

# NECTEC’s Participation in WAT-2021

Zar Zar Hlaing<sup>1</sup>, Ye Kyaw Thu<sup>2,3</sup>, Thazin Myint Oo<sup>2</sup>, Mya Ei San<sup>4</sup>,  
Sasiporn Usanavasin<sup>4</sup>, Ponrudee Netisopakul<sup>1</sup>, Thepchai Supnithi<sup>3</sup>

<sup>1</sup>King Mongkut’s Institute of Technology Ladkrabang, Thailand

<sup>2</sup>Language Understanding Lab., Myanmar

<sup>3</sup>NECTEC, Thailand

<sup>4</sup>Sirindhorn International Institute of Technology (SIIT), Thammasat University, Thailand

{zarzarhlaing.it,yktnlp,queenofthazin,myaeisan1995}@gmail.com

sasiporn.us@siit.tu.ac.th, ponrudee@it.kmitl.ac.th,

thepchai.supnithi@nectec.or.th

## Abstract

In this paper, we report the experimental results of Machine Translation models conducted by a NECTEC team (Team-ID: NECTEC) for the WAT-2021 Myanmar-English translation task (Nakazawa et al., 2021). Basically, our models are based on neural methods for both directions of English-Myanmar and Myanmar-English language pairs. Most of the existing Neural Machine Translation (NMT) models mainly focus on the conversion of sequential data and do not directly use syntactic information. However, we conduct multi-source neural machine translation (NMT) models using the multilingual corpora such as string data corpus, tree data corpus, or POS-tagged data corpus. The multi-source translation is an approach to exploit multiple inputs (e.g. in two different formats) to increase translation accuracy. The RNN-based encoder-decoder model with attention mechanism and transformer architectures have been carried out for our experiment. The experimental results showed that the proposed models of RNN-based architecture outperform the baseline model for the English-to-Myanmar translation task, and the multi-source and shared-multi-source transformer models yield better translation results than the baseline.

## 1 Introduction

Machine translation (MT) is a quick and very effective way to communicate one language to another. MT consists of the automatic translation of human languages by using computers. The first machine translation systems were rule-based built only using linguistic information. The translation rules were manually created by experts. Although the rules are well defined, this process is very expensive and cannot translate well for all domains and languages. Currently, many researchers had successfully built the most popular machine translations such as SMT (Statistical

Machine Translation) and NMT (Neural Machine Translation) for various languages instead of rule-based translation.

NMT has become the state-of-the-art approach compared to the previously dominant phrase-based statistical machine translation (SMT) approaches. However, the existing NMT models do not directly use syntactic information. Therefore, we propose tree-to-string and pos-to-string NMT systems by the multi-source translation models. We conducted these multi-source translation models with Myanmar-English and English-Myanmar in both directions. The multi-source translation models conducted in our experiments are based on the multi-source and shared-multi-source approaches of the previous research work (Junczys-Dowmunt and Grundkiewicz, 2017). Figure 1 and Figure 2 show the architecture of multi-source translation models. For doing the training processes of proposed models by the transformer and s2s architectures, word-level segmentation and tree-format on the English corpus side and syllable-level segmentation on the Myanmar corpus side are applied in English-to-Myanmar translation. In addition, we used the syllable-level segmentation and POS-tagged word on the Myanmar corpus side, and word-level segmentation on the English side for conducting the Myanmar-to-English translation.

In this paper, section 2 will describe our MT systems. The experimental setup will be proposed in section 3. In section 4, the results of our experiments will be reported, and section 5 will present the error analysis on translated outputs. Finally, section 6 will conclude the report.

