation

Bert is a new language representation model, which stands for Bidirectional Encoder Representations from Transformers. Transformer is a model architecture relying entirely on an attention mechanism to draw global dependencies between input and output. It has been ubiquitously used for various tasks since it was proposed [7]. Bert built on Transformers contains a number of layers(Transformer blocks) L . Each layer is identical with a fixed number of hidden units H and a fixed number of multithreading self-attention heads A. Particularly we use the $BERT - Chinese_{BASE}$ (Bert-C) model with $L = 12, H = 768, A = 12$ as hyperparameters. It was trained by Google on using a giant Wikipedia text dump [5].

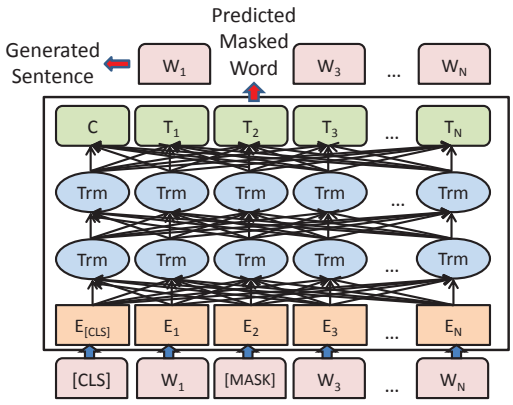

Figure 1: *The process of generating a sentence by Bert.*

We propose to apply Bert to generate Mandarin-English code-switching data from monolingual sentences to overcome some of the challenges we observed with the current start-of-art models. We are motivated by a few thoughts. First, the pre-trained Bert representations can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of downstream tasks, such as question answering and language inference, without substantial task-specific architecture modifications. We believe code-switching data generation is another downstream task, which can take advantage of Bert. Second, the Bert-C model was trained on the Chinese Wikipedia data, which contains a decent size of mixed-in English words, though the way English words switched in might be not as popular and representative as our spontaneous speech data. Hence it is a good starting point. Third, we are motivated to generate code-switching data on sentence level and considering longer context such as examples for the 3rd and 4th categories discussed in Section 2.1. Bert is designed to consider long dependency.

In our experiments, we examine if experimental results meet our arguments. We use the Bert-C model exactly as it is. Our training data to fine tune the model are Mandarin-English code-switching sentences. English words in these sentences are then masked as input. The training goal is to predict these words using its both left and right context. This is almost exactly same setup as the Bert pre-training process. While training we initiate all model parameters with the pre-trained Bert-C model and fine tune all parameters to maximize the probability to predict the code-switched English words. In experiments, we try both masking only the English words and randomly masking Chinese words. Figure 1 illustrates the data generation process.

2.3. GAN with Bert for Generation

We propose in this Section to use GAN together with Bert for the Generator and the Discriminator to generate code-switching data. Figure 2 shows the diagram of the proposed method. The data generator generates a code-switching sentence from a mono-lingual sentence. The discriminator takes a sentence as input and outputs a probability indicating if the input sentence is generated by the Generator or from real. Both the generator and the discriminator are based on the Bert-C model. The generator uses only the $BERT_{BASE}$ architecture and the pre-trained model, while the discriminator adds a softmax layer to output discriminative probabilities.

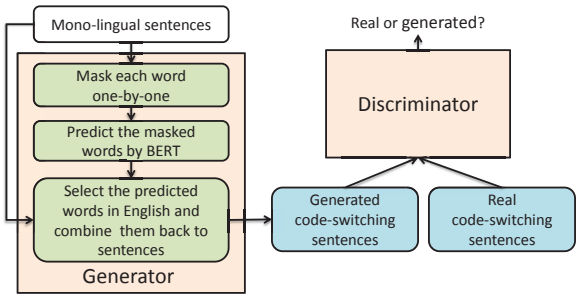

Figure 2: *The diagram of the GAN method with Bert.*

The loss function of the discriminator in GAN is as below:

$$L_D = \frac{1}{M}\sum_i \log(1 - D(x_i)) + \frac{1}{M}\sum_j \log(D(g_j)) \quad (1)$$

where x_i is the real sentence sent into the discriminator; g_j is the generated sentence, i and j are the sentence indices respectively; M is the number of training examples. $D(\mathbf{.})$ refers to the discriminative probability scaled from 0 (generated) to 1 (real). The learning target is to minimize the average loss of the discriminator, which means to maximize the score of the discriminator $D(x_i)$ for real data and minimize the score of the discriminator $D(g_j)$ for the generated one.

The loss function of the generator we use is given in Equation (2). The former one L_{G_o} is the generator loss function in GAN, which aims at maximizing the discriminative probability of the generated data. The second part L_{masked_lm} is the masked LM loss which ensures the accuracy of the word prediction [5].

$$L_G = L_{G_o} + L_{masked\_lm} \quad (2)$$

$$L_{G_o} = \frac{1}{M}\sum_j \log(1 - D(g_j)) \quad (3)$$

To encourage GAN to generate English words, we further change the target of maximizing the score $D(g_j)$ in Equation (3) into maximizing the score of the generated data in which the predicted tokens are in English $D(g_j^{Eng})$. In this way, we hope to increase the possibility of English tokens to be predicted in the generated data. Accordingly, $D(g_j)$ in the loss function of discriminator L_D is also changed into $D(g_j^{Eng})$.

3. Experiments

We apply three criteria to evaluate our proposed models. First, we evaluate the English word prediction accuracy of the Bert model and the Bert-GAN model. Second, we use perplexity (PPL) to measure the language model (LM) that we trained on data augmented by generated code-switching sentences. Third, the obtained LMs are used for ASR. The contribution of the generated code-switching data are measured by ASR word error rate reduction ratio.

3.1. Corpus and Model Setup

We use the Mandarin-English code-switching spontaneous speech corpus OC16-CE80 [8] in our experiments. The entire corpus is divided into three sets, namely, 58,132 utterances for training, 6,974 utterances as the development set and 4,328 utterances for testing. Chinese words make up 80% of the data, making it the predominate language in this corpus. The transcriptions of another open source Mandarin speech corpus AISHELL-ASR0009-OS1 [8] are used as the monolingual texts to generate new code-switching data which contains 141,600 sentences in total. A bi-lingual lexicon is used in speech recognition, which comprises 126,911 Chinese words from CVTE dictionary[9] and 6,229 English words from TIMIT dictionary, together with 26 English letters. The phone set consists of 217 phones for Chinese and 77 for English.

Most of the hyperparameters of Bert-C are consistent with the original configuration in [5], except that the batch size is reduced to 32, the maximum sequence length is set to 128 and the learning rate is fixed as 2e-5. The number of training steps is determined through development data sets and set to 2000, where the loss decreases quite slowly. The GAN model is trained with 10 iterations based on the fine-tuned Bert model. The discriminator is trained 3 steps in each iteration.

We built the ASR engine by following the Aishell2 recipe [10] in Kaldi [11]. We use the time delay neural network (TDNN) [12] as acoustic model (AM) with Relu as the activation function. The network consists of 11 hidden layers with 1,280 nodes each, and the output dimension is 4,416. The input features are Mel frequency cepstral coefficients (MFCC). The word-level tri-gram LM is chosen, which is trained by SRILM [13], using Kneser-Ney (KN) smoothing [14]. Both the AM and the LM are trained by the training set and the development set in OC16-CE80.

