Neural Machine Translation Between Myanmar (Burmese) and Rakhine (Arakanese)

Thazin Myint Oo[‡] , **Ye Kyaw Thu**[†] and **Khin Mar Soe**[‡]
Natural Language Processing Lab., University of Computer Studies Yangon, Myanmar[‡]
Language and Semantic Technology Research Team (LST), NECTEC, Thailand [†]
Language and Speech Science Research Lab., Waseda University, Japan [†]
{thazinmyintoo, khinmarsoe}@ucsy.edu.mm, ka2pluskha2@gmail.com

Abstract

This work explores neural machine translation between Myanmar (Burmese) and Rakhine (Arakanese). Rakhine is a language closely related to Myanmar, often considered a dialect. We implemented three prominent neural machine translation (NMT) systems: recurrent neural networks (RNN), transformer, and convolutional neural networks (CNN). The systems were evaluated on a Myanmar-Rakhine parallel text corpus developed by us. In addition, two types of word segmentation schemes for word embeddings were studied: Word-BPE and Syllable-BPE segmentation. Our experimental results clearly show that the highest quality NMT and statistical machine translation (SMT) performances are obtained with Syllable-BPE segmentation for both types of translations. If we focus on NMT, we find that the transformer with Word-BPE segmentation outperforms CNN and RNN for both Myanmar-Rakhine and Rakhine-Myanmar translation. However, CNN with Syllable-BPE segmentation obtains a higher score than the RNN and transformer.

1 Introduction

The Myanmar language includes a number of mutually intelligible Myanmar dialects, with a largely uniform standard dialect used by most Myanmar standard speakers. Speakers of the standard Myanmar may find the dialects hard to follow. The alternative phonology, morphology, and regional vocabulary cause some problems in communication. Machine translation (MT) has so far neglected the importance of properly handling the spelling, lexical, and grammar divergences among language varieties. In the Republic of the Union of Myanmar, there are many ethnical groups, and dialectal varieties exist within the standard Myanmar language.

To address this problem, we are developing a Myanmar and Rakhine dialectal corpus with monolingual and parallel text. We conducted statistical machine translation (SMT) experiments and obtained results similar to previous research (Oo et al., 2018).

Deep learning revolution brings rapid and dramatic change to the field of machine translation. The main reason for moving from SMT to neural machine translation (NMT) is that it achieved the fluency of translation that was a huge step forward compared with the previous models. In a trend that carries over from SMT, the strongest NMT systems benefit from subtle architecture modifications and hyperparameter tuning.

NMT models have advanced the state of the art by building a single neural network that can learn representations better (Sutskever et al., 2014a). Other authors (Rikters et al., 2018) conducted experiments with different NMTs for less-resourced and morphologically rich languages, such as Estonian and Russian. They compared the multi-way model performance to one-way model performance, by using different NMT architectures that allow achieving state-of-the-art translation. For the multiway model trained using the transformer network architecture, the reported improvement over the baseline methods was +3.27 bilingual evaluation understudy (BLEU) points.

(Honnet et al., 2017) proposed solutions for the machine translation of a family of dialects, Swiss German, for which parallel corpora are scarcee. The authors presented three strategies for normalizing Swiss German input to address the regional and spelling diversity. The results show that character-based neural machine translation was the most promising strategy for text normalization and that in combination with phrase-based statistical machine translation it achieved 36% BLEU score. In their study, NMT outperformed SMT.

In our study, we performed the first comparative NMT analysis of Myanmar dialectal language with three prominent architectures: recurrent neural network (RNN), convolutional

Proceedings of VarDial, pages 80–88

neural network (CNN), and transformer. We investigated the translation quality of the corresponding hyper-parameters (batch size, learning rate, cell type, and activation function) in machine translation between the standard Myanmar and national varieties of the same group of languages. In addition, we used two types of segmentation schemes: word byte pair encoding (Word-BPE) segmentation and syllable byte pair encoding (Syllable-BPE) segmentation. We compared the performance of this method to SMT and NMT experiments with the RNN, transformer, and CNN. We found that the transformer with Word-BPE segmentation outperformed both CNN and RNN for both Myanmar-Rakhine and Myanmar-Rakhine translations. We also found that CNN with Syllable-BPE segmentation obtained a higher score compared with RNN and the transformer.