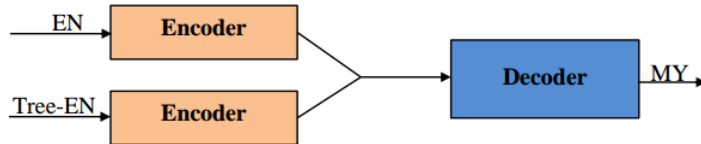


Figure 1: The basic idea of multi-source translation model for English-to-Myanmar translation

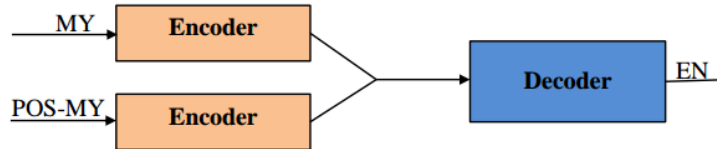


Figure 2: The basic idea of multi-source translation model for Myanmar-to-English translation

## 2 System Description

In this section, we describe the methodology used in our experiments for this paper. To build NMT systems, we chose the Marian framework<sup>1</sup>(Junczys-Dowmunt et al., 2018) with the architectures of Transformer and RNN based encoder-decoder model with attention mechanism (s2s). Marian is a self-contained neural machine translation toolkit focus on efficiency and research. This framework, the reimplementation of Nematus (Sennrich et al., 2017), is an efficient Neural Machine Translation framework written in pure C++ with minimal dependencies.

The main features of Marian are pure C++ implementation, one engine for GPU/CPU training and decoding, fast multi-GPU training and batched translation on GPU/CPU, minimal dependencies on external software (CUDA or MKL, and Boost), the static compilation (i.e., compile once, copy the binary and use anywhere), and permissive open-source MIT license. There are several model types supported by the Marian framework. Among them, we used *transformer*, *multi-transformer*, *shared-multi-transformer*, *s2s* (RNN-based encoder-decoder model with attention mechanism), *multi-s2s*, and *shared-multi-s2s* models for our experiment.

*transformer*: a model originally proposed by Google (Vaswani et al., 2017) based on attention mechanisms. *multi-transformer*: a transformer model but uses multiple encoders. *shared-multi-transformer*: is the same as multi-transformer but the difference is that the two encoders in shared-multi-transformer share parameters during training. *s2s*: an RNN-based encoder-decoder model with atten-

tion mechanism. The architecture is equivalent to the Nematus models (Sennrich et al., 2017). *multi-s2s*: s2s model but uses two or more encoders allowing multi-source neural machine translation. *shared-multi-s2s*: is the same as multi-s2s but the difference is that the two encoders in shared-multi-s2s share parameters during training.

In our experiments, two baseline models (transformer and RNN based attention: s2s) are used for the translation tasks of English-to-Myanmar and Myanmar-to-English. For the first translation task, the baseline models take single input of English tree data {tree-en} and produce the output of Myanmar string {my}. The multi-transformer, shared-multi-transformer, multi-s2s, and shared-multi-s2s models use two inputs of English string data and tree data {en, tree-en} and produce the output of Myanmar string {my}. For the second translation task, the input of Myanmar POS data {pos-my} is taken by the baseline models and produces the output of English string {en}. The multi-source and shared multi-source models take two inputs of Myanmar sting data and Myanmar POS data {my, pos-my} and yield the output of English string {en}. The baseline models, the multi-source and shared-multi-source models do the same action as the first translation task with different inputs and outputs.

## 3 Experimental Setup

### 3.1 Parallel Data

The parallel data for Myanmar-English and English-Myanmar translation tasks was provided by the organizers of the competition and consists of two corpora: the ALT corpus and the UCSY corpus. The ALT cor-

<sup>1</sup><https://github.com/marian-nmt/marian>

pus is one part of the Asian Language Treebank (ALT) Project (Riza et al., 2016) which consists of twenty thousand Myanmar-English parallel sentences from the Wikinews. The UCSY corpus (Yi Mon ShweSin et al., 2018) contains 238,014 sentences from various domains, including news articles and textbooks. The UCSY corpus for WAT-2021 is not identical to those used in WAT 2020 due to the extension of corpus size. Unlike the ALT corpus, Myanmar text in the UCSY corpus is not segmented. ALT corpus size is extremely small. And thus, the development data and test data were chosen from the ALT corpus. Moreover, we planned to do the experimental settings in training data with and without ALT training data because the test data are retrieved only from the ALT corpus. Due to the very limited hardware (only 2 GPUs and 8 GB memory workstation), the training time took very long and also crash several times, and we couldn’t manage to finish both of the experiments. Therefore, in this paper, we present the experimental results with the training data only using the UCSY corpus that contained around 238,000 lines. Table 1 shows data statistics used for the experiments.

