Modeling Code-Switch Languages Using Bilingual Parallel Corpus

Grandee Lee
National University of Singapore
grandee.lee@u.nus.edu

Haizhou Li
National University of Singapore
Kriston AI Lab, China
haizhou.li@nus.edu.sg

Abstract

Language modeling is the technique to estimate the probability of a sequence of words. A bilingual language model is expected to model the sequential dependency for words across languages, which is difficult due to the inherent lack of suitable training data as well as diverse syntactic structure across languages. We propose a bilingual attention language model (BALM) that simultaneously performs language modeling objective with a quasi-translation objective to model both the monolingual as well as the cross-lingual sequential dependency. The attention mechanism learns the bilingual context from a parallel corpus. BALM achieves state-of-the-art performance on the SEAME code-switch database by reducing the perplexity of 20.5% over the best-reported result. We also apply BALM in bilingual lexicon induction, and language normalization tasks to validate the idea.

1 Introduction

Monolingual language modeling has enabled many NLP tasks (Devlin et al., 2019; Dai et al., 2019; Radford et al., 2019). However, the bilingual language model was not well studied. The recent advances in cross-lingual word embedding (CLWE) (Ruder et al., 2019), which projects word of different languages into a shared embedding space for cross-lingual representations (Devlin et al., 2019; Lample and Conneau, 2019), make possible some cross-lingual applications. Unfortunately, they are not optimized to model the sequential dependency for word prediction in a bilingual text.

In this paper, we would like to propose a bilingual language model that can learn word embeddings to represent the equivalent words between two languages, and more importantly, to model the sequential dependency for words across languages at the same time. For instance, the model should be able to predict the appropriate word to fill in the blank, given the bilingual context:

昨 晚 的 movie (____). [1]

The above sentence is an example of code-switching or code-mixing (henceforth, CS), where a bilingual speaker alternates words of two or more languages within a single sentence. The switches could happen at sentence boundaries or word boundaries and for some agglutinative languages even within words. Code-switching is common in both spoken and, to some extent, written communication in many multilingual societies, such as Southeast Asia. Hence, the study of code-switch in linguistics and bilingual language modeling is becoming imperative, especially for NLP tasks such as code-switching automatic speech recognition (ASR) (Adel et al., 2013b; Li and Fung, 2013; Lee et al., 2019), cross-lingual language normalization.

It is tempting to think that, given enough of code-switching text data, bilingual language modeling could be approached in the same way as that for monolingual data. The main challenge is the lack of such CS data. We note that CS mainly occurs in the spoken form, and CS does not occur in every sentence. Therefore, collecting enough pure CS data is just not practical or even feasible (Lee et al., 2017; Pratapa et al., 2018).

The problem is further exacerbated by the syntactic constraints of the two diverse languages, such as Chinese and English. Three dominant theories seek to explain the syntactic formation of CS sentences. They are the Matrix Language Frame theory (Myers-Scotton, 1997), which shows that individual monolingual sentences will conform to the grammar of the matrix language. The Equivalence Constraint theory (Poplack, 2000; Sankoff, 1998), which further constrains the intra-sentential CS points to the syntactic boundaries shared by both languages, and the Functional Head Constraint theory (Di Sciullo et al., 1986; Belazi et al., 1994) that imposes constraints on the functional head and its

[1] English: The movie last night (____)

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 860–870
July 5 - 10, 2020.

complements.

A bilingual language model should be able to predict a word, either in the matrix language or otherwise, given either a bilingual or monolingual context. Therefore, it has to respect the respective monolingual word sequential dependency, the cross-lingual word correspondence, as well as the switching rules between languages. The contributions of this paper are summarized as follows:

1. We propose an attention-based, auto-regressive model, bilingual attention language model (BALM), that not only learns the latent alignment from a parallel corpus for cross-lingual word embedding but also captures the word sequential dependency.

2. Adhering to the Matrix Language Frame theory (Myers-Scotton, 1997) and Equivalence Constraint theory (Poplack, 2000; Sankoff, 1998), we implement an objective function by jointly optimizing the cross-entropy loss as the monolingual constraint and the quasi-translation loss as the cross-lingual constraint.

3. We show that BALM can learn from bilingual parallel data without the need for CS data. When adapted on CS data, it outperforms the best reported result on the SEAME dataset in the perplexity test. We also successfully apply BALM in bilingual lexicon induction, and language normalization tasks to validate the idea.

2 Related Work

Several prior studies related to bilingual language modeling are the inspiration for this work.

Cross-lingual correspondence: Several studies are focused on projecting words of different languages onto the common embedding space to establish cross-lingual correspondence. One idea is to train a model using bilingual information from corpora aligned at the sentence level (Zou et al., 2013; Hermann and Blunsom, 2014; Luong et al., 2015) and document level (Vulic and Moens, 2016; Levy et al., 2017). Another is to exploit the isomorphic structure (Conneau et al., 2017; Artetxe et al., 2018), dictionary (Mikolov et al., 2013; Faruqui and Dyer, 2014; Huang et al., 2015; Zhang et al., 2016), shared cognate, vocab (Hauer et al., 2017; Smith et al., 2017), numeral (Artetxe et al., 2017) through ad-hoc projection.

