Code-Switching Language Modeling using Syntax-Aware Multi-Task Learning

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{giwinata, eeandreamad, cwuak}@ust.hk, pascale@ece.ust.hk

Abstract

Lack of text data has been the major issue on code-switching language modeling. In this paper, we introduce multi-task learning based language model which shares syntax representation of languages to leverage linguistic information and tackle the low resource data issue. Our model jointly learns both language modeling and Part-of-Speech tagging on code-switched utterances. In this way, the model is able to identify the location of code-switching points and improves the prediction of next word. Our approach outperforms standard LSTM based language model, with an improvement of 9.7% and 7.4% in perplexity on SEAME Phase I and Phase II dataset respectively.

1 Introduction

Code-switching has received a lot of attention from speech and computational linguistic communities especially on how to automatically recognize text from speech and understand the structure within it. This phenomenon is very common in bilingual and multilingual communities. For decades, linguists studied this phenomenon and found that speakers switch at certain points, not randomly and obeys several constraints which point to the code-switched position in an utterance (Poplack, 1980; Belazi et al., 1994; Myers-Scotton, 1997; Muysken, 2000; Auer and Wei, 2007). These hypotheses have been empirically proven by observing that bilinguals tend to code-switch intra-sententially at certain (morpho)-syntactic boundaries (Poplack, 2015). Belazi et al. (1994) defined the well-known theory that constraints the code-switch between a functional head and its complement is given the strong relationship between the two constituents, which corresponds to a hierarchical structure in terms of Part-of-Speech (POS) tags. Muysken (2000) introduced Matrix-Language Model Framework for an intra-sentential case where the primary language is called Matrix Language and the second one called Embedded Language (Myers-Scotton, 1997). A language island was then introduced which is a constituent composed entirely of the language morphemes. From the Matrix-Language Frame Model, both matrix language (ML) island and embedded language (EL) islands are well-formed in their grammars and the EL islands are constrained under ML grammar (Namba, 2004). (Fairchild and Van Hell, 2017) studied determiner–noun switches in Spanish–English bilinguals .

Code-switching can be classified into two categories: intra-sentential and inter-sentential switches (Poplack, 1980). Intra-sentential switch defines a shift from one language to another language within an utterance. Inter-sentential switch refers to the change between two languages in a single discourse, where the switching occurs after a sentence in the first language has been completed and the next sentence starts with a new language. The example of the intra-sentential switch is shown in (1), and the inter-sentential switch is shown in (2).

(1) 我 要 去 check.

(I want to go) check.

(2) 我 不 懂 要 怎么 讲 一 个 小时 seriously I didn't have so much things to say

(I don't understand how to speak for an hour) seriously I didn't have so much things to say

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 62–67
Melbourne, Australia, July 19, 2018.

Language modeling using only word lexicons is not adequate to learn the complexity of code-switching patterns, especially in a low resource setting. Learning at the same time syntactic features such as POS tag and language identifier allows to have a shared grammatical information that constraint the next word prediction. Due to this reason, we propose a multi-task learning framework for code-switching language modeling task which is able to leverage syntactic features such as language and POS tag.

The main contribution of this paper is two-fold. First, multi-task learning model is proposed to jointly learn language modeling task and POS sequence tagging task on code-switched utterances. Second, we incorporate language information into POS tags to create bilingual tags - it distinguishes tags between Chinese and English. The POS tag features are shared towards the language model and enrich the features to better learn where to switch. From our experiments result, we found that our method improves the perplexity on SEAME Phase I and Phase II dataset (Nanyang Technological University, 2015).

2 Related Work

The earliest language modeling research on code-switching data was applying linguistic theories on computational modelings such as Inversion Constraints and Functional Head Constraints on Chinese-English code-switching data (Li and Fung, 2012; Ying and Fung, 2014). Vu et al. (2012) built a bilingual language model which is trained by interpolating two monolingual language models with statistical machine translation (SMT) based text generation to generate artificial code-switching text. Adel et al. (2013a,b) introduced a class-based method using RNNLM for computing the posterior probability and added POS tags in the input. Adel et al. (2015) explored the combination of brown word clusters, open class words, and clusters of open class word embeddings as hand-crafted features for improving the factored language model. In addition, Dyer et al. (2016) proposed a generative language modeling with explicit phrase structure. A method of tying input and output embedding helped to reduce the number of parameters in language model and improved the perplexity (Press and Wolf, 2017).