2 Rakhine Language

Rakhine (Arakanese) is one of the eight national ethnic groups in the Republic of the Union of Myanmar. The Arakan was officially altered to “Rakhine” in 1989 and is located on a narrow coastal strip on the west of Myanmar, 300 miles long and 50 to 20 miles wide. The total population in all countries is nearly 3 million. The Rakhine language has been studied by researchers. L.F-Taylor’s “The Dialects of Burmese” described comparative pronunciation, sentence construction, and grammar usage in Rakhine, Dawei, In-tha, Taung-yoe, Danu, and Yae. Professor Denise Bernot, in “The vowel system of Arakanese and Tavoyan,” mainly emphasized the vowels of standard Myanmar and Tavoyan (Dawei) in 1965. In “Three Burmese Dialects” (1969), the linguist John Okell studied the spoken language of Myanmar, Dawei, and In-tha: specifically, usage of grammar and vowel differences (OKELL, 1995). Although the Rakhine language used the script as Arakanese or Rakkhawanna Akkhara before at least the 8th century A.D., the current Rakhine script is nearly the same as the Myanmar script. Generally, the Arakanese language is mutually intelligible with the Myanmar language and has the same word order (namely, subject-object-verb (SOV)). Examples of parallel sentences in Myanmar (my) and Rakhine (rk) are given as follows.

rk: ဒယော တစ် ထည် ဇာလောက်လေး ။
my: လုံချည် တစ် ထည် ဘယ်လောက်လဲ ။
(“How much for a longyi?” in English)

rk: ကလေချေ တိ ဘောလုံး ကျောက် နီရေ ။
my: ကောင်လေး တွေ ဘောလုံး ကန် နေတယ် ။
(“Boys are playing football” in English)

rk: ဇာ ပြော နီချင့် ယင်းသူရို့ ။
my: သူတို့ ဘာ ပြော နေတာလဲ ။
(“What are they talking about” in English)

rk: အဘောင်သျင် ဈီး က သပုံ ဝယ် လာတယ် ။
my: အဘွား ဈေး က ဆပ်ပြာ ဝယ် လာတယ် ။
(“The grandmother buys soap from the market” in English)

3 Difference between the Rakhine and standard Myanmar language

The Rakhine language is a largely monosyllabic and analytic language, with a SOV word order, and it uses the Myanmar script. It is considered by some to be a dialect of the Myanmar language, though it differs significantly from the standard Myanmar language in its vocabulary and includes loan words from Bengali, Hindi, and English. Compared with the Myanmar language, the speech of the Rakhine language is likely to be closer to the written form. The Rakhine language notably retains an /r/ sound that has become /j/ in the Myanmar language. Rakhine speakers pronounce the medial " ျ" as “Yapint” (i.e., /j/ sound) and the medial "ြ" as “Rayit” (i.e., /r/ sound). Moreover, Myanmar vowel "ေ" (/e/ sound) is pronounced as "ိ" (/i/ sound) in Rakhine language. Thus, for example, the word “dog” in the Myanmar language is written as "ခွေး" (Khwe), and in the Rakhine language it is written as "ခွိး" (khwii). Similarly, Rakhine pronounce "ေး" (/e:/) for Myanmar pronunciation of " ဲ" (/ai/) syllable. Thus, Myanmar word "ပဲဟင်း" (peh-hinn) (pea curry in English) is pronounced "ပေးဟင်း" (pay-hinn) in the Rakhine language. Some Pali words are also used in the Rakhine language. For example, the word “guest” of Myanmar monks "အာဂန္တု" (Agantu) is used in normal speech of Rakhine and it is similar to the normal Myanmar word “guest,” "ဧည့်သည်" (Ai thay). In summary, the most significant differences between the Rakhine and Myanmar languages are in their pronunciation and vocabulary; there are no grammatical differences.

