Facebook AI's WAT19 Myanmar-English Translation Task Submission

Peng-Jen Chen*[1] Jiajun Shen*[1] Matt Le[1] Vishrav Chaudhary[2]
Ahmed El-Kishky[2] Guillaume Wenzek[1] Myle Ott[1] Marc'Aurelio Ranzato[1]
[1]Facebook AI Research [2]Facebook AI Applied Research
{pipibjc,jiajunshen,mattle,vishrav,
ahelk,guw,myleott,ranzato}@fb.com

Abstract

This paper describes Facebook AI's submission to the WAT 2019 Myanmar-English translation task (Nakazawa et al., 2019). Our baseline systems are BPE-based transformer models. We explore methods to leverage monolingual data to improve generalization, including self-training, back-translation and their combination. We further improve results by using noisy channel re-ranking and ensembling. We demonstrate that these techniques can significantly improve not only a system trained with additional monolingual data, but even the baseline system trained exclusively on the provided small parallel dataset. Our system ranks first in both directions according to human evaluation and BLEU, with a gain of over 8 BLEU points above the second best system.

1 Introduction

While machine translation (MT) has proven very successful for high resource language pairs (Ng et al., 2019; Hassan et al., 2018), it is still an open research question how to make it work well for the vast majority of language pairs which are low resource. In this setting, relatively little parallel data is available to train the system and the translation task is even more difficult because the language pairs are usually more distant and the domains of the source and target language match less well (Shen et al., 2019).

English-Myanmar is an interesting case study in this respect, because i) the language of Myanmar is morphologically rich and very different from English, ii) Myanmar language does not bear strong similarities with other high-resource languages and therefore does not benefit from multilingual training, iii) there is relatively little parallel data available and iv) even monolingual data in Myanmar language is difficult to gather due to the multiple encodings of the language.

Motivated by this challenge, we participated in the 2019 edition of the competition on Myanmar-English, organized by the Workshop on Asian Translation. This paper describes our submission, which achieved the highest human evaluation and BLEU score (Papineni et al., 2002) in the competition.

Following common practice in the field, we used back-translation (Sennrich et al., 2015) to leverage target side monolingual data. However, the domain of Myanmar monolingual data is very different from the test domain, which is English originating news (Shen et al., 2019). Since this may hamper the performance of back-translation, we also explored methods that leverage monolingual data on the source side, which is in-domain with the test set when translating from English to Myanmar. We investigated the use of self-training (Yarowski, 1995; Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019) which augments the original parallel data with synthetic data where sources are taken from the original source monolingual dataset and targets are produced by the current machine translation system. We show that self-training and back-translation are often complementary to each other and yield additional improvements when applied in an iterative fashion.

In fact, back-translation and self-training can also be applied when learning from the parallel dataset alone, greatly improving performance over the baseline using the original bitext data. We also report further improvements by swapping beam search decoding with noisy channel re-ranking (Yee et al., 2019) and by ensembling.

We will start by discussing the data preparation process in §2, followed by our model details in §3 and results in §4. We conclude with some final remarks in §5. In Appendix A we report training details and describe the methods that have not proved useful for this task in Appendix B.

[*] Equal contribution.

Proceedings of the 6th Workshop on Asian Translation, pages 112–122
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

2 Data

In this section, we describe the data we used for training and the pre-processing we applied.

2.1 Parallel Data

The parallel data was provided by the organizers of the competition and consists of two datasets. The first dataset is the Asian Language Treebank (ALT) corpus (Thu et al., 2016; Ding et al., 2018, 2019) which consists of 18,088 training sentences, 1,000 validation sentences and 1,018 test sentences from English originating news articles. In this dataset, there is a space character separating each Myanmar morpheme (Thu et al., 2016).

The second dataset is the UCSY dataset[1] which contains 204,539 sentences from various domains, including news articles and textbooks. The originating language of these sentences is not specified. Unlike the ALT dataset, Myanmar text in the UCSY dataset is not segmented and contains very little spacing as it is typical in this language.

The organizers of the competition evaluate submitted systems on the ALT test set.

We denote the parallel dataset by $\mathcal{P} = \{X, Y\}$.