Learning multiple NLP tasks using multi-task learning have been recently used in many domains (Collobert et al., 2011; Luong et al., 2016; Hashimoto et al., 2016). They presented a joint many-task model to handle multiple NLP tasks and share parameters with growing depth in a single end-to-end model. A work by Aguilar et al. (2017) showed the potential of combining POS tagging with Named-Entity Recognition task.

3 Methodology

This section shows how to build the features and how to train our multi-task learning language model. Multi-task learning consists of two NLP tasks: Language modeling and POS sequence tagging.

3.1 Feature Representation

In the model, word lexicons and syntactic features are used as input.

Word Lexicons: Sentences are encoded as 1-hot vectors and our vocabulary is built from training data.

Syntactic Features: For each *language island*, phrase within the same language, we extract POS Tags iteratively using Chinese and English Penn Tree Bank Parser (Tseng et al., 2005; Toutanova et al., 2003). There are 31 English POS Tags and 34 Chinese POS Tags. Chinese words are distinguishable from English words since they have different encoding. We add language information in the POS tag label to discriminate POS tag between two languages.

3.2 Model Description

Figure 1 illustrates our multi-task learning extension to recurrent language model. In this multi-task learning setting, the tasks are language modeling and POS tagging. The POS tagging task shares the POS tag vector and the hidden states to LM task, but it does not receive any information from the other loss. Let w_t be the word lexicon in the document and p_t be the POS tag of the corresponding w_t at index t. They are mapped into embedding matrices to get their d-dimensional vector representations x_t^w and x_t^p. The input embedding weights are tied with the output weights. We concatenate x_t^w and x_t^p as the input of LSTM_{lm}. The information from the POS tag sequence is shared to the language model through this step.

$$u_t = \text{LSTM}_{lm}(x_t^w \oplus x_t^p, u_{t-1})$$

$$v_t = \text{LSTM}_{pt}(x_t^p, v_{t-1})$$

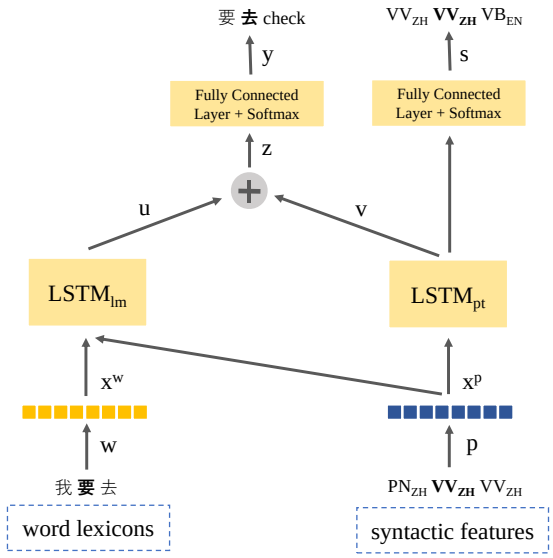

Figure 1: Multi-Task Learning Framework

where $\oplus$ denotes the concatenation operator, u_t and v_t are the final hidden states of LSTM$_{lm}$ and LSTM$_{pt}$ respectively. u_t and v_t, the hidden states from both LSTMs are summed before predicting the next word.

$$z_t = u_t + v_t$$

$$y_t = \frac{e^{z_t}}{\sum_{j=1}^{T} e^{z_j}}, \text{where } j = 1, .., T$$

The word distribution of the next word y_t is normalized using softmax function. The model uses cross-entropy losses as error functions $\mathcal{L}_{lm}$ and $\mathcal{L}_{pt}$ for language modeling task and POS tagging task respectively. We optimize the multi-objective losses using the Back Propagation algorithm and we perform a weighted linear sum of the losses for each individual task.

$$\mathcal{L}_{total} = p\mathcal{L}_{lm} + (1-p)\mathcal{L}_{pt}$$

where p is the weight of the loss in the training.