4 Segmentation

4.1 Word Segmentation

In both Myanmar and Rakhine texts, spaces are used to separate the phrases for easier reading. The spaces are not strictly necessary and are rarely used in short sentences. There are no clear rules for using spaces. Thus, spaces may (or may not) be inserted between words, phrases, and even between root words and their affixes. Although Myanmar sentences of ASEAN-MT corpus (Boonkwan and Supnithi, 2013) are already segmented, we have to consider some rules for manual word segmentation of Rakhine sentences. We defined Rakhine "word" to be a meaningful unit. Affix, root word, and suffix (s) are separated such as "စား ဗျာယ်", "စား ပီးဗျာယ်", "စား ဖို့ဗျာယ်". Here, "စား" ("eat" in English) is a root word and the others are suffixes for past and future tenses. As Myanmar language, Rakhine plural nouns are identified by the following particle. We added a space between the noun and the following particle: for example a Rakhine word "ကလိန်မေချေ တိ" (ladies) is segmented as two words "ကလိန်မေချေ" and the particle "တိ". In Rakhine grammar, particles describe the type of noun and are used after a number or text number. For example, a Rakhine word "အကြိုစေ့နှစ်ခတ်" ("two coins" in English) is segmented as "အကြိုစေ့ နှစ် ခတ်". In our manual word segmentation rules, compound nouns are considered as one word. Thus, a Rakhine compound word "ဖေ့သာ" + "အိတ်" ("money" + "bag" in English) is written as one word "ဖေ့သာအိတ်" ("wallet" in English). Rakhine adverb words such as "အဂယောင့်" ("really" in English), "အမြန်" ("quickly" in English) are also considered as one word. The following is an example of word segmentation for a Rakhine sentence in our corpus, and the meaning is "Among the four air conditioners in our room, two are out of order."

Unsegmented sentence:
အကျွန်ုပ်ရို့အခန်းထဲမှာဟိရေလီအီးစက်လေးလုံးမှာနှစ်လုံးပျက်နီရေ ။

Segmented sentence:
အကျွန်ုပ်ရို့ အခန်း ထဲမှာ ဟိ ရေ လီအီးစက် လေး လုံး မှာ နှစ် လုံး ပျက် နီရေ ။

4.2 Syllable Segmentation

Generally, Myanmar words are composed of multiple syllables, and most of the syllables are composed of more than one character. Syllables are also the basic units for pronunciation of Myanmar words. If we only focus on consonant-based syllables, the structure of the syllable can be described with Backus normal form (BNF) as follows:

$$Syllable := CMW[CK][D]$$

Here, C stands for consonants, M for medials, V for vowels, K for vowel killer character, and D for diacritic characters. Myanmar syllable segmentation can be done with a rule-based approach (Maung and Makami, 2008; Thu et al., 2013), finite state automaton (FSA) (Hlaing, 2012), or regular expressions (RE) (`https://github.com/ye-kyawthu/sylbreak`). In our experiments, we used RE-based Myanmar syllable segmentation tool named "sylbreak."

4.3 Byte-Pair-Encoding

(Sennrich et al., 2016) proposed a method to enable open-vocabulary translation of rare and unknown words as a sequence of subword units representing BPE algorithm (Gage, 1994). The input is a monolingual corpus for a language (one side of the parallel training data, in our case) and starts with an initial vocabulary, the characters in the text corpus. The vocabulary is updated using an iterative greedy algorithm. In every iteration, the most frequent bigram (based on the current vocabulary) in the corpus is added to the vocabulary (the merge operation). The corpus is again encoded using the updated vocabulary, and this process is repeated for a predetermined number of merge operations. The number of merge operations is the only hyperparameter of the system that needs to be tuned. A new word can be segmented by looking up the learnt vocabulary. For instance, a new word "rocket," ဒုံးပျံ may be segmented as ဒ@@ုံး ပျံ after looking up the learnt vocabulary, assuming ဒ and ုံး ပျံ as BPE units learnt during the training.

5 Encoder-Decoder Models for NMT

The core idea is to encode a variable-length input sequence of tokens into a sequence of vector representations, and then decode these representations into a sequence of output tokens. Formally, with a given sentence $X = x_1, ..., x_n$ and target sentence $Y = y_1, ..., y_m$, an

NMT system models $p(Y|X)$ as a target language sequence model, conditioning the probability of target word y_t on target history $Y_{1:t-1}$ and source sentence X. Both x_i and y_t are integer IDs given by the source and target vocabulary mapping, $\mathbf{V}_{src}$ and $\mathbf{V}_{trg}$, built from the training data tokens and represented as one-hot vectors $x_i \in \{0,1\}^{|\mathbf{V}_{src}|}$ and $y_t \in \{0,1\}^{|\mathbf{V}_{trg}|}$. These are embedded into e-dimensional vector representations (Vaswani et al., 2017) $\mathbf{E}_S\mathbf{x}_i$ and $\mathbf{E}_T\mathbf{y}_t$, using embedding matrices $\mathbb{R}^{e\times|\mathbf{V}_{src}|}$ and $\mathbf{E}_T \in \mathbb{R}^{e\times|\mathbf{V}_{trg}|}$. The target sequence is factorized as $p(Y|X;\theta) = \prod_{t=1}^{m} p(y_t|Y_{1:t-1}, X;\theta)$. The model, parameterized by θ, consists of an encoder and decoder part, which vary depending on the model architecture. $p(y_t|Y_{1:t-1}, X;\theta)$ is parameterized via a softmax output layer over some decoder representations $\bar{S}_t$:

$$p(y_t|Y_{1:t-1}, X;\theta) = softmax(\mathbf{W}_o\bar{s}_t+\mathbf{b}_o), \quad (1)$$

where $\mathbf{W}_o$ scales to the dimension of the target vocabulary $\mathbf{V}_{trg}$.

5.1 Stacked RNN with attention

The encoder consists of a bidirectional RNN followed by a stack of unidirectional RNNs. An RNN at layer l produces a sequence of hidden states $\mathbf{h}_1^l \dots \mathbf{h}_n^l$:

$$\mathbf{h}_i^l = f_{enc}(\mathbf{h}_i^{l-1}, \mathbf{h}_{i-1}^l), \quad (2)$$

where f_{rnn} is some non-linear function, such as a gated recurrent unit (GRU) or long short-term memory (LSTM) cell, and $\mathbf{h}_i^{l-1}$ is the output from the lower layer at step i. The bidirectional RNN on the lowest layer uses embeddings $\mathbf{E}_S\mathbf{x}_i$ as input and concatenates the hidden states of a forward and a reverse RNN: $\mathbf{h}_i^0 = [\overrightarrow{\mathbf{h}_i^0}; \overleftarrow{\mathbf{h}_i^0}]$. With deeper networks, learning becomes increasingly difficult (Hochreiter et al., 2001; Pascanu et al., 2012), and residual connections of the form $\mathbf{h}_i^l = \mathbf{h}_i^{l-1} + f_{enc}(\mathbf{h}_i^{l-1}, \mathbf{h}_{i-1}^l)$ become essential (He et al., 2016).

The decoder consists of an RNN to predict one target word at a time through a state vector $\mathbf{s}$:

$$\mathbf{s}_t = f_{dec}([\mathbf{E}_T\mathbf{y}_{t-1}; \bar{\mathbf{s}}_{t-1}], \mathbf{s}_{t-1}), \quad (3)$$

where f_{dec} is a multilayer RNN, $\mathbf{s}_{t-1}$ the previous state vector, and $\bar{\mathbf{s}}_{t-1}$ the source-dependent *attentional vector*. Providing the attentional vector as an input to the first decoder layer is also called *input feeding* (Luong et al., 2015). The initial decoder hidden state is a non-linear transformation of the last encoder hidden state: $\mathbf{s}_0 = \tanh(\mathbf{W}_{init}\mathbf{h}_n + \mathbf{b}_{init})$. The attentional vector $\bar{\mathbf{s}}_t$ combines the decoder state with a *context vector* $\mathbf{c}_t$:

$$\bar{\mathbf{s}}_t = \tanh(\mathbf{W}_{\bar{s}}[\mathbf{s}_t; \mathbf{c}_t]), \quad (4)$$

where $\mathbf{c}_t$ is a weighted sum of encoder hidden states: $\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{ti}\mathbf{h}_i$. The attention vector α_t is computed by an attention network (Bahdanau et al., 2014; Luong et al., 2015):

$$\begin{aligned} \alpha_{ti} &= \text{softmax}(\text{score}(\mathbf{s}_t, \mathbf{h}_i)) \\ \text{score}(\mathbf{s}, \mathbf{h}) &= \mathbf{v}_a^\top \tanh(\mathbf{W}_u\mathbf{s} + \mathbf{W}_v\mathbf{h}). \end{aligned} \quad (5)$$

5.2 Self-Attentional Transformer

The transformer model (Vaswani et al., 2017) uses attention to replace recurrent dependencies, making the representation at time step i independent from the other time steps. This requires the explicit encoding of positional information in the sequence by adding fixed or learned positional embeddings to the embedding vectors.