As the above approaches do not explicitly consider the sequential dependency of words, the embedding doesn't encode the word ordering information. The multilingual techniques, such as M-BERT (Devlin et al., 2019) and XLM (Lample and Conneau, 2019), do not explicitly model the syntactic constraints for CS as formulated in the Equivalence Constraint theory, thus not making full use of the information which could potentially improve their performance.

Code-switching modeling: Another school of thoughts is to extend the monolingual language modeling technique to accommodate code-switch content. Adel et al. (2013b, 2014) use factored language models and recurrent neural network (RNN) language model to improve the bilingual language model for CS ASR rescoring. They include additional linguistic information such as Part-of-Speech, language identifier to improve model generalization. Inversion constraints (Li and Fung, 2013) and Functional Head constraints (Li and Fung, 2014) are also used in language models for the ASR decoding process. Lee and Li (2019) use cross-lingual embedding to tie the input and output layer, and incorporate classes in the RNN language model. While these models are effective, they rely on the availability of CS training data. Therefore, they are not easily scalable. To address this, we propose a way to make use of the existing abundant parallel corpora. The method will be explained in Section 3.3.

Code-switching text generation: Closer to our line of research, Pratapa et al. (2018) propose to use synthetic data following the Equivalence Constraint theory, while Lee et al. (2019) apply the Matrix Language Frame theory. In their works, a parser or an aligner is required to process the parallel corpus, which is followed by the standard monolingual language modeling process. Such techniques suffer from inaccurate alignment or parsing errors. These errors will be carried forward when training the language model. More recently, Winata et al. (2019) propose a technique to generate neural-based synthetic data using parallel sentences, in which a Point-Gen network is used to synthesize CS data without external aligner or parser. In this paper, we propose to learn the bilingual context and the CS language model jointly by attending to the parallel sentences directly without the need for an external aligner, parser or explicitly generating the synthetic data.

3 Bilingual Attention Language Model

Next, we discuss the motivation and the theoretical formulation of the proposed Bilingual Attention Language Model (BALM). In a bilingual text, we could encounter a sequence of word, $\mathbf{w} = w_1^{l_1}, w_2^{l_2}, \dots w_t^{l_2}, \dots, w_T^{l_1}$, code mixed between languages l_1 and l_2. However, such code mixed training data are not easily available. Let us assume that only parallel corpus at sentence level between l_1 and l_2 languages is available to us.

Assuming the validity of the Matrix Frame theory, and Equivalence Constraint theory, the above code-switch sentence, $\mathbf{w}$, can be constructed from two parallel sentences, $\mathbf{w}^{l_1} = w_1^{l_1}, w_2^{l_1}, \dots, w_{T_1}^{l_1}, \mathbf{w}^{l_2} = w_1^{l_2}, w_2^{l_2}, \dots, w_{T_2}^{l_2}$. For a monolingual case, the language model maximizes the log-likelihood of $p(w_t|\mathbf{w}_{<t})$ which effectively captures the monolingual word sequential dependency. For a CS case, we would like to maximize $p(w_t|\mathbf{w}_{<t})$, whereby the bilingual context, $\mathbf{w}_{<t}$, is non-existent during training. In the subsequent section, we will explain the idea to encode the bilingual context using an attention mechanism.

3.1 Background

A bilingual language model has to be built on a common word representation. The continuous space word embedding is an effective solution. We first draw some principled insights from the cross-lingual word embedding (CLWE) study, which motivates this work.

Building on the idea of CLWE, we refer to the general form of the loss function, J , summarized by Ruder et al. (2019) as follows,

$$J = \mathcal{L}(\mathbf{X}^{l_1}) + \mathcal{L}(\mathbf{X}^{l_2}) + \Omega(\mathbf{X}^{l_1}, \mathbf{X}^{l_2}, \mathbf{A}). \quad (1)$$

The monolingual language constraint $\mathcal{L}$, which could be implemented with negative sampling, preserves the monolingual integrity. Importantly, there has to be a cross-lingual constraint, which could be the mean squared error (MSE) between the l_2 embedding space $\mathbf{X}^{l_2} = \{x_i^{l_2}\}$, and the transformed l_1 embedding space, $\mathbf{X}^{l_1} = \{x_i^{l_1}\}$. We use x_i to denote the embedding of a word w_i, which is also referred to as a token. The vocabulary size is v. The cross-lingual language constraint Ω maps the two monolingual embeddings into a common space using the transformation matrix $\mathbf{A}$,

$$\Omega_{MSE} = \sum_{i=1}^{v} ||\mathbf{A}x_i^{l_1} - x_i^{l_2}||. \quad (2)$$

The CLWE network can also be jointly learned (Luong et al., 2015) with the alignment information as the regularization loss, Ω. While CLWE lays the foundation for many cross-lingual applications, it is not designed to model word sequential dependency.

3.2 Bilingual Objective

We draw inspiration from the CLWE loss function and extend the objective function to the modeling of word sequential dependency while preserving its general form.