2.2 Monolingual Data

We gather English monolingual data by taking a subset of the 2018 Newscrawl dataset provided by WMT (Barrault et al., 2019), which contains approximately 79 million unique sentences. We choose Newscrawl data to match the domain of the ALT dataset, which primarily contains news originating from English sources.

For Myanmar language, we take five snapshots of the Commoncrawl dataset and combine them with the raw data from Buck et al. (2014). After de-duplication, this resulted in approximately 28 million unique lines. This data is not restricted to the news domain.

We denote by $\mathcal{M}_\mathcal{S}$ the source monolingual dataset and by $\mathcal{M}_\mathcal{T}$ target monolingual dataset.

2.3 Data Preprocessing

The Myanmar monolingual data we collect from Commoncrawl contains text in both Unicode and Zawgyi encodings. We use the `myanmar-tools`[2] library to classify and convert all Zawgyi text to Unicode. Since text classification is performed at the document level, the corpus is left with many embedded English sentences, which we filter by running the fastText classifier (Joulin et al., 2017) over individual sentences.

We tokenize English text using Moses (Koehn et al., 2007) with aggressive hyphen splitting. We explored multiple approaches for tokenizing Myanmar text, including the provided tokenizer and several open source tools. However, initial experiments showed that leaving the text untokenized yielded the best results. When generating Myanmar translations at inference time, we remove separators introduced by BPE, remove all spaces from the generated text, and then apply the provided tokenizer[3].

Finally, we use SentencePiece (Kudo and Richardson, 2018) to learn a BPE vocabulary of size 10,000 over the combined English and Myanmar parallel text corpus.

3 System Overview

Our architecture is a transformer-based neural machine translation system trained with `fairseq`[4] (Ott et al., 2019). We tuned model hyper-parameters via random search over a range of possible values (see Appendix A for details). We performed early stopping based on perplexity on the ALT validation set, and final model hyper-parameter selection based on the BLEU score on the same validation set. We never used the ALT test set during development, and only used it for the final reporting at submission time.

Next, we describe several enhancements to this baseline model (§3.1) and to the decoding process (§3.2). We also describe several methods for leveraging monolingual data, including our final iterative approach (§3.3).

3.1 Improvements to the Baseline Model

We improve our baseline neural machine translation system with: tagging (Sennrich et al., 2016; Kobus et al., 2016; Caswell et al., 2019), fine-tuning and ensembling.

Tagging: Since our test set comes from the ALT corpus and our training set is composed by sev-

[1] http://lotus.kuee.kyoto-u.ac.jp/WAT/my-en-data/

[2] https://github.com/google/myanmar-tools

[3] `myseg.py` can be found in the parallel dataset file on the page http://lotus.kuee.kyoto-u.ac.jp/WAT/my-en-data/

[4] https://github.com/pytorch/fairseq

eral datasets from different domains, we prepend to the input source sentence a token specifying the domain of the input data. We have a total of four domain tokens, indicating whether the input source sentence comes from the ALT dataset, the UCSY dataset, the source monolingual data or if it is a back-translation of the target monolingual data (see §3.3 for more details).

Fine-tuning: The models submitted for final evaluation have also been fine-tuned to the training set of the ALT dataset, as a way to better adapt to the domain of the test set. Fine-tuning is early-stopped based on BLEU on the validation set.

Ensembling: Finally, since we tune our model hyper-parameters via randomized grid search, we are able to cheaply build an ensemble model from the top k best performing hyper-parameter choices. Ensembling yielded consistent gains of about 1 BLEU point.

3.2 Improvements to Decoding

Neural machine translation systems typically employ beam search decoding at inference time to find the most likely hypothesis for a given source sentence. In this work, we improve upon beam search through noisy-channel reranking (Yee et al., 2019). This approach was a key component of the winning submission in the WMT 2019 news translation shared task for English-German, German-English, English-Russian and Russian-English (Ng et al., 2019).

More specifically, given a source sentence x and a candidate translation y, we compute the following score:

$$\log P(y|x) + \lambda_1 \log P(x|y) + \lambda_2 \log P(y) \quad (1)$$

where $\log P(y|x)$, $\log P(x|y)$ and $\log P(y)$ are the forward model, backward model and language model scores, respectively. This combined score is used to rerank the n-best target hypotheses produced by beam search. In our experiments we set n to 50 and output the highest-scoring hypothesis from this set as our translation. The weights λ_1 and λ_2 are tuned via random search on the validation set. The ranges of values for λ_1 and λ_2 are reported in Appendix A.