The encoder uses several identical blocks consisting of two core sublayers: self-attention and a feedforward network. The self-attention mechanism is a variation of the dot-product attention (Luong et al., 2015) generalized to three inputs: query matrix $\mathbf{Q} \in \mathbb{R}^{n\times d}$, key matrix $\mathbf{K} \in \mathbb{R}^{n\times d}$, and value $\mathbf{V} \in \mathbb{R}^{n\times d}$, where d denotes the number of hidden units. (Vaswani et al., 2017) further extend attention to multiple *heads*, allowing for focusing on different parts of the input. A single *head* u produces a context matrix

$$\mathbf{C}_u = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{W}_u^Q(\mathbf{K}\mathbf{W}_u^K)^T}{\sqrt{d_u}}\right)\mathbf{V}\mathbf{W}_u^V, \quad (6)$$

where matrices $\mathbf{W}_u^Q$, $\mathbf{W}_u^K$, and $\mathbf{W}_u^V$ are in $R^{d\times d_u}$. The final context matrix is given by concatenating the heads, followed by a linear transformation: $\mathbf{C} = [\mathbf{C}_1; \dots; \mathbf{C}_h]\mathbf{W}^O$. The form in Equation 6 suggests parallel computation across all time steps in a single large matrix multiplication. Given a sequence of hidden states $\mathbf{h}_i$ (or input embeddings), concatenated to $\mathbf{H} \in \mathbb{R}^{n\times d}$, the encoder computes self-attention using $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{H}$. The second subnetwork of an encoder block is a feedforward network with ReLU activation defined as

$$FFN(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \quad (7)$$

which is also easily parallelizable across time steps. Each sublayer, self-attention and feedforward network, is followed by a postprocessing stack of dropout, layer normalization, and residual connection.

The decoder uses the same self-attention and feedforward networks subnetworks. To maintain auto-regressiveness of the model, self-attention is masked out on future time steps according to (Vaswani et al., 2017). In addition to self-attention, a source attention layer, which uses the encoder hidden states as key and value inputs, is added. Given decoder hidden states $\mathbf{S} \in \mathbb{R}^{m \times s}$ and the encoder hidden states of the final encoder layer $\mathbf{H}^l$, source attention is computed as in Equation 5 with $\mathbf{Q} = \mathbf{S}, \mathbf{K} = \mathbf{H}^l, \mathbf{V} = \mathbf{H}^l$. As in the encoder, each sublayer is followed by a postprocessing stack of dropout, layer normalization (Ba et al., 2016), and residual connection.

5.3 Fully Convolutional Models

The convolutional model (Gehring et al., 2017) uses convolutional operations and also dispenses with recurrence. Hence, input embeddings are again augmented with explicit positional encodings.

The convolutional encoder applies a set of (stacked) convolutions that are defined as

$$\mathbf{h}_i^l = v(\mathbf{W}^l[\mathbf{h}_{i-\lfloor k/2 \rfloor}^{l-1}; \ldots; \mathbf{h}_{i+\lfloor k/2 \rfloor}^{l-1}] + \mathbf{b}^l) + \mathbf{h}_i^{l-1}, \quad (8)$$

where v is a non-linearity such as a gated linear unit (Gehring et al., 2017; Dauphin et al., 2016), and $\mathbf{W}^l \in \mathbb{R}^{d_{cnn} \times kd}$ are the convolutional filters. To increase the context window captured by the encoder architecture, multiple layers of convolutions are stacked. To maintain sequence length across multiple stacked convolutions, inputs are padded with zero vectors.

The decoder is similar to the encoder but adds an attention mechanism to every layer. The output of the target side convolution

$$\mathbf{s}_t^{l*} = v(\mathbf{W}^l[\bar{\mathbf{s}}_{t-k+1}^{l-1}; \ldots; \bar{\mathbf{s}}_t^{l-1}] + \mathbf{b}^l) \quad (9)$$

is combined to form $\mathbf{S}^*$ and then fed as an input to the attention mechanism of Equation 6 with a single attention head and $\mathbf{Q} = \mathbf{S}^*, \mathbf{K} = \mathbf{H}^l, \mathbf{V} = \mathbf{H}^l$, resulting in a set of context vectors $\mathbf{c}_t$. The full decoder hidden state is a residual combination with the context such that

$$\bar{\mathbf{s}}_t^l = \mathbf{s}_t^{l*} + \mathbf{c}_t + \bar{\mathbf{s}}_t^{l-1} \quad (10)$$

To avoid convolving over future time steps at time t, the input is padded to the left.