The monolingual objective, $\mathcal{L}(\mathbf{X}^l)$ as formulated in Equation 3, is set to be the cross entropy loss between the target distribution, y^l and the predicted distribution $\log p(w_t^l|\mathbf{w}_{<t}^l)$, for the respective language, which preserves the monolingual word sequential order.

$$\mathcal{L}(\mathbf{X}^l) = y^l \log p(w_t^l|\mathbf{w}_{<t}^l),\ l \in \{l_1, l_2\} \quad (3)$$

This allows the bilingual language model to adhere to the monolingual syntactic rules of the Matrix Language Frame and the Equivalent Constraint theory during word prediction, that the dominant language still abide by its own syntactic principle.

We also define a quasi-translation loss, Ω, that optimizes the model to learn the correspondence of tokens between languages as well as the dependencies between the current token in l_1 and the preceding context in l_2. The quasi-translation loss can be interpreted as satisfying the requirement of the code-switching principle as described by the two theories.

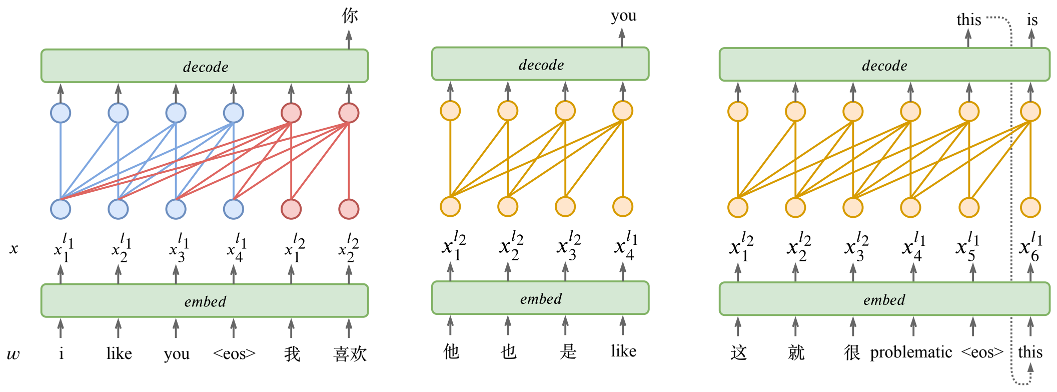
$$\Omega_{l_1 l_2 \rightarrow l_1} = y^{l_1} \log p(w_t^{l_1}|\mathbf{w}^{l_2}, \mathbf{w}_{<t}^{l_1}) \quad (4)$$

Equation 4 is the quasi-translation loss, $\Omega_{l_1 l_2 \rightarrow l_1}$, when predicting a word in l_1 given a bilingual context. Similarly, we have $\Omega_{l_1 l_2 \rightarrow l_2}$ to predict a word in l_2.

3.3 Bilingual Attention

Motivated by the self-attention model (Vaswani et al., 2017), we hypothesize that an autoregressive translation-cum-language modeling objective could leverage on parallel sentences to learn the bilingual context.

To start with, let us consider a monolingual case that deals with l_1. We define a transformer language model, f, using a causal mask (Radford et al., 2019), which can be further broken down


你
decode
embed
x
$x_1^{l_1}$ $x_2^{l_1}$ $x_3^{l_1}$ $x_4^{l_1}$ $x_1^{l_2}$ $x_2^{l_2}$
w
i
like
you
<eos>
我
喜欢
you
decode
embed
$x_1^{l_2}$ $x_2^{l_2}$ $x_3^{l_2}$ $x_4^{l_1}$
他
也
是
like
this
is
decode
embed
$x_1^{l_2}$ $x_2^{l_2}$ $x_3^{l_2}$ $x_4^{l_1}$ $x_5^{l_1}$ $x_6^{l_1}$
这
就
很
problematic
<eos>
this

(a) Train on $l_1l_2 \rightarrow l_2$ data.

(b) Estimate $p(\text{“}you\text{”}|\mathbf{w}_{<5})$

(c) Normalize $l_1l_2 \rightarrow l_1$.

Figure 1: (a) Trained on a parallel sentence pair l_1l_2, “i like you” and “我喜欢你” , BALM learns to predict the next l_2 word, “你”, given its context $\mathbf{x}^{l_2}_{<3}$, “我喜欢”, and its whole sentence translation $\mathbf{x}^{l_1}_{<5}$, “i like you”. (b) During perplexity evaluation, BALM estimates the probability of $p(\text{“}you\text{”}|\mathbf{w}_{<5})$, given a bilingual context $\mathbf{w}_{<5}$, “他也是like”. (c) Normalizing a l_1l_2 code-switch sentence to l_1 with BALM by generating the l_1 sentence sequentially in an auto-regressive manner. $x = embed(w)$ is the cross-lingual word embedding layer and the transpose of the $embed$ weight is used for the output projection layer to $decode$ the word distribution.

into individual layer n in a total of N layers,

$$f_1^n = Attention(\mathbf{x}^{l_1}_{<t}))$$
$$f_2^n = FeedForward(f_1^n)$$
$$f^n = f_2^n \circ f_1^n$$

The model will take in the embedding, $x_t^{l_1} = embed(w_t^{l_1})$ of each word, $w_t^{l_1}$, in l_1 at the first layer, f_1^1, and the output will encode the contextual information that is a weighted sum of its preceding context, $f^1 = f_2^1(Attention(\mathbf{x}^{l_1}_{<t}))$. In this way, the output of the last layer f_2^N contains the information, that is necessary for decoding $p(w_t^{l_1}|\mathbf{w}^{l_1}_{<t})$. This process is carried out on the monolingual side of the parallel data respectively for l_1 and l_2 to minimize the loss function in Equation 3.