Throughout this work we use noisy channel reranking every time we decode, whether it is to generate forward or backward translations or to generate translations from the final model for evaluation purposes.

Model	My→En	En→My
$\mathcal{P}$ — beam	25.1	35.9
$\mathcal{P}$ — reranking	26.3	36.9
$\mathcal{P} \cup \mathcal{M}_\mathcal{T}$, beam — beam	32.2	38.8
$\mathcal{P} \cup \mathcal{M}_\mathcal{T}$, reranking — beam	32.5	38.9
$\mathcal{P} \cup \mathcal{M}_\mathcal{T}$, reranking — reranking	35.2	39.4

Table 1: Effect of noisy channel reranking when evaluating on the validation set. On the left of the ”—” symbol there is the dataset used to train the system and the decoding process used to generate back-translated data (if any). On the right of the ”—” symbol there is the decoding process used to generate hypotheses from the forward model. $\mathcal{P}$ refers to the parallel dataset and $\mathcal{M}_\mathcal{T}$ refers to the target monolingual dataset.

Our language models are also based on the transformer architecture and follow the same setup as Radford et al. (2018). The English language model is trained on the CC-News dataset (Liu et al., 2019) and consists of 12 transformer layers and a total of 124M parameters. The Myanmar language model is first trained on the Commoncrawl monolingual data and then fine-tuned on the Myanmar portion of the ALT parallel training data; it consists of 6 transformer layers and 70M parameters. For our constrained submission, which does not make use of additional data, we trained smaller transformer language models for each language (5 transformer layers, 8M parameters) using each side of the provided parallel corpus. For both directions, we observed gains when applying noisy channel reranking, as shown in Table 1.

3.3 Leveraging Monolingual Data

In this section we describe basic approaches to leverage monolingual data. Notice however that these methods also improve system performance in the absence of additional monolingual data (i.e., by reusing the available parallel data), see §4.1.

We denote by $\overrightarrow{f}$ and $\overleftarrow{g}$ the forward (from source to target) and the backward (from target to source) machine translation systems.

Back-translation (BT) (Sennrich et al., 2015) is an effective data augmentation method leveraging target side monolingual data. To perform back-translation, we first train $\overleftarrow{g}$ on $\{Y, X\}$ and use it to translate $\mathcal{M}_\mathcal{T}$ to produce synthetic source side data, denoted by $\overleftarrow{g}(\mathcal{M}_\mathcal{T})$. We then concatenate the original bitext data $\{X, Y\}$ with the back-translated data $\{\overleftarrow{g}(\mathcal{M}_\mathcal{T}), \mathcal{M}_\mathcal{T}\}$ and train the forward translation model from scratch. We typi-

Model	My→En	En→My
BT	33.1	39.5
ST	33.2	39.9
BT + ST	34.1	40.3

Table 2: Combining BT and ST yields better BLEU score than BT and ST.

cally upsample the original parallel data, with the exact rate tuned together with the other hyper-parameters on the validation set (see Appendix A for the upsample ratio range).

Self-Training (ST) (Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019) instead augments the original parallel dataset $\mathcal{P} = \{X, Y\}$ with synthetic pairs composed by a sentence from the source monolingual dataset with the corresponding forward model translation as target, $\{(\mathcal{M}_\mathcal{S}, \overrightarrow{f}(\mathcal{M}_\mathcal{S}))\}$. The potential advantage of this method is that the source side monolingual data can be more in-domain with the test set, which is the case for the English to Myanmar direction. The shortcoming is that synthetic targets are often incorrect and may deteriorate performance.

Combining BT + ST: Self-training and back-translation are complementary to each other. The former is better when the source monolingual data is in-domain while the latter is better when the target monolingual data is in-domain, relative to the domain of the test set.