## 3.2 Data Preprocessing

In this section, we describe the preprocessing steps before doing the training processes. Proper syllable segmentation or word segmentation is essential for the quality improvement of machine translation in the Myanmar language because this language has no clear definition of word boundaries. Although Myanmar text data in the ALT corpus are manual word segmentation data, those in the UCSY corpus are not segmented. Thus, we need to segment these data. We prepared both syllable and word segmentation for Myanmar language data. We used in-house **myWord**<sup>2</sup> segmenter for Myanmar word segmentation and Myanmar **sybreak**<sup>3</sup> segmenter for syllable segmentation. The myWord segmenter is a useful tool that can make the syllable segmentation, word segmentation, and phrase segmentation for the Myanmar language. In this paper, we used this tool only for word segmentation. The myWord segmenter tool will be released soon.

After doing the word segmentation process, we need to apply POS tagging to the segmented Myanmar data. In addition, for the

<sup>2</sup><https://github.com/ye-kyaw-thu/myWord>

<sup>3</sup><https://github.com/ye-kyaw-thu/sybreak>

English tree data, we also need to parse the English data. There are some reasons that we had implemented a multi-source NMT system for this paper. To the best of our knowledge, no experiments have been conducted for the multi-source NMT system using POS data and syntactic tree information. In particular, this multi-source NMT system has not been developed in the Myanmar language. There is only one Factored SMT paper (Ye Kyaw Thu et al., 2014) using Myanmar POS data. Thus, we had implemented a multi-source NMT system for Myanmar-to-English and English-to-Myanmar translations in this paper. To implement this system, we need to apply the POS tagging on the Myanmar data side and the tree data format on the English side. Although we desired to use the tree format on the Myanmar side, Myanmar data cannot be currently built like the English syntactic tree data format. And thus, we can only use Myanmar POS(Part-of-speech) data and English tree data format for implementing the multi-source translation models. Part-of-speech tagging and the parser that we used in our experiment will be described in the following sections.

### 3.2.1 Part-of-speech Tagging

For the part-of-speech (POS) Myanmar data, the segmented data obtained by the myWord segmenter was tagged by using the RDR model built-in **myPOS** version 2.0<sup>4</sup> (Zar Zar Hlaing et al., 2020). 16 POS Tag-sets (Khin War War Htike et al., 2017) were used in myPOS version 2.0. These POS tag-sets are **abb** (Abbreviation), **adj** (Adjective), **adv** (Adverb), **conj** (Conjunction), **fw** (Foreign word), **int** (Interjection), **n** (Noun), **num** (Number), **part** (Particle), **part\_neg** (Negative particle), **ppm** (Post-positional Marker), **pron** (Pronoun), **punc** (Punctuation), **sb** (Symbol), **tn** (Text Number) and **v** (Verb). Supervised tagging algorithms, namely, Conditional Random Fields (CRFs), Hidden Markov Model (HMM), Ripple Down Rules-based (RDR), and neural sequence labeling approach of Conditional Random Fields (NCRF++) were used to compare the tagging accuracies of the original myPOS version 1.0 (Khin War War Htike et al., 2017) and myPOS version 2.0. Among these four tagging methods, the RDR model gave the best tagging accuracy. Thus, we chose the RDR model for tagging the Myanmar data for our experiment. The example of POS-tagged