Extending the context of l_1 to include words in l_2, we enable the model to learn from a bilingual context, as shown in Figure 1a. The question is how to find the appropriate context in both l_1 and l_2 to predict a word in l_2. The attention mechanism with the quasi-translation loss provides a solution. Figure 1a is an illustration for $l_1l_2 \rightarrow l_2$ training case.

At the last layer, the encoded output for the time step t in l_2 will be, $f_2^N(Attention(\mathbf{x}^{l_1}, \mathbf{x}^{l_2}_{\leq t}))$. It is important to note that the model architecture allows learnable alignment between current word x_t with its preceding context in its own language l_2 as well as the whole sentence translation $\mathbf{x}^{l_1}$ in l_1. The use of preceding context can be seen as an auto-regressive process over the words in a sentence.

As the predicted word always follows its preceding context sequentially, the word order in the matrix language matters in BALM. However, the attention mechanism does not attempt to distinguish word order within the encoded context, which is a weighted sum of the bilingual context (see discussions in Section 3.5). This can be observed in the quasi-translation loss, as formulated in Equation 4.

3.4 Training and Inference

During training, we use the two sides of the parallel corpus independently as two monolingual corpora and both sides together as the bilingual constraint. When presented with monolingual text in l_1 or l_2, the network learns to attend to the words in either l_1 or l_2 using a causal mask for monolingual word prediction. When presented with l_1l_2 parallel sentences, and predicting a word in l_1 or l_2, the network learns to attend to the bilingual context for word prediction.

To summarize, given a parallel corpus, BALM is trained with 4 $input \rightarrow output$ pairs, $l_1 \rightarrow l_1$, $l_2 \rightarrow l_2$, $l_1l_2 \rightarrow l_1$, and $l_1l_2 \rightarrow l_2$. The bilingual attention in theory allows BALM to take any of l_1, l_2 or l_1l_2 as input, and generate any of l_1, l_2 or l_1l_2 as output in 6 possible combinations. $l_1l_2 \rightarrow l_1, l_2$ represents the code-switch language modeling task of our interest. For brevity, we only illustrate the case of $l_1l_2 \rightarrow l_2$ in Figure 1a.

At run time inference, we do not have the two parallel sentences, but rather a code-switch sentence that consists of a mixture of words $\mathbf{w}_{<t}$ from

the two languages, as in Figure 1b. To predict $p(w_t^{l_2}|\mathbf{w}_{<t})$ for a code-switch sentence at run time, we assume that the model would have encountered some variants of the bilingual context through $(Attention(\mathbf{x}^{l_1}, \mathbf{x}^{l_2}_{<t}))$. In this way, the model can estimate the run time probability according to the similarity between the encoding of the code-switch sequence, $\mathbf{w}_{<t}$, and the learned bilingual representation. The attention-based alignment is expected to find the appropriate bilingual context that was trained under the objective function to maximize $p(w_t^{l_2}|\mathbf{w}^{l_1}, \mathbf{w}^{l_2}_{<t})$.

3.5 Positional Embedding

In stark contrast to the masked language model (MLM), which employs positional embedding on top of its sequence ordering invariant setup, BALM does not use positional embedding. We argue that under the auto-regressive objective, positional embedding is not necessary.

In BALM, the amount of information in an auto-regressive setup is strictly increasing. Taking one of its intermediate layers as an example, the hidden representation for the current token h_t is the weighted sum of the previous tokens, and the weights are computed through the learned query and key matrix, $\mathbf{A}_Q, \mathbf{A}_K$.

$$\begin{aligned} h_t &= a_{1,t}x_1 + a_{2,t}x_2 + \cdots + a_{t,t}x_t \\ a_{n,m} &= \mathbf{A}_K x_n \cdot \mathbf{A}_Q x_m \end{aligned}$$

In comparison with a RNN layer, whereby the hidden state is a gated sum of the previous hidden states, i.e. $h_t = tanh(\mathbf{W}_h h_{t-1} + \mathbf{W}_x x_t)$, the difference is that the weight matrix, $\mathbf{W}_h$, for RNN is applied on the gated sum, h_{t-1}, at each time step while the weight for the attention model, $a_{n,m}$, is a similarity comparison of the current token's query with the previous tokens' keys.

The two networks are similar in the sense that they both compute the weights and incorporate the past information. They only differ in their implementation. We argue that the sequential information is already included in the attention model under an auto-regressive setup. Thus the positional encoding is not necessary. This is corroborated by Irie et al. (2019), which shows that the removal of positional encoding slightly improves the language model performance. By dropping the positional embedding, we can mix the bilingual context, as discussed in Section 3.3.