3.3 Experimental Setup

In this section, we present the experimental setting for this task

Corpus: SEAME (South East Asia Mandarin-English), a conversational Mandarin-English code-switching speech corpus consists of spontaneously spoken interviews and conversations (Nanyang Technological University, 2015). Our dataset (LDC2015S04) is the most updated version of the Linguistic Data Consortium (LDC) database. However, the statistics are not identical to Lyu et al. (2010). The corpus consists of two phases. In Phase I, only selected audio segments were transcribed. In Phase II, most of the audio segments were transcribed. According to the authors, it was not possible to restore the original dataset. The authors only used Phase I corpus. Few speaker ids are not in the speaker list provided by the authors Lyu et al. (2010). Therefore as a workaround, we added these ids to the train set. As our future reference, the recording lists are included in the supplementary material.

Table 1: Data Statistics in SEAME Phase I

	Train set	**Dev set**	**Test set**
# Speakers	139	8	8
# Utterances	45,916	1,938	1,228
# Tokens	762K	31K	17K
Avg. segments length	3.67	3.68	3.18
Avg. switches	3.60	3.47	3.67

Table 2: Data Statistics in SEAME Phase II

	Train set	**Dev set**	**Test set**
# Speakers	138	8	8
# Utterances	78,815	4,764	3,933
# Tokens	1.2M	65K	60K
Avg. segment length	4.21	3.59	3.99
Avg. switches	2.94	3.12	3.07

Table 3: Code-Switching Trigger Words in SEAME Phase II

POS Tag	**Freq**	**POS Tag**	**Freq**
VV$_{ZH}$	107,133	NN$_{EN}$	31,031
AD$_{ZH}$	97,681	RB$_{EN}$	12,498
PN$_{ZH}$	92,117	NNP$_{EN}$	11,734
NN$_{ZH}$	45,088	JJ$_{EN}$	5,040
VA$_{ZH}$	27,442	IN$_{EN}$	4,801
CD$_{ZH}$	20,158	VB$_{EN}$	4,703

Preprocessing: First, we tokenized English and Chinese word using Stanford NLP toolkit (Manning et al., 2014). Second, all hesitations and punctuations were removed except apostrophe, for examples: "let's" and "it's". Table 1 and Table 2 show the statistics of SEAME Phase I and II corpora. Table 3 shows the most common trigger POS tag for Phase II corpus.

Training: The baseline model was trained using RNNLM (Mikolov et al., 2011)[1]. Then, we trained our LSTM models with different hidden sizes [200, 500]. All LSTMs have 2 layers and unrolled for 35 steps. The embedding size is equal to the LSTM hidden size. A dropout regularization (Srivastava et al., 2014) was applied to the word embedding vector and POS tag embedding vector, and to the recurrent output (Gal and Ghahramani, 2016) with values between [0.2, 0.4]. We used a batch size of 20 in the training. EOS tag was used to separate every sentence. We chose Stochastic Gradient Descent and started with a learning rate of 20 and if there was no improvement during the evaluation, we reduced the learning rate by a factor of 0.75. The gradient was clipped to a maximum of 0.25. For the multi-task learning, we used different loss weights hyper-parameters p in the range of [0.25, 0.5, 0.75]. We tuned our model with the development set and we evaluated our best model using the test set, taking perplexity as the final evaluation metric. Where the latter was calculated by taking the exponential of the error in the negative log-form.

$$\text{PPL}(w) = e^{\mathcal{L}_{total}}$$

4 Results

Table 4 and Table 5 show the results of multi-task learning with different values of the hyper-parameter p. We observe that the multi-task model with $p = 0.25$ achieved the best performance. We compare our multi-task learning model against RNNLM and LSTM baselines. The baselines correspond to recurrent neural networks that are trained with word lexicons. Table 6 and Table 7 present the overall results from different models. The multi-task model performs better than LSTM baseline by 9.7% perplexity in Phase I and 7.4% perplexity in Phase II. The performance of our model in Phase II is also better than the RNNLM (8.9%) and far better than the one presented in Adel et al. (2013b) in Phase I.