3.2. English Word Prediction

Several models are compared on the prediction accuracy of English words using the development set of OC16-CE80 corpus: 1) the Bert-C model; 2) Bert-C fine tuned by code-switching training set in OC16-CE80, where the predicted words during training are randomly masked (FineTuned-RandMask) and only the English tokens are masked (FineTuned-EngMask); 3) GAN with fine-tuned Bert-C without data selection module in Figure2 (FineTuned-EngMask+GAN); 4) and GAN with fine-tuned Bert-C and with data selection that selects the generated sentences in which the predicted tokens are in English during the learning of GAN (FineTuned EngMask+GAN+sel).

Table 2 presents the prediction accuracies of different models. The Bert-C model gets a poor performance on English prediction. This is attributed to the fact that Bert-C is mainly trained by Chinese corpus and English words are far less than Chinese ones. After fine tuned by code-switching texts, Bert-C performs much better. And masking only English words during fine-tuning surpasses masking at random. Taking this English-masked fine-tuned Bert-C as pre-trained model for generator, we add adversarial learning on it. The bottom two lines in Table 2 demonstrate that GAN helps the generator produce more correct English predictions, and the data selection module further improves the performance. This final model is used in the following experiments.

Table 2: *The prediction accuracy of the masked English words*

Model	ACC
Bert-C	13.71%
+FineTuned-RandMask	42.69%
+FineTuned-EngMask	55.61%
+FineTuned-EngMask + GAN	58.67%
+FineTuned-EngMask + GAN+sel	**60.00**%

3.3. Language Model Evaluation

The mono-lingual texts from AISHELL corpus are transformed into code-switching texts through the proposed generator in Figure 2. The quality of the generated texts is evaluated through the original code-switching LM. The results are listed in Table 3. According to this table, we find that the AISHELL mono-lingual texts have a quite large PPL, which demonstrates the mismatching between AISEHLL corpus and OCC16-CE80 corpus. The generated mixed-lingual texts have a much smaller PPL, which indicates that the generated texts are more similar with the real code-switching texts. Furthermore, we introduce a threshold via log probabilities of the predicted words to select more reliable generated texts. From the last two lines in Table 3, we validate the efficiency of the threshold, and when the threshold is more strict ($logP \geq -1$), the PPL further decreases.

Table 3: *The perplexities of the generated data evaluated by original LM*

Data	PPL
AISHELL mono	3591.66
Generated mix	1935.97
Generated mix $logP \geq -2$	1879.25
Generated mix $logP \geq -1$	**1755.70**

The newly generated texts are then added to the training data to train new LMs. Table 4 shows the PPLs of the test set in OCC16-CE80 evaluated by different LMs. The second line in Table 4 is the result of the retrained LM by the original data blended with AISHELL mono-lingual data. The increase of PPL is still due to mismatching between different corpus. However, the proposed generator with a selection threshold (log probability $\geq$ -1) helps the PPL reduce.

Table 4: *The perplexities of the test set in OCC16-CE80 evaluated by different LMs*

LM	PPL
Original LM	272.34
+AISHELL mono	449.27
+Generated mix	478.16
+Generated mix $logP \geq -2$	451.96
+Generated mix $logP \geq -1$	**383.78**

3.4. Code-switching Speech Recognition

The expanded LMs are utilized in an ASR system. Compared to the host language, we pay more attention on the performance of the guest language speech recognition, since it is much harder to improve. The word error rate (WER) of the test set in OCC16-CE80 is taken as the measurement. During the calculation of WER, each single Chinese character is regarded as a word, which means the WER on the Chinese part is essentially the character error rate (CER).

Table 5 presents the decoding results of the test set in OC16-CE80 through different LMs. The second row shows the contribution of introducing more data to train a LM. However, the improvement mainly happens on the Chinese WER. The last three rows are the results with the newly expanded LMs. The generated data with the selection criterion ($logP \geq -2$) performs the best for the guest language speech recognition.