<sup>4</sup><https://github.com/ye-kyaw-thu/myPOS>

Table 1: English-Myanmar Parallel Dataset

Data Type	File Name	Number of Sentence
TRAIN	train.ucsy.[my   en]	238,014
DEV	dev.alt.[my   en]	1,000
TEST	test.alt.[my   en]	1,018

data for the sentence “ကျွန်တော် က သုတေသီ တစ်ယောက် ပါ။” (I am a researcher.) is described in the following:

ကျွန်တော်/pron က/ppm သုတေသီ/n တစ်/tn  
ယောက်/part ပါ/part ။/punc

We also evaluated the accuracy of the RDR model. To evaluate this model, 1,300 Myanmar sentences were retrieved from the UCSY corpus, and these sentences were tagged by the selected RDR model. On the other hand, we manually tagged these Myanmar sentences. Finally, we evaluated the accuracy of the RDR model by comparing these two tagged data. We found that the RDR model provides the tagging accuracy of 77% Precision, 81% Recall, and 79% F-Measure.

### 3.2.2 RegexpParser

Word-level segmentation and tree data format were used on the English side for the experiment. English data given by the WAT-2021 are already segmented. Thus, no segmentation process is needed to do for the English side. For parsing the English data, some parsers such as English PCFG (Probabilistic Context-Free Grammar) parser from Stanford Parser<sup>5</sup>, BLLIP Parser<sup>6</sup>, Berkeley Neural Parser<sup>7</sup>, and RegexpParser<sup>8</sup> were tested with our experiment data of English side. **PCFG Parser** is used to parse the English sentence into tree data format. This parser cannot parse long sentences of more than 70 words. The longest sentence in our experiment data contains approximately 1,000 words. And thus, this PCFG parser cannot be used for parsing our experiment data. **BLLIP Parser** is a statistical natural language parser that includes a generative constituent parser and discriminative maximum entropy re-ranker. It can be

<sup>5</sup><https://nlp.stanford.edu/software/lex-parser.shtml>

<sup>6</sup><https://github.com/BLLIP/bllip-parser>

<sup>7</sup><https://github.com/nikitakit/self-attentive-parser>

<sup>8</sup><https://www.programcreek.com/python/example/91255/nltk.RegexpParser>

used as Python version or Java version. This parser cannot parse the long sentences in our experiment data although it can accept more sentence length 853 than the PCFG parser.

**Berkeley Neural Parser** is a high-accuracy parser with models for 11 languages which is implemented by Python. It is based on constituency parsing with a self-attentive encoder, with additional changes in multilingual constituency parsing with self-attention and pre-training. Although this parser can parse the long sentences in our experiment data, training time takes a lot more than the **RegexpParser**<sup>9</sup> (**grammar-based chunk parser**) from **nltk** package. By comparing the aforementioned parsers, RegexpParser can parse the longest sentences and all the experiment data within a few minutes. Moreover, this RegexpParser is the simplest parser for generating the parse tree data. Thus, we chose the RegexpParser for the tree data format of the English side of our experiment data.

A grammar-based chunk parser **RegexpParser** uses a set of regular expression patterns to specify the behavior of the parser. The chunking of the text is encoded by using a ChunkString, and each rule performs by modifying the chunking in the ChunkString. The rules are implemented by using regular expression matching and substitution. A grammar contains one or more clauses in the following form:

```
{< DT | JJ >} #chunk determiners and adjectives
}< [\ · VI] · * > +{ #strip any tag beginning with V, I, or .
```

```
< · * >}{< DT > #split a chunk at a determiner
< DT | JJ > } < NN · * > #merge chunk ending with det /adj with one starting with a noun
```

The clauses of a grammar are also executed in order. A cascaded chunk parser is one having more than one clause. The maximum depth of a parse tree generated by RegexpParser is the same as the number of clauses in the gram-