Moreover, the results show that adding shared POS tag representation to LSTM$_{lm}$ does not hurt the performance of the language modeling task. This implies that the syntactic information helps the model to better predict the next word in the sequence. To further verify this hypothesis, we conduct two analysis by visualizing our prediction examples in Figure 2:

a) Measure the improvement of the target word's log probability by multi-task model compared to standard LSTM model. This is computed by calculating the log probability difference between two models. According to Figure 2, in most of the

[1]downloaded from Mikolov's website http://www.fit.vutbr.cz/ imikolov/rnnlm/

Table 4: Multi-task results with different weighted loss hyper-parameter in Phase I

Hidden size	p	PPL Dev	PPL Test
200	**0.25**	**180.90**	**178.18**
	0.5	182.6	178.75
	0.75	**180.90**	**178.18**
500	**0.25**	**173.55**	**174.96**
	0.5	175.23	173.89
	0.75	185.83	178.49

Table 5: Multi-task results with different weighted loss hyper-parameter in Phase II

Hidden size	p	PPL Dev	PPL Test
200	**0.25**	**149.68**	**149.84**
	0.5	150.92	152.38
	0.75	150.32	151.22
500	**0.25**	**141.86**	**141.71**
	0.5	144.18	144.27
	0.75	145.08	144.85

Table 6: Results in Phase I

Model	PPL Dev	PPL Test
RNNLM (Adel et al., 2013a)	246.60	287.88
(Adel et al., 2015)	238.86	245.40
FI + OF (Adel et al., 2013a)	219.85	239.21
FLM (Adel et al., 2013b)	177.79	192.08
LSTM	190.33	185.91
+ syntactic features	178.51	176.57
Multi-task	**173.55**	**174.96**

Table 7: Results in Phase II

Model	PPL Dev	PPL Test
RNNLM	178.35	171.27
LSTM	150.65	153.06
+ syntactic features	147.44	148.38
Multi-task	**141.86**	**141.71**

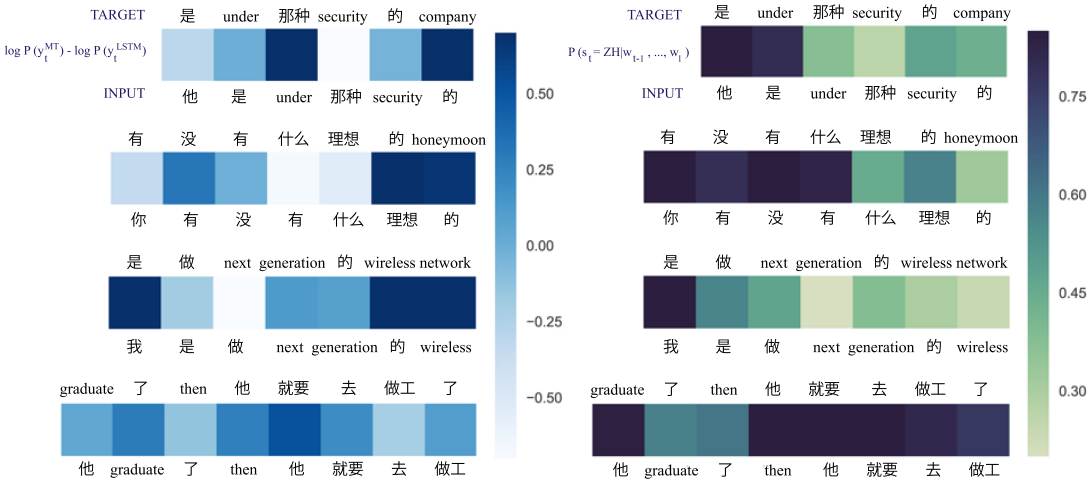

Figure 2: Prediction examples in Phase II. **Left:** Each square shows the target word's log probability improvement by multi-task model compared to LSTM model (Darker color is better). **Right:** Each square shows the probability of the next POS tag is Chinese (Darker color represents higher probability)

cases, the multi-task model improves the prediction of the monolingual segments and particularly in code-switching points such as "under", "security", "generation", "then", "graduate", "他", and "的". It also shows that the multi-task model is more precise in learning where to switch language. On the other hand, Table 3 shows the relative frequency of the trigger POS tag. The word "then" belong to RB_{EN}, which is one of the most common trigger words in the list. Furthermore, the target word prediction is significantly improved in most of the trigger words.