In Table 2, we show that these two approaches can be combined and yield better performance than either method individually. Specifically, we combine bitext data together with self-trained and back-translated data, $\{X, Y\} \cup \{\overleftarrow{g}(\mathcal{M}_\mathcal{T}), \mathcal{M}_\mathcal{T}\} \cup \{(\mathcal{M}_\mathcal{S}, \overrightarrow{f}(\mathcal{M}_\mathcal{S}))\}$. As for BT, we upsample the bitext data, concatenate it with the forward and backward translations and train a new forward model from scratch. The upsample ratios for each dataset are tuned via hyper-parameter search on the validation set.

3.3.1 Final Iterative Algorithm

The final algorithm proceeds in rounds as described in Alg. 1. At each round, we are provided with a forward model $\overrightarrow{f}$ and a backward model $\overleftarrow{g}$. The forward model translates source side monolingual data (line 6). This is used as forward-translated data to improve the forward model, and as back-translated data to improve the backward model. Similarly, the backward model is used to back-translate target monolingual data (line 7). This data is then used to improve the forward model via back-translation, but also the backward model via self-training. All these datasets are concatenated and weighted to train new forward and backward models (see lines 8 and 9). At the very last iteration, models are fine-tuned on the ALT training set (line 11 and 12), and either way, the best models from the random search are combined into an ensemble to define the new forward and backward models (line 13 and 14) to be used at the next iteration. This whole process of generation and training then repeats as many times as desired. In our experiments we iterated at most three times.

```
1  Data: Given a parallel dataset $\{X, Y\}$, a source
         monolingual dataset $\mathcal{M}_\mathcal{S}$ and a target
         monolingual dataset $\mathcal{M}_\mathcal{T}$;
2  Given an initial forward model $\overrightarrow{f}$ and backward model
     $\overleftarrow{g}$ trained on $\{X, Y\}$;
3  Let $N$ be the number of hyper-parameter configurations
     evaluated during random search;
4  Let $k$ be the number of models used in the ensemble;
5  for $t$ in $[1 \dots T]$ do
6      forward-translated data: $\mathcal{F} \longleftarrow \overrightarrow{f}(\mathcal{M}_\mathcal{S})$;
7      back-translated data: $\mathcal{B} \longleftarrow \overleftarrow{g}(\mathcal{M}_\mathcal{T})$;
8      $\{\overrightarrow{f}_i\}_{i=1 \dots N} \longleftarrow$ random search using:
         $\{X, Y\} \cup \{\mathcal{M}_\mathcal{S}, \mathcal{F}\} \cup \{\mathcal{B}, \mathcal{M}_\mathcal{T}\}$;
9      $\{\overleftarrow{g}_i\}_{i=1 \dots N} \longleftarrow$ random search using:
         $\{Y, X\} \cup \{\mathcal{F}, \mathcal{M}_\mathcal{S}\} \cup \{\mathcal{M}_\mathcal{T}, \mathcal{B}\}$;
10     if $t == T$ then
11         Fine-tune $\{\overrightarrow{f}_i\}_{i=1 \dots N}$ and $\{\overleftarrow{g}_i\}_{i=1 \dots N}$ on
             the in-domain ALT dataset;
       end
12     $\overrightarrow{f} \longleftarrow$ ensemble of top k best models from
         $\{\overrightarrow{f}_i\}_{i=1 \dots N}$;
13     $\overleftarrow{g} \longleftarrow$ ensemble of top k best models from
         $\{\overleftarrow{g}_i\}_{i=1 \dots N}$;
   end
   Result: Forward MT system $\overrightarrow{f}$ and backward MT
           system $\overleftarrow{g}$
```

Algorithm 1: Iterative Learning Algorithm

4 Results

In this section we report validation BLEU scores for the intermediate iterations and ablations, and test BLEU scores only for our final submission. Details of the model architecture, data processing and optimization algorithm are reported in Appendix A.

Our baseline system is trained on the provided parallel datasets with the modeling extensions described in §3.1. According to our hyper-parameter search, the optimal upsampling ratio of the smaller in-domain ALT dataset is three and the best for-

	Description	My → En	En → My
1	Baseline (single)	23.3	34.9
2	Baseline (ensemble)	25.1	35.9
3	2 + reranking	26.3	36.9
4	3 + ST	26.4	38.2
5	3 + BT	26.5	36.9
6	3 + (ST + BT)	27.0	38.1

Table 3: BLEU scores of systems trained only on the provided parallel datasets.

ward and backward model have 5 encoder and 5