XMU Neural Machine Translation Systems
for WAT2018 Myanmar-English Translation Task

Boli Wang, Jinming Hu, Yidong Chen and Xiaodong Shi∗
School of Information Science and Engineering, Xiamen University, Fujian, China
{boliwang, todtom}@stu.xmu.edu.cn
{ydchen, mandel}@xmu.edu.cn

Abstract
This paper describes the Neural Machine Translation systems of Xiamen University for the Myanmar-English translation tasks of WAT 2018. We apply Unicode normalization, training data filtering, different Myanmar tokenizers, and subword segmentation in data pre-processing. We try to train NMT models with different architectures. The experimental results show that the RNN-based shallow models can still outperform Transformer models in some settings. And we also found that replacing the official Myanmar tokenizer with syllable segmentation does help improve the result.

1 Introduction
In recent years, Neural Machine Translation (NMT) (Bahdanau et al., 2015; Cho et al., 2014; Sutskever et al., 2014) has achieved state-of-the-art performance on various language pairs (Sennrich et al., 2016a; Wu et al., 2016; Zhou et al., 2016; Vaswani et al., 2017). This paper describes the NMT systems of Xiamen University (XMU) for the WAT 2018 evaluation (Nakazawa et al., 2018). We participated in Myanmar→English and English→Myanmar translation subtasks.

In both two translation directions, we compare state-of-the-art Transformer models (Vaswani et al., 2017) with our reimplementation of RNN-based dl4mt models†. In pre-processing, We try Unicode normalization, data filtering and Myanmar syllable segmentation. We also use Byte Pair Encoding (BPE) (Sennrich et al., 2016b) to achieve open-vocabulary translation.

The remainder of this paper is organized as follows: Section 2 describes architecture of NMT we use, including the training details. Section 3 describes the processing of the data. Section 4 shows the results of our experiments. Finally, we conclude in section 5.

2 Baseline System
We compare two NMT architectures:

• DL4MT: We use an in-house reimplementation of dl4mt-tutorial model with minor changes and new features such as dropout (Srivastava et al., 2014).

• Transformer: We use the reimplementation of Transformer model in THUMT toolkit (Zhang et al., 2017).

For both two subtasks, we train our models with almost the same hyper-parameters. For DL4MT, we use word embeddings of size 512 and hidden layers of size 1024. We use mini-batches of size 128 and adopt Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$) as the optimizer. The initial learning rate is set to $5 \times 10^{-4}$. During the training process, we halve the learning rate after every 10K batches. As a common way to train RNN models, we clip the norm of gradients to a predefined value 1.0 (Pascanu et al., 2013). We use dropout to avoid over-fitting with a keep probability of 0.8.
For Transformer, we set both word embeddings and hidden layers as 512 dimension. Transformer models are trained on 8 Nvidia GeForce GTX 1080 Ti graphics cards with batch size of 6400 tokens each card. The initial learning rate is set to 1.0 and Linear Warm-up RSqrt decay function is used with 5000 warm-up steps.

During the training process, we save the parameters as checkpoints for every 5K steps and evaluate the intermediate models on validation set. We train DL4MT models for 40K steps and Transformer models for 100K steps.

3 Data Processing

We use all training data provided by ALT corpus and UCSY corpus and the data processing in both Myanmar→English and English→Myanmar are almost the same. We normalize both Myanmar and English texts by converting Normalization Form Canonical Decomposition to Normalization Form Canonical Composition and applying a modified version of Moses\(^2\) normalize-punctuation.perl script with more punctuation normalization rules.

On the Myanmar side, the original training set is pre-tokenized and -Romanized with the official tokenizer myan2roma.py. However, as illustrated in Figure 1, we found a number of worse tokenized word types with multiple syllables in the long tail of Myanmar vocabulary, which intensify data sparsity. Therefore, we try to import Myanmar syllable segmentation before Romanization. We first recover the original Myanmar texts using official myan2roma.py script and then segment Myanmar syllables with MyanmarParser\(^3\) toolkit. Finally, we use myan2roma.py to Romanize the syllabificated Myanmar texts, without further tokenization. On the English side, Moses tokenizer and truecaser are applied.

Furthermore, we found that the official Myanmar tokenizer myan2roma.py split numbers into sequences of digits and Latin words into sequences of letters, which makes the sentences become longer and inconsistent with the English side. Therefore, we split numbers in English texts into digits and remove sentence pairs which contains Latin words in Myanmar side.

We filter training data in several steps. We first remove duplicated sentence pairs. Secondly, we filter out bad encoded or untranslated sentence pairs. Thirdly, we use Moses clean-corpus-n.perl script to remove sentence pairs with too much tokens or imbalanced length ratio. Finally, we use fast-align toolkit\(^4\) to train word alignment and filter out bad sentence pairs according to the alignment scores.

To enable open-vocabulary, we apply subword-based translation approaches. In our preliminary experiments, we found that Byte Pair Encoding (BPE) works better than mixed word/character segmentation techniques. As Myanmar texts are already syllabificated, we only apply BPE\(^5\) on English texts with 20K operations.

In the post-processing step, we recover Myanmar sentences using official myan2roma.py and then remove all spaces and Romanize sentences again with myan2roma.py. For English sentences, we first restore words from subword pieces and then apply Moses detruecaser and detokenizer scripts.

\(^2\)http://statmt.org/moses/  
\(^3\)https://github.com/thantthet/MyanmarParser-Py  
\(^4\)https://github.com/clab/fast_align  
\(^5\)https://github.com/rsennrich/subword-nmt
4 Results

4.1 Experiments on Myanmar Tokenizers

Table 1 shows the experimental results of different Myanmar tokenization methods. We found that integrating Myanmar syllable segmentation to the official script significantly improve results on Myanmar→English translation, whatever NMT architecture used. This proves that Myanmar syllable segmentation does help alleviate data-sparsity problem. However, Myanmar syllable segmentation underperform the official Myanmar tokenizer on English→Myanmar translation with both two types of NMT architectures. This maybe due to the longer sequences and more ambiguities of target side outputs.

<table>
<thead>
<tr>
<th>Tokenizer</th>
<th>DL4MT EN-MY</th>
<th>Transformer EN-MY</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M2R</td>
<td>22.03</td>
<td>21.95</td>
<td></td>
</tr>
<tr>
<td>MP + M2R</td>
<td>21.23</td>
<td>20.57</td>
<td>13.86</td>
</tr>
<tr>
<td></td>
<td>11.45</td>
<td>14.22</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Experimental results on validation sets of different Myanmar tokenization methods. M2R denotes myan2roma.py and MP denotes MyanmarParser. Here, we use tokenized case-sensitive BLEU score with multi-bleu.perl script of Moses.

<table>
<thead>
<tr>
<th>System</th>
<th>EN-MY</th>
<th>MY-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL4MT</td>
<td>22.76</td>
<td>12.11</td>
</tr>
<tr>
<td>Transformer</td>
<td>21.57</td>
<td>12.71</td>
</tr>
</tbody>
</table>

Table 2: Experimental results on test sets of different NMT Architectures. Here, we report the online results provided by the automatic evaluation server.

4.2 Experiments on NMT Architectures

In this section, we compare NMT systems with different architectures. The results of online automatic evaluation⁶ are shown in Table 2. The deep self-attention based Transformer model beats the shallow RNN-based DL4MT model in Myanmar→English translation with +0.6 BLEU score, while DL4MT outperforms Transformer in English→Myanmar translation with +1.2 BLEU score.

5 Conclusion

We describe XMU’s neural machine translation systems for the WAT 2018 Myanmar→English and English→Myanmar translation tasks. In such a low-resourced settings, experiments show that shallow RNN-based models can still outperform Transformer models and Myanmar syllable segmentation is effective to alleviate data-sparsity.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant No. 61573294), the Ph.D. Programs Foundation of Ministry of Education of China (Grant No. 20130121110040), the Foundation of the State Language Commission of China (Grant No. WT135-10), the Outstanding Achievement Late Fund of the State Language Commission of China (Grant No. WT135-38) and the National High-Tech R&D Program of China (Grant No. 2012BAH14F03).

References


⁶http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html