6 Experiments

6.1 Corpus Preparation and Statistics

We used 18,373 Myanmar sentences (with no name entity tags) of the ASEAN-MT Parallel Corpus (Boonkwan and Supnithi, 2013), which is a parallel corpus in the travel domain. It contains six main categories: people (greeting, introduction, and communication), survival (transportation, accommodation, and finance), food (food, beverages, and restaurants), fun (recreation, traveling, shopping, and nightlife), resource (number, time, and accuracy), special needs (emergency and health). Manual translation into the Rakhine language was done by native Rakhine students from two Myanmar universities, and the translated corpus was checked by the editor of a Rakhine newspaper. Word segmentation for Rakhine was done manually, and there are exactly 123,018 words in total. We used 14,076 sentences for training, 2,485 sentences for development, and 1,812 sentences for evaluation.

6.2 Moses SMT system

We used the Moses toolkit (Koehn et al., 2007) for training the operation sequence model (OSM) statistical machine translation systems. We did not consider phrase-based statistical machine translation (PBSMT) and hierarchical phrase-based statistical machine translation (HPBSMT), because the OSM approach achieved the highest BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010) scores among three approaches (Oo et al., 2018) for both Myanmar-Rakhine to Rakhine-Myanmar statistical machine translations. The word-segmented (i.e., Syllable-BPE and Word-BPE) source language was aligned with the word-segmented target language using GIZA++. The alignment was symmetrized by *grow-diag-final* and *heuristic*. The lexicalized reordering model was trained with the *msd-bidirectional-fe* option. We used KenLM (Heafield, 2011) for training the 5-gram language model with modified Kneser-Ney discounting. Minimum error rate training (MERT) (Och, 2003) was used to tune the decoder parameters, and the decoding was done using the Moses decoder (version 2.1.1) (Koehn et al., 2007). We used the default settings of Moses for all experiments.

Batch Size	RNN		Transformer		CNN	
	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my
128	79.86	81.44	79.64	82.01	**80.82**	**83.59**
256	80.76	82.94	79.47	81.37	80.33	83.54
512	80.00	82.26	79.47	80.79	79.86	81.38

Table 1: BLEU scores of Syllable-BPE segmentation with different batch sizes for three NMT models

Batch Size	RNN		Transformer		CNN	
	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my
128	60.02	44.44	72.70	72.82	69.03	72.24
256	60.31	46.47	**73.39**	72.45	65.61	68.26
512	42.76	34.93	73.30	**72.95**	67.89	71.68

Table 2: BLEU scores of Word-BPE segmentation with different batch sizes for three NMT models

Learning rate	RNN				Transformer			
	GRU		LSTM		GRU		LSTM	
	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my
0.0001	79.47	81.37	79.48	80.88	80.76	82.94	80.26	83.02
0.0002	79.82	81.65	**82.85**	82.07	80.88	81.54	80.90	82.99
0.0003	80.22	82.23	80.24	82.13	80.92	82.63	81.78	83.30
0.0004	80.65	82.66	80.85	82.33	81.25	82.54	**81.92**	**84.06**
0.0005	80.41	81.46	81.98	**83.86**	80.57	82.30	80.65	82.51

Table 3: BLEU scores for batch size 256 of Syllable-BPE segmentation with different learning rates and two memory cell types on RNN and the transformer

Learning rate	Batch Size 128				Batch Size 256			
	ReLu		Soft-ReLu		ReLu		Soft-ReLu	
	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my
0.0001	**81.37**	83.29	80.00	81.97	80.26	81.97	80.03	81.08
0.0002	81.01	82.24	79.89	82.50	80.07	82.29	80.01	81.51
0.0003	80.99	81.59	80.11	**83.34**	81.16	81.69	82.14	**84.08**
0.0004	N/A	N/A	N/A	N/A	79.74	80.87	**83.75**	83.06
0.0005	N/A	N/A	N/A	N/A	79.05	82.43	81.44	83.31

Table 4: BLEU scores for batch sizes 128 and 256 of Syllable-BPE segmentation with different learning rates and two activation functions on CNN

Segmentation	OSM		RNN		Transformer		CNN	
	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my	my-rk	rk-my
Syllable-BPE	82.71	**84.36**	82.03	83.98	82.85	82.65	**83.75**	84.08
Word-BPE	77.12	75.27	60.31	46.47	73.39	72.95	69.03	72.24

Table 5: Comparison of SMT and NMT (top BLEU scores) on two segmentation schemes

6.3 Framework for NMT

An open-source sequence-to-sequence toolkit for NMT written in Python (Hieber et al., 2017) and built on Apache MXNET (Chen et al., 2015), the toolkit offers scalable training and inference for the three most prominent encoder-decoder architectures: attentional recurrent neural network (Schwenk, 2012; Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014b), self-attentional transformers (Vaswani et al., 2017), and fully convolutional networks (Gehring et al., 2017).