<sup>9</sup><https://www.programcreek.com/python/example/91255/nltk.RegexpParser>



mar. To parse a sentence, firstly, we need to create the chunker by using the `RegexpParser` function with the built grammar. Secondly, an input sentence is needed to tokenize and the tokenized sentence will need to be tagged by using the functions from `nlk` package. After tagging the tokenized sentence, the chunker calls the `parse` function with the tagged string parameter. Later, we will get the parse tree format output and need to convert this tree format to the tree format string. These procedures were used for parsing the English side of our experiment data. The example of English parse tree produced by this `RegexpParser` is shown as follow:

```
(S I/PRP (VP (V love/VBP)) (VP (V programming/VBG)) ./.)
```

### 3.3 Training

All our NMT systems were trained on 2 GPUs with the following parameters for Marian framework. Two architectures such as *transformer* and *s2s* (RNN-based encoder-decoder model with attention mechanism) are applied in our experiment. For the first architecture, we used the different model types (`--type transformer` for Transformer Model, `--type multi-transformer` for Multi-Transformer Model, and `--type shared-multi-transformer` for Shared-Multi-Transformer Model) with the following parameters:

```
--max-length 500 --maxi-batch
100 --valid-freq 5000
--valid-metrics cross-entropy
perplexity bleu --save-freq
5000 --disp-freq 500
--valid-mini-batch 64
--beam-size
6 --normalize 0.6 --enc-depth 2
--mini-batch-fit -w 1000
--dec-depth 2 --transformer-heads
8 --transformer-dropout 0.3
--label-smoothing 0.1
--early-stopping 10
--tied-embeddings
--exponential-smoothing
--learn-rate 0.0003 --lr-warmup 0
--lr-decay-inv-sqrt 16000
--clip-norm 5 --devices 0 1
--sync-sgd --seed 1111
```

For the second architecture, we also used the different model types (`--type s2s` for RNN with attention Model, `--type`

`multi-s2s` for Multi-s2s Model, and `--type shared-multi-s2s` for Shared-Multi-s2s Model) with the following parameters:

```
--max-length 500 --workspace
500 --enc-depth 2 --enc-type
alternating --enc-cell
lstm --enc-cell-depth 2
--dec-depth 2 --dec-cell
lstm --dec-cell-base-depth
2 --dec-cell-high-depth
2 --mini-batch-fit
--valid-mini-batch 16
--valid-metrics cross-entropy
perplexity translation
bleu --valid-freq 5000
--save-freq 5000 --disp-freq
500 --dropout-rnn
0.3 --early-stopping
10 --tied-embeddings
--mini-batch-fit --dropout-src
0.3 --devices 0 1 --sync-sgd
--seed 1111
```

## 4 Evaluation Results

Our systems are evaluated on the ALT test set and the evaluation results are shown in Table 2. For the evaluation of Myanmar-to-English and English-to-Myanmar translation pairs, we used the different evaluation metrics such as Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002), Rank-based Intuitive Bilingual Evaluation Score (RIBES) (Isozaki et al., 2010), and Adequacy-Fluency Metrics (AMFM) (Banchs et al., 2015).

The BLEU score measures the precision of n-gram (overall  $n \leq 4$  in our case) with respect to a reference translation with a penalty for short translations. Intuitively, the BLEU score measures the adequacy of the translation and a larger BLEU score indicates a better translation quality. RIBES is an automatic evaluation metric based on rank correlation coefficients modified with precision and special care is paid to the word order of the translation results. The RIBES score is suitable for distance language pairs such as Myanmar and English. Larger RIBES scores indicate better translation quality. AM-FM is a two-dimensional automatic evaluation metric for machine translation, which is used to evaluate the machine translation systems. The evaluation metric designed to address independently the semantic and syntactic aspects of the translation. The larger the AMFM scores, the better the trans-

Table 2: BLEU, RIBES and AMFM scores for English-to-Myanmar and Myanmar-to-English translations