4 Experiments

4.1 Datasets

We evaluate the language models on the text transcripts of the South East Asia Mandarin-English (SEAME) corpus (LDC2015S04) (Lee et al., 2017), a well-documented database for spontaneous conversational speech code-switching between Chinese Mandarin (ZH) and English (EN). A large number of CS studies were reported on SEAME.

We adopt a slightly different setup as we focus on how BALM is able to learn from a parallel corpus alone without the need of CS training data. We use SEAME data mainly for adaptation and evaluation. We split the SEAME Phase II text transcripts equally into three portions, labeled as *Adapt*, *Valid* and *Test* respectively in Table 1. Such split also ensures that the individual component within the *Test* data, *e.g. Test EN*, is of sufficient size.

Additionally, we also split the dataset following approximately the same proportion as in the previous works (Winata et al., 2019; Lee et al., 2019) for a fair benchmarking, labeled as *Train*, *Dev*, and *Eval* respectively. We use a random split of $1.1M/60.8K/60.3K$ for the number of tokens in *Train*/*Dev*/*Eval* as compared to $1.2M/65K/60K$ in the previous works.

We use a bilingual parallel corpus from Ted and OpenSubtitle (Tiedemann, 2012; Lison and Tiedemann, 2016) for BALM training because they are text transcripts of spontaneous speech similar to SEAME. The English text is tokenized using NLTK tokenizer (Bird et al., 2009) while the Chinese text is tokenized using Stanford Word Segmenter (Chang et al., 2008). We also develop a test set of 200 sentences for language normalization experiments, labeled as SEAME *Norm*.

4.2 Experimental Setup

We conduct a series of experiments, namely BALM, Synthetic CS, CS-Only, and Mono, using the same BALM network architecture to evaluate different modeling strategies.

During training, we construct a 50K vocabulary consisting of the most frequent words in the combined SEAME and parallel dataset, of which there are $17.7K$ and $32.3K$ unique Chinese and English words, respectively. Only for the benchmarking in Table 3, we use the SEAME vocabulary, a subset of the 50K vocabulary, for the perplexity evaluation to meaningfully compare the perplexity with the prior work on SEAME corpus.

Dataset	#Lines	#Tokens	#Vocab	SPF
Ted+OpenSubtitle*	3.6M	234.4M	366.7K	0
SEAME *Adapt*	30.9K	398.4K	14.1K	0.17
SEAME *Valid*	30.9K	399.1K	14.1K	0.17
SEAME *Test*	30.9K	400.8K	14.0K	0.17
-*Test CS*	18.9K	284.8K	11.9K	0.23
-*Test EN*	5.8K	58.5K	4.3K	0
-*Test ZH*	6.2K	57.5K	3.3K	0
SEAME *Norm*	200	1.8K	650	0.26
SEAME *Train*	82.3K	1.1M	20.7K	0.17
SEAME *Dev*	4.6K	60.8K	5.7K	0.16
SEAME *Eval*	4.6K	60.3K	5.9K	0.17

Table 1: Test *CS*, Test *EN*, and Test *ZH* represent code-switching, pure English, and pure Chinese partition of SEAME *Test* respectively. SPF refers to Switching Point Fraction Pratapa et al. (2018). Ted+OpenSubtitle* is a bilingual parallel corpus.

Unless otherwise stated, we train for 60 epochs with 100K lines per epoch and adapt for 17 epochs with the full *Adapt* dataset. We use Adam optimizer (Kingma and Ba, 2014) for all the experiments.

BALM The attention mechanism follows largely the implementation of GPT (Radford et al., 2019), with 384-dimension hidden states, 12 layers and 12 heads. While Dai et al. (2019) reports state-of-the-art results using the recurrence mechanism within the attention, we exclude this in our experiment for two reasons. Firstly, the context beyond the given parallel sentence is not meaningful after shuffling the sentences. Furthermore, attending target sequence to context beyond the source sequence may introduce noise and depart from the theoretical motivation of the experiment. Secondly, for many downstream tasks like ASR, the decoding remains at the utterance level.

We first train the BALM on the parallel corpus as described in Section 3.4. The trained network is then adapted with SEAME *Adapt* to bridge the domain gap, namely from $l_1l_2 \rightarrow l_1$ and $l_1l_2 \rightarrow l_2$ towards $l_1l_2 \rightarrow l_1l_2$.

Synthetic CS In this contrastive experiment, we remove the bilingual constraint, *i.e.* equation 4, from BALM, and use offline synthetic CS text outlined in Lee et al. (2019) in the training. The idea of synthetic CS is motivated by the Matrix Language Frame theory. The phrase alignment is performed on the same parallel dataset in Table 1, using Giza++ (Och and Ney, 2003). The aligned parallel sentences are then used to randomly switch phrases between the languages according to an empirical probability of 0.7. At the same, time the phrase table is used to inhibit switch within frequently occurring phrases. We train the same BALM network with both the synthetic CS data and the monolingual side of the parallel data. The model is finally adapted with SEAME *Adapt*.