6.4 Training Details

We used the Sockeye toolkit, which is based on MXNet, to train NMT models. The initial learning rate is set to 0.0001. If the per-

formance on the validation set has improved for 8 checkpoints, the learning rate is multiplied by 32 checkpoints. All the neural networks have eight layers. For RNN Seq2Seq, the encoder has one bi-directional LSTM and six stacked unidirectional LSTMs, and the encoder is a stack of eight unidirectional LSTMs. The size of hidden states is 512. We apply layer-normalization and label smoothing (0.1) in all models. We tie the source and target embeddings. The dropout rate of the embeddings and transformer blocks is set to (0.1). The dropout rate of RNNs is (0.2). The attention mechanism in the transformer has eight heads.

We applied three different batch sizes (128, 256, and 512) for RNN, Transformer, and CNN network architectures. The learning rate varies from 0.0001 to 0.0005. Two memory cell types GRU and LSTM were used for the RNN and transformer. Moreover, two activation functions were applied to the CNN architecture. The comparison between Syllable-BPE and Word-BPE segmentation schemes was done for both SMT (i.e., OSM) and NMT (RNN, Transformer, and CNN) techniques. All experiments are run on NVIDIA Tesla K80 24GB GDDR5. We trained all models for the maximum number of epochs using the AdaGrad and adaptive moment estimation (Adam) optimizer. The BPE segmentation models were trained with a vocabulary of 8,000.

6.5 Evaluation

We used automatic criteria to evaluate the machine translation output. The metric BLEU (Papineni et al., 2002) measures the adequacy of the translation between language pairs, such as Myanmar and English. The Higher BLEU scores are better. Before computing BLEU, the translations were decomposed into their constituent syllables to ensure that the results are cross-comparable.

7 Results and Discussion

The BLEU score results for three NMT approaches (RNN, Transformer, and CNN) with three batch sizes (128, 256, and 512) for Syllable-BPE segmentation scheme are shown in Table 1. Bold numbers indicate the highest BLEU score among different batch sizes. CNN achieved the highest BLEU scores for both Myanmar-Rakhine and Rakhine-Myanmar translations. However, the transformer architecture achieved the top BLEU scores for Word-BPE segmentation schemes for both Myanmar-Rakhine and Rakhine-Myanmar neural machine translations (see Table 2).

From the experimental results of Table 1 and Table 2, we noticed that RNN and Transformer NMT with Syllable-BPE have a decreased translation performance for batch size 512. Thus, we used batch size 256 for further experiments with the RNN and transformer architectures. The NMT performance of the RNN and transformer with Syllable-BPE segmentation schemes together with different learning rates (from 0.0001 to 0.0005) and two different memory cell types (GRU and LSTM) can be seen in Table 3. From these BLEU scores of the RNN and transformer approaches, LSTM gave the highest NMT performance for both Myanmar-Rakhine dialect translation and vice versa.

To observe the maximum translation performance of CNN architecture, we extended experiments by using two activation functions (ReLu and Soft-ReLu), two batch sizes (128 and 256), and five learning rates (from 0.0001 to 0.0005) (see Table 4). Here, bold numbers indicate the highest BLEU scores of each batch size. From these results, we can clearly see that Soft-ReLu achieved the highest BLEU scores for both Myanmar to Rakhine and Rakhine to Myanmar translations. We found that the training processes with learning rate 0.0004 and 0.0005 were stopped for the batch size 128 for both ReLu and Soft-ReLu activation functions.

We also made a comparison between SMT and NMT, and the results can be seen in Table 5. In this study, we run only OSM approach for the SMT experiments based on the our previous SMT work (Oo et al., 2018). The Table 5 presents that although CNN achieved the top BLEU score (83.75) for Myanmar to Rakhine translation, OSM gave the best BLEU (84.36) score for Rakhine to Myanmar translation. Furthermore, we also found that Syllable-BPE segmentation scheme is working well for both SMT and NMT for Myanmar-Rakhine dialect language pair.