Models	English-to-Myanmar			Myanmar-to-English		
	BLEU	RIBES	AMFM	BLEU	RIBES	AMFM
<b>transformer</b>	12.72	<b>0.610951</b>	0.645760	<b>6.24</b>	<b>0.620840</b>	<b>0.424640</b>
<b>multi-transformer</b>	12.94	0.598012	<b>0.654780</b>	4.44	0.577247	0.393760
<b>shared-multi-transformer</b>	<b>13.90</b>	0.608810	0.645260	4.62	0.587155	0.391710
<b>s2s</b>	12.35	0.620377	0.618420	<b>6.72</b>	<b>0.616469</b>	<b>0.395310</b>
<b>multi-s2s</b>	<b>12.82</b>	0.625476	<b>0.638870</b>	4.73	0.578146	0.357150
<b>shared-multi-s2s</b>	12.11	<b>0.626460</b>	0.631630	6.13	0.609560	0.376140

lation quality. Experiments are conducted by tuning different parameter settings for the proposed models. The best scores among those of the experimental results are submitted in this description. The highest scores of the proposed models are indicated as bold numbers. Since the UCSY corpus is updated annually, we cannot directly compare the official baseline results of WAT-2020 and our experimental results of WAT-2021. Thus, the experimental results are compared only with our baseline model results.

Table 2 shows the experimental results of the first and second architectures. The first part of the table consists of English-to-Myanmar translation scores and the second part consists of Myanmar-to-English translation scores. For the first architecture (i.e., transformer) in the first part of the table, the shared-multi-transformer model achieves higher BLEU scores (+1.18) than the baseline transformer model. Furthermore, the multi-transformer model performs better than the baseline transformer in terms of AMFM scores. However, RIBES scores of multi-transformer and shared-multi-transformer models are lower than the baseline transformer model. For the second architecture (i.e., s2s or RNN-based Attention), the multi-s2s model outperforms the baseline s2s model and shared-multi-s2s in terms of BLEU and AMFM scores. The shared-multi-s2s model provides better RIBES scores (0.626460). The highest BLEU scores (13.90) of the shared-multi-transformer model and the highest AMFM scores (0.654780) of the multi-transformer model are produced by the first architecture while the highest RIBES scores (0.625476) are achieved by the multi-s2s model of the second architecture.

Myanmar-to-English translation results are

shown in the second part of the Table 2. For Myanmar to English translation, the two baseline models (i.e., transformer and s2s) outperform the other models in terms of BLEU, RIBES, and AMFM scores. No improvements occur in this translation task. On the other hand, from English to Myanmar translation, the multi-transformer model is better than the baseline transformer model in terms of AMFM score, and the shared-multi-transformer model performs better than the baseline in terms of BLEU score. Moreover, the multi-s2s and shared-multi-s2s models also provide better translation results compared with the baseline model.

## 5 Error Analysis

For both English-to-Myanmar and Myanmar-to-English translation models, we analyzed the translated outputs by using Word Error Rate<sup>10</sup>. For doing the error analysis, we used SCLITE (score speech recognition system output) program from the NIST scoring toolkit SCTK<sup>11</sup> version 2.4.10 for making dynamic programming based alignments between reference (ref) and hypothesis (hyp) and calculation of WER. The WER formula can be described as the following equation:

$$WER = \frac{(I + D + S)100}{N} \quad (1)$$

where  $S$  is the number of substitutions,  $D$  is the number of deletions,  $I$  is the number of insertions,  $C$  is the number of correct words and  $N$  is the number of words in the reference ( $N = S + D + C$ ). The percentage of WER can be greater than 100% when the number of insertions is very high.