Mono & CS-Only In the Mono setting, we simply use parallel corpus as two independent monolingual corpora without any form of bilingual constraint. The monolingual sentences are passed alternating between the two languages to ensure a balanced training curriculum. The model is finally adapted with SEAME *Adapt*. This is similar to the Multilingual BERT pre-training under causal masking and subsequently fine-tune on the task dataset. The CS-Only model is trained only on the SEAME *Adapt* data without involving the parallel data.

Positional Embedding We also implement the sinusoidal encoding matrix (Vaswani et al., 2017) and the learned weight matrix for the positional embedding in model PE-S and PE-L respectively. Both models are implemented on top of the BALM model using the same training data. The positional embedding is an element-wise addition to the word embedding layer. For the learned matrix in PE-L, we treat it as another lookup table. We simply extend the embedding matrix with the additional entries for each pos. In the case of sinusoidal encoding, the extended matrix is fixed to be,

$$PE_{(pos,2i)} = sin(pos/10000^{2i/384})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/384}).$$

4.3 CS Point Perplexity

While the perplexity test on SEAME Test *CS* describes the overall performance of the model on CS sentences. As shown in Table 1, CS only takes place at an average occurrence (SPF) of 23% in the CS sentences. We would like to take a closer look at how the model performs only at those CS points, which is the main focus of this work. A lower perplexity suggests a better word prediction ability. The perplexity is evaluated on SEAME *Test CS*, in which we only include perplexity for the word that is preceded by a different language.

4.4 Bilingual Lexicon Induction

While BALM is mainly optimized for word prediction, it also establishes cross-lingual word correspondence through word embedding. To examine the quality of cross-lingual embedding, we conduct bilingual lexicon induction (BLI) experiments, and compare with other major cross-lingual pretraining models. The same parallel corpus in Ta-

Models	Training Data	PPL (SEAME Test)	PPL (Test EN/ZH)	PPL (Test CS)	PPL (CS Points)	WER
CS only	SEAME *Adapt*	180.09	147.42/139.96	198.09	650.82	28.02%
Mono	Monolingual+SEAME *Adapt*	131.54	96.33/99.99	146.37	554.71	27.62%
Synthetic CS	Parallel+SEAME *Adapt*	124.65	95.13/99.91	139.17	506.81	26.42%
BALM	Parallel+SEAME *Adapt*	**118.25**	**91.74/94.41**	**130.49**	**477.78**	**19.73%**
+ PE-S	Parallel+SEAME *Adapt*	135.22	101.78/106.12	151.05	561.11	26.24%
+ PE-L	Parallel+SEAME *Adapt*	143.29	107.34/109.54	161.12	578.02	27.16%

Table 2: Perplexity is reported on different test subsets, and at CS Points of *Test CS*. Word Error Rate (WER) for language normalization is reported for experiments in Section 4.5.

Model	SEAME *Dev*	SEAME *Eval*
RNNLM* (Adel et al., 2013a)	246.60	287.88
FL + OF* (Adel et al., 2013a)	219.85	239.21
FLM* (Adel et al., 2013b)	177.79	192.08
LSTM (Winata et al., 2018)	150.65	153.06
Multi-task (Winata et al., 2018)	141.86	141.71
Synthetic CS (Lee et al., 2019)	142.41	142.53
CSLM (Lee and Li, 2019)	128.12	129.85
BALM	102.79	103.20

Table 3: Code-switch language models trained on SEAME *Train* (see Table 1). The models with '*' are trained and tested on SEAME Phase I, which is approximately 60% smaller than SEAME Phase II.

ble 1 is used for training and the same dictionary[2] is used for testing for all models.

VecMap[3] (Artetxe et al., 2018) is a projection based CLWE alignment method which gives robust results using a unsupervised strategy (Glavaš et al., 2019). The respective monolingual embeddings are trained using fastText[4] (Bojanowski et al., 2017) with the default setup and 384 dimensions. The two monolingual embedding space are then mapped using the VecMap. BiSkip[5] (Luong et al., 2015) is jointly trained with word alignment constraint. We prepare the alignment using fast_align[6] (Dyer et al., 2013) following the similar procedure outlined in the paper.

For the BALM model, we use the embedding from the model without the SEAME adaptation phase for a fair comparison. These three models represent three distinct categories in CLWE implementation, *i.e.* projection-based, jointly learned, and deep learning based embedding for VecMap, BiSkip and BALM, respectively.

4.5 Language Normalization

Suppose that l_1 is the matrix language in a code-switch sentence $\mathbf{w}$. We would like to replace all l_2 tokens in $\mathbf{w}$ with their l_1 equivalent tokens, that is referred to as $l_1 l_2 \rightarrow l_1$. The normalized sentence $\hat{\mathbf{w}}^{l_1}$ can be expressed as, $\hat{\mathbf{w}}^{l_1} = \arg\max_{\mathbf{w}^{l_1}} p(\mathbf{w}^{l_1}|\mathbf{w})$.