b) Report the probability that the next produced POS tag is Chinese. It is shown that words "then", "security", "了", "那种", "做", and "的" tend to switch the language context within the utterance. However, it is very hard to predict all the cases correctly. This is may due to the fact that without any switching, the model still creates a correct sentence.

5 Conclusion

In this paper, we propose a multi-task learning approach for code-switched language modeling. The multi-task learning models achieve the best performance and outperform LSTM baseline with 9.7% and 7.4% improvement in perplexity for Phase I and Phase II SEAME corpus respectively. This implies that by training two different NLP tasks together the model can correctly learn the correlation between them. Indeed, the syntactic information helps the model to be aware of code-switching points and it improves the performance over the language model. Finally, we conclude that multi-task learning has good potential on code-switching language modeling research and there are still rooms for improvements, especially by adding more language pairs and corpora.

Acknowledgments

This work is partially funded by ITS/319/16FP of the Innovation Technology Commission, HKUST 16214415 & 16248016 of Hong Kong Research Grants Council, and RDC 1718050-0 of EMOS.AI.

References

Heike Adel, Ngoc Thang Vu, Katrin Kirchhoff, Dominic Telaar, and Tanja Schultz. 2015. Syntactic and semantic features for code-switching factored language models. *IEEE Transactions on Audio, Speech, and Language Processing*, 23(3):431–440.

Heike Adel, Ngoc Thang Vu, Franziska Kraus, Tim Schlippe, Haizhou Li, and Tanja Schultz. 2013a. Recurrent neural network language modeling for code switching conversational speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8411–8415. IEEE.

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. 2013b. 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)*, volume 2, pages 206–211.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153.

Peter Auer and Li Wei. 2007. *Handbook of multilingualism and multilingual communication*, volume 5. Walter de Gruyter.

Hedi M Belazi, Edward J Rubin, and Almeida Jacqueline Toribio. 1994. Code switching and x-bar theory: The functional head constraint. *Linguistic inquiry*, pages 221–237.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of NAACL-HLT*, pages 199–209.

Sarah Fairchild and Janet G Van Hell. 2017. Determiner-noun code-switching in spanish heritage speakers. *Bilingualism: Language and Cognition*, 20(1):150–161.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Ying Li and Pascale Fung. 2012. Code-switch language model with inversion constraints for mixed language speech recognition. *Proceedings of COLING 2012*, pages 1671–1680.

Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning.

Dau-Cheng Lyu, Tien-Ping Tan, Eng-Siong Chng, and Haizhou Li. 2010. An analysis of a mandarin-english code-switching speech corpus: Seame. *Age*, 21:25–8.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. 2011. Rnnlm-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

Pieter Muysken. 2000. *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Kazuhiko Namba. 2004. An overview of myers-scotton's matrix language frame model. *Senri International School (SIS) Educational Research Bulletin*, 9:1–10.

Universiti Sains Malaysia Nanyang Technological University. 2015. Mandarin-english code-switching in south-east asia ldc2015s04. web download. philadelphia: Linguistic data consortium.

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espanol: toward a typology of code-switching1. *Linguistics*, 18(7-8):581–618.

Shana Poplack. 2015. Code-switching (linguistic). *International encyclopedia of the social and behavioral sciences*, pages 918–925.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 157–163.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. 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*, pages 173–180. Association for Computational Linguistics.

Huihsin Tseng, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. In *Proceedings of the fourth SIGHAN workshop on Chinese language processing*.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A first speech recognition system for mandarin-english code-switch conversational speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4889–4892. IEEE.

LI Ying and Pascale Fung. 2014. 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)*, pages 907–916.