Table 5: *The speech recognition WERs with different LMs*

LM	Overall	Chi	Eng
Original LM	24.29	22.68	36.12
+AISHELL mono	**23.27**	**21.53**	36.09
+Generated mix	24.04	22.41	36.09
+Generated mix $logP \geq -2$	**23.92**	22.33	**35.57**
+Generated mix $logP \geq -1$	24.05	22.42	36.05

3.5. Examples

Table 6 lists a few generated sentence examples. Some sentences remain the same meaning after being transformed into code-switching, while some others are changed into a different meaning, but still reasonable. The last two rows show two examples that the generated sentences are different while using different models, which demonstrates that GAN helps the generator to generate more natural sentences. People who know both languages can obviously find the naturalness.

Table 6: *Several generated examples by the proposed model*

Original mono	Generated mixed
玩 得(Played) 相 当(Quite)**尽兴**(Happy)	玩得 相当 **high**
如 今(Now) **三 星**(Samsung) 正 在(Is) 进 一 步(Further) 推 动(Promoting) 自 己 的(Its) 虚 拟 现 实(Virtual reality) 业 务(Business)	如今 **Google** 正在 进一步 推动 自己的 虚拟现实 业务
保 监 会(CIRC) 五 招(Five measures) 解 决(Solve) **车 险**(Vehicle insurance) 理 赔(Claim settlement) 难(Difficulty)	保监会 五招 解决 **p2p** 理赔 难
按照(According) 谁的(Whose) **孩 子**(Child) 谁 抱 的(Who hold) 原则 (Principle)	*Bert:* 按照 谁的 孩子 谁抱的 **baby** *Bert-GAN:* 按照 谁的 **baby** 谁抱的 原则
为 (For) 各自的(Each other's) **团 队**(Team) 争 取(Fight) 胜 利(To win)	*Bert:* 为 各自的 团队 **solo** 胜利 *Bert-GAN:* 为 各自的 **team** 争取 胜利

4. Conclusion

In this work, we present a code-switching data generator based on Bert and GAN. Bert is applied to predict the mixed-in foreign words considering the bidirectional context. And GAN assures the generated sentences similar enough to the real ones. Experiments show the evident improvement with our various models. The effectiveness is measured by English word prediction accuracy, the PPL contribution of the LM trained with the generated data as well as the effect for ASR improvement. For the future work, we will consider generating longer sequence of foreign words such as phrase or segment mixed in a sentence.

5. References

[1] C. T. Chang, S. P. Chuang, and H. Y. Lee, "Code-switching sentence generation by generative adversarial networks and its application to data augmentation," *arXiv preprint arXiv:1811.02356*, 2018.

[2] H. P. Shen, C. H. Wu, Y. T. Yang, and C. S. Hsu, "CECOS: a chinese-english code-switching speech database," in *International Conference on Speech Database & Assessments*, 2011, pp. 120–123.

[3] N. T. Vu, D. C. Lyu, J. Weiner, D. Telaar, and H. Li, "A first speech recognition system for mandarin-english code-switch conversational speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 4889–4892.

[4] D.-C. Lyu, T. P. Tan, C. E. Siong, and H. Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia," in *INTERSPEECH*, 2010.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. [Online]. Available: https://github.com/google-research/bert

[6] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues," *arXiv: Computation and Language*, vol. 1, 2016.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[8] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source mandarin speech corpus and A speech recognition baseline," *CoRR*, vol. abs/1709.05522, 2017. [Online]. Available: http://arxiv.org/abs/1709.05522

[9] L. Yanqiang, "Cvte mandarin model," 2016. [Online]. Available: http://arxiv.org/abs/1709.05522

[10] J. Du, X. Na, X. Liu, and H. Bu, "AISHELL-2: transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, Aug 2018.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlł?ek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.

[12] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," pp. 3214–3218, 2015.

[13] A. Stolcke, "Srilm-an extensible language modeling toolkit," vol. 2, 2002, pp. 901–904. [Online]. Available: https://ci.nii.ac.jp/naid/10026674170/en/

[14] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling." in *ICASSP*. IEEE Computer Society, 1995, pp. 181–184.