In practice, when $\mathbf{w}$ is presented to BALM, as illustrated in Figure 1c, the network predicts a sequence of tokens one by one in the matrix language as follows,

$$\hat{\mathbf{w}}^{l_1} = \arg\max_{\{w_t^{l_1}\}} \prod_{i=1}^{t} p(w_t^{l_1}|\mathbf{w}, \mathbf{w}_{i<t}^{l_1}), \qquad (5)$$

The generated tokens $\mathbf{w}_{i<t}^{l_1}$ becomes the context for the next token $w_t^{l_1}$ in an auto-regressive manner. The sequence with the highest probability is simply computed using beam search, which is performed when the `eos` token is observed.

5 Results and Analysis

5.1 Perplexity Evaluation

We conduct two perplexity (PPL) test experiments, one for comparing the variations of BALM, another for benchmarking against the state-of-the-art.

Comparing the variations of BALM, we report the overall test PPL as well as the PPL of each components, *i.e.* Test EN/ZH and Test CS for each model discussed in Section 4.2. It is observed in Table 2 that BALM outperforms all other variations, with a PPL of 118.25 on SEAME *Test*. Mono, Synthetic CS and BALM all benefit from the use of data beyond SEAME Adapt. BALM represents the most effective use of the bilingual parallel corpus. All the results are reported according to the best performing model on SEAME *Valid* dataset.

Benchmarking against the state-of-the-art, we show in Table 3 that BALM achieves a PPL of 103.20 on SEAME *Eval*, which is a 20.52% reduc-

[2] https://github.com/facebookresearch/MUSE#ground-truth-bilingual-dictionaries

[3] https://github.com/artetxem/vecmap

[4] https://github.com/facebookresearch/fastText

[5] https://github.com/lmthang/bivec

[6] https://github.com/clab/fast_align

No.	Code-switch sentence	Normalized	Reference
1	when there is still test then 他们会练 like once a week or once in two weeks	when there is still test then they will practise like once a week or once in two weeks	when there is still test then they will practise like once a week or once in two weeks
2	i have a high chance of being 拒绝 by her because obviously 我跟她很不熟	i have a high chance of being rejected by her because obviously i am not very familiar with	i have a high chance of being rejected by her because obviously i am not very familiar with **her**
3	but comparative to last year i think 已经蛮不错了	but comparative to last year i think is quite good **lah**	but comparative to last year i think is quite good **already**
4	开学之前 i have already secured a job	开学之前我已经**有**了一个工作	开学之前我已经**找到**了一个工作
5	这种活动 is a bit challenging	这种活动是有点困难的	这种活动是有点困难的
6	星期六我就要 hand in 我的 assignment 了	星期六我就要去**做**我的**任务**了	星期六我就要**交**我的**功课**了

Table 4: Samples of language normalized CS text and the reference

tion over the best reported result of 129.85 (Lee and Li, 2019) on the same test data in the literature.

5.1.1 CS point perplexity

Let us examine the perplexity only at CS points. In Table 2, from CS-Only to Mono, we observe a 14.8% PPL reduction, from 650.82 to 554.71, as a result of the additional monolingual data. We have seen similar results in Lee et al. (2019); Gonen and Goldberg (2019). Our observation is also very similar to M-Bert and corroborates with the findings of Pires et al. (2019). The monolingual data contribute to a better word embedding, which is an integral part of the BALM. As the quality of the word embedding improves, so does the word prediction at the CS points.

We also observe that Synthetic CS shows a 8.6% PPL reduction, from 554.71 to 506.81 with the inclusion of the synthetic CS data. This is consistent with the observations in Lee et al. (2019) and Pratapa et al. (2018).

We further observe that BALM, which is trained on exactly the same parallel data as in Synthetic CS, but with a different objective function, outperforms Synthetic CS by 5.73% . This suggests that the quasi-translation loss function is an effective regularizer to enforce the linguistic constraint governing CS. We also confirm our aforementioned hypothesis that self-attention mechanism is able to attend to the appropriate bilingual context for word prediction without violating the grammar of the matrix language by qualitatively analysing the generated sentences from the model not yet adapted with CS *adapt*.

5.1.2 Positional embedding

Both the sinusoidal encoding and the learned encoding matrix degrade the model performance by 14.4% and 21.2% respectively. This result confirms our hypothesis that the attention mechanism is able to encode the mixed context well without positional embedding. The improvement of BALM over BALM+PE in the monolingual PPL also demonstrates that dropping the positional embedding is in fact beneficial.

Method	EN-ZH	ZH-EN
VecMap (Artetxe et al., 2018)	57.13%	48.46%
BiSkip (Luong et al., 2015)	35.54%	33.39%
BALM (our work)	56.24%	55.87%
Vocabulary Coverage	38.84%	31.72%

Table 5: BLI accuracy (%) for different methods on the same parallel corpus in Table 1 for training and the same dictionary[2] for testing.

5.2 Bilingual Lexicon Induction

The comparable performance justifies the premise that the model is able to find word-level correspondence, which enables the subsequent bilingual context encoding. As shown in Table 5, when inferring ZH (Chinese) words from EN (English), BALM (56.24%) shows comparable performance with VecMap (57.13%), that reported the state-of-the-art results in CLWE. However, BALM significantly outperforms VecMap in the inverse pair ZH-EN with an absolute 7.41% improvement (48.46% → 55.87%).