As shown in the experimental results of Table 1 to Table 5, our dialect NMT experiments give significantly higher BLEU scores than other SMT on different language pairs such as Myanmar-Chinese, Myanmar-German, Myanmar-Japanese, Myanmar-Malaysian, Myanmar-Korean, Myanmar-Spanish, Myanmar-Thai, Myanmar-Vietnamese (Thu et al., 2016), and also for NMT on Myanmar-English (Sin and Soe, 2018). As we discussed in Section 3, Rakhine and Myanmar have the

same word order of SOV and also share a lot of vocabulary. For these reasons, we assume that both SMT and NMT systems reach a very high machine translation performance.

8 Conclusion

This paper presents the first study of the neural machine translation between Standard Myanmar and Rakhine (a spoken Myanmar dialect). We implemented three prominent NMT systems: RNN, transformer, and CNN. The systems were evaluated on a Myanmar-Rakhine parallel text corpus that we are developing. We also investigated two types of segmentation schemes (Word-BPE segmentation and Syllable-BPE segmentation). Our results clearly show that the highest performance of SMT and NMT was obtained with Syllable-BPE segmentation for both Myanmar-Rakhine and Rakhine-Myanmar translation. If we only focus on NMT, we find that the transformer with Word-BPE segmentation outperforms CNN and RNN for both Myanmar-Rakhine and Rakhine-Myanmar. We also find that CNN with syllable-BPE segmentation obtains a higher BLEU score compared with the RNN and transformer. In the near future, we plan to conduct a further study with a focus on NMT models with one more subword segmentation scheme SentencePiece for Myanmar-Rakhine NMT. Moreover, we intend to investigate SMT and NMT approaches for other Myanmar dialect languages, such as Myeik and Dawei.

References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Prachya Boonkwan and Thepchai Supnithi. 2013. Technical report for the network-based asean language translation public service project. *Online Materials of Network-based ASEAN Languages Translation Public Service for Members, NECTEC*.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

Tin Htay Hlaing. 2012. Manually constructed context-free grammar for myanmar syllable structure. In *In Proceedings of the Student Research Workshop at the 13 th Conference of the European Chapter of the Association for Computational Linguistics (EACL'12)*, pages 32–37.

Sepp Hochreiter, Yoshua Bengio, and Paolo Frasconi. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In J. Kolen and S. Kremer, editors, *Field Guide to Dynamical Recurrent Networks*. IEEE Press.

Pierre-Edouard Honnet, Andrei Popescu-Belis, Claudiu Musat, and Michael Baeriswyl. 2017. Machine translation of low-resource spoken dialects: Strategies for normalizing swiss german. *CoRR*, abs/1710.11035.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

Zin Maung Maung and Yoshiki Makami. 2008. A rule-based syllable segmentation of myanmar text. In *Proceedings of IJCNLP-08 work-shop of NLP for Less Privileged Language*, pages 51–58.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

John OKELL. 1995. Three burmese dialects. *Papers in Southeast Asian Linguistics No.13, Studies in Burmese Languages*, 13:1–138.

Thazin Myint Oo, Ye Kyaw Thu, and Khin Mar Soe. 2018. Statistical machine translation between myanmar (burmese) and rakhine (arakanese). In *Proceedings of ICCA2018*, pages 304–311.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063.

Matīss Rikters, Mārcis Pinnis, and Rihards Krišlauks. 2018. Training and Adapting Multilingual NMT for Less-resourced and Morphologically Rich Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING 2012: Posters*, pages 1071–1080. The COLING 2012 Organizing Committee.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Yi Mon Shwe Sin and Khin Mar Soe. 2018. Syllable-based myanmar-english neural machine translation. In *Proceedings of ICCA2018*, pages 228–233.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014a. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014b. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Ye Kyaw Thu, Finch Andrew, Win Pa Pa, and Sumita Eiichiro. 2016. A large-scale study of statistical machine translation methods for myanmar language. In *Proceedings of SNLP2016*.

Ye Kyaw Thu, Finch Andrew, Sagisaka Yoshinori, and Sumita Eiichiro. 2013. A study of myanmar word segmentation schemes for statistical machine translation. In *Proceedings of ICCA2013*, pages 167–179.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.