<sup>10</sup>[https://en.wikipedia.org/wiki/Word\\_error\\_rate](https://en.wikipedia.org/wiki/Word_error_rate)

<sup>11</sup><https://github.com/usnistgov/SCTK>

Table 3: WER scores for English-to-Myanmar and Myanmar-to-English translation models (Generally, lower WER indicates better translation performance)

Models	English-to-Myanmar	Myanmar-to-English
	WER(%)	WER(%)
<b>transformer</b>	<b>81.3%</b>	<b>82.6%</b>
<b>multi-transformer</b>	83.9%	90.0%
<b>shared-multi-transformer</b>	83.5%	88.2%
<b>s2s</b>	84.2%	<b>85.1%</b>
<b>multi-s2s</b>	82.7%	91.8%
<b>shared-multi-s2s</b>	<b>82.5%</b>	86.0%

Table 3 shows the WER scores of English-to-Myanmar and Myanmar-to-English translation models. In this table, lower WER scores are highlighted as bold numbers. The lower the WER scores, the better the translation models. For the first architecture of English-to-Myanmar translation, the baseline transformer model gives lower WER scores (81.3%) than the multi-transformer and shared-multi-transformer models. However, in the second architecture, the shared-multi-s2s model provides lower WER scores (82.5%) compared with the baseline (s2s) and multi-s2s models. In Myanmar-to-English translation, the multi-transformer and shared-multi-transformer models yield greater WER scores (90.0% and 88.2%) than the baseline transformer model of the first architecture. In addition, the multi-s2s and shared-multi-s2s model also give higher WER scores (91.8% and 86.0%) than the baseline s2s models (85.1%). Due to the higher WER scores in Myanmar-to-English translation models, the multi-transformer and shared-multi-transformer models couldn't provide better translation results than the baseline transformer model, and the multi-s2s and shared-multi-s2s models couldn't also yield the improvements than the baseline s2s model.

After we analyzed the confusion pairs of English-to-Myanmar and Myanmar-to-English translation models in detail, we found that most of the confusion pairs in the translations are caused by (1) the nature of the Myanmar language (written or speaking form), (2) the incorrect word segmentation or data cleaning errors of English language, (3) the Myanmar language with no articles (i.e., **a**,

**an**, and **the**), and (4) the different nature and language gaps of Myanmar and English languages. The top 10 confusion pairs of English-to-Myanmar and Myanmar-to-English translations of the model transformer are shown in Table 4. In this table, the first column is the reference and hypothesis pair (i.e., output of the translation model) for English-to-Myanmar translation. The third one is for that of Myanmar-to-English translation.

All of the confusion pairs in the first column are caused by the nature of the Myanmar language. For example, in Myanmar written or speaking form, the word “သည် (“is” in English)” are the same as the word “တယ် (“is” in English)”. Moreover, the words “၏ (“of or 's” in English)” and “ရဲ့ (“of or 's” in English)” in the possessive place and the words “များ (“plural form” in English)” and “တွေ (“plural form” in English)” are the same meanings. In other words, these hypotheses are synonyms of the reference words. In the third column of the Table 4, for the Myanmar-to-English translation, the confusion pairs of “apos → quot”, “quot → apos”, “the → &”, “, → the” and “the → s” are caused by the incorrect word segmentation or data cleaning errors of English language. Furthermore, we found that the confusion pairs of “the → a” and “a → the” are caused by the Myanmar language with no articles (i.e., **a**, **an**, and **the**). The confusion pairs of “in → of”, “to → of” and “with → and” are caused due to the different nature and language gaps of Myanmar and English languages. Occasionally, most of the Myanmar people misused the usage of the words “**in**, **of**, and **with**” in English writing.

For instance, for the Myanmar sentence “သူ

Table 4: An example of confusion pairs of the model Transformer

EN-MY Ref→Hyp	Freq	MY-EN Ref→Hyp	Freq
သည် → တယ်	371	apos→ quot	30
များ → တွေ	63	the → a	29
၏ → ရဲ့	33	quot → apos	24
တယ် → သည်	36	, → the	23
သော → တဲ့	17	the → &amp;	23
ရန် → ဖို့	9	in → of	18
ယောက် → ဦး	9	a → the	17
မည် → မယ်	8	the → s	14
တို့ → များ	7	to → of	10
၎င်း → ဒါ	3	with → and	6