Two points to take note of, firstly, Glavaš et al. (2019) point out that BLI cannot be used as the only metric to assess the word embedding quality and we do not intend to do so. Secondly, while it is true that VecMap does not need the corpus to be parallel and ours does, so the comparison did not showcase the best ability of VecMap. However, the focus of this paper is not on comparing the best cross-lingual word embedding methods. We use

BLI performance as evidence to support our claim that BALM does not compromise on its CLWE while focusing on sequential modeling.

5.3 Language Normalization

As the code-switch sentence follows the syntactic structure of the matrix language, we assume that the matrix language is known in advance, for example, English for sentences 1-3, and Chinese for sentences 4-6 in Table 4. We observe that sometimes, mistakes can take the form of bad translation, however the normalized sentence still maintains an appropriate structure of the matrix language. The 6^{th} sentence of Table 4 is an example, which is wrongly normalized to "to do my assignment (in the sense of task)" instead of "hand in my assignment (in the sense of homework)". We report the WER on SEAME *Norm* between the normalized text and the reference. We observe in Table 2 that, with a WER of 19.73%, BALM outperforms other models in the same way as in the perplexity tests.

6 Conclusion

We note that BALM is an implementation of $l_1 l_2 \rightarrow l_1 l_2$. The experiments show that it outperforms all state-of-the-art models in the literature for similar tasks. The results validate the idea of bilingual attention. The same BALM can be used in $l_1 l_2 \rightarrow l_1$ or l_2 for language normalization. It can be further extended for $l_1 \rightarrow l_1 l_2$, or $l_2 \rightarrow l_1 l_2$ for code switch sentence generation, and $l_1 \rightarrow l_2$, or $l_2 \rightarrow l_1$ for machine translation.

Acknowledgments

We thank the reviewers for their insightful feedback. This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award No. AISG-GC-2019-002); the Programmatic Grant No. A18A2b0046 from the Singapore Government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain), Project title: Human Robot Collaborative AI for AME; and the National Research Foundation, Prime Minister's Office, Singapore under the National Robotics Programme, Project title: Human-Robot Interaction Phase 1 (Grant No. 192 25 00054).

References

Heike Adel, Dominic Telaar, Ngoc Thang Vu, Katrin Kirchhoff, and Tanja Schultz. 2014. Combining recurrent neural networks and factored language models during decoding of code-switching speech. In *NTERSPEECH-2014*, pages 1415–1419.

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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8411–8415.

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)*, pages 206–211, Sofia, Bulgaria. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

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.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anne-Marie Di Sciullo, Pieter Muysken, and Rajendra Singh. 1986. Government and code-mixing. *Journal of linguistics*, 22(1):1–24.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4173–4183, Hong Kong, China. Association for Computational Linguistics.

Bradley Hauer, Garrett Nicolai, and Grzegorz Kondrak. 2017. Bootstrapping unsupervised bilingual lexicon induction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 619–624, Valencia, Spain. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland. Association for Computational Linguistics.

Kejun Huang, Matt Gardner, Evangelos Papalexakis, Christos Faloutsos, Nikos Sidiropoulos, Tom Mitchell, Partha P. Talukdar, and Xiao Fu. 2015. Translation invariant word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1084–1088, Lisbon, Portugal. Association for Computational Linguistics.

Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. Language Modeling with Deep Transformers. In *Proc. Interspeech 2019*, pages 3905–3909.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining.

Grandee Lee, Thi-Nga Ho, Eng-Siong Chng, and Haizhou Li. 2017. A review of the Mandarin-English code-switching corpus: SEAME. In *Asian Language Processing (IALP), 2017 International Conference on*, pages 210–213. IEEE.

Grandee Lee and Haizhou Li. 2019. Word and class common space embedding for code-switch language modelling. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6086–6090.

Grandee Lee, Xianghu Yue, and Haizhou Li. 2019. Linguistically Motivated Parallel Data Augmentation for Code-Switch Language Modeling. In *Proc. Interspeech 2019*, pages 3730–3734.

Omer Levy, Anders Søgaard, and Yoav Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, Valencia, Spain. Association for Computational Linguistics.

Ying Li and Pascale Fung. 2013. Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7368–7372.

Ying Li 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, Doha, Qatar. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth*

International Conference on Language Resources and Evaluation (LREC), pages 923–929.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

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

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *CoRR*, abs/1906.01502.

Shana Poplack. 2000. Sometimes i'll start a sentence in spanish y termino en espanol: Toward a typology of code-switching. *The bilingualism reader*, 18(2):221–256.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. 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)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

David Sankoff. 1998. 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*, pages 8–21. Association for Computational Linguistics.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR*, abs/1702.03859.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Ivan Vulic and Marie-Francine Moens. 2016. Bilingual distributed word representations from document-aligned comparable data. *J. Artif. Int. Res.*, 55(1):953–994.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Code-switching language modeling using syntax-aware multi-task learning. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 62–67. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola. 2016. Ten pairs to tag – multilingual POS tagging via coarse mapping between embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1307–1317, San Diego, California. Association for Computational Linguistics.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA. Association for Computational Linguistics.