က အတန်း ထဲ မှာ အတော်ဆုံး ကျောင်းသား ဖြစ်တယ်။”, they translate this sentence to the English sentence “He is the most clever student **of** the class.”. In this case, they misused the word “**of**” instead of the word “**in**”. The correct English sentence is “He is the most clever student **in** the class.” For another example of Myanmar sentence “စားပွဲ ကို သစ်သား ဖြင့် ပြုလုပ် ထားတယ်။”, they translate to English sentence “The table is made **with** wooden.” with the misused of the word “**with**” instead of “**of**”. The correct sentence for this example is “The table is made **of** wooden.” When the prepositions “**in**, **of**, and **with**” are combined with the main verbs, the prepositions “**in** and **of**” and “**with** and **of**” have generally same meanings in Myanmar language. These may cause the Myanmar-to-English translation models hard to learn well during the training processes compared with the English-to-Myanmar translation models.

## 6 Conclusion

In this system description for WAT-2021, we submitted our NMT systems with two architectures such as transformer and RNN with attention. We evaluated our proposed models in both directions of Myanmar-English and English-Myanmar translations at WAT-2021. In this paper, for English to Myanmar translation, multi-source and shared-multi-source models outperform the baseline models in terms of BLEU, RIBES, and AMFM scores.

In the Myanmar-to-English translation task, the proposed models could not provide better translation quality than the baselines. The top 10 frequent errors in the model’s hypothesis could be clearly found from our error analysis. For examples, the confusion pairs of “သည် → တယ် 371”, “များ → တွေ 63”, “၏ → ရဲ့ 33”, and so on. Moreover, our study also made a contribution to the fact that if these errors can be cleaned up, the translation performance of the shared task will improve. In the future, we intend to apply post-editing techniques in Myanmar to English translation to improve the translation quality. Furthermore, we intend to extend string-to-tree and string-to-pos translation approaches for under-resourced languages such as Myanmar and Thai.

## Acknowledgments

This work was supported by King Mongkut’s Institute of Technology Ladkrabang (KMIL) and in collaboration with Artificial Intelligence Research Group, National Electronics and Computer Technology Center (NECTEC), Thailand.

## References

Rafael E. Banchs, Luis F. D’Haro, and Haizhou Li. 2015. *Adequacy-fluency metrics: Evaluating mt in the continuous space model framework.*



- IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):472–482.
- 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.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. [An exploration of neural sequence-to-sequence architectures for automatic post-editing](#).
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. [Marian: Fast neural machine translation in C++](#). In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.
- Khin War War Htike, Ye Kyaw Thu, Zuping Zhang, and Win Pa Pa. 2017. [Comparison of six pos tagging methods on 10k sentences myanmar language \(burmese\) pos tagged corpus](#). In *18th International Conference on Computational Linguistics and Intelligent Text Processing (CI-Ling 2017)*.
- Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, and Sadao Oda, Yusuke Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation*, Bangkok, Thailand. Association for Computational Linguistics.
- 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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Hammam Riza, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, L. C. Mai, Vu Tat Thang, N. Thai, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng, K. Soe, K. Nwet, M. Utiyama, and Chenchen Ding. 2016. [Introduction of the asian language treebank](#). *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6.
- Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. [Nematus: a toolkit for neural machine translation](#). In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Ye Kyaw Thu, Andrew Finch, Akihiro Tamura, Eiichiro Sumita, and Yoshinori Sagisaka. 2014. [Factored machine translation for myanmar to english, japanese and vice versa](#). In *Proceedings of the 12th International Conference on Computer Applications (ICCA 2014)*, pages 171–177, Yangon, Myanmar. Association for Computational Linguistics.
- Yi Mon ShweSin, Khin Mar Soe, and Khin Yadanar Htwe. 2018. [Large scale myanmar to english neural machine translation system](#). In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 464–465.
- Zar Zar Hlaing, Ye Kyaw Thu, Myat Myo Nwe Wai, Thepchai Supnithi, and Ponrudee Netisopakul. 2020. [Myanmar pos resource extension effects on automatic tagging methods](#). In *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pages 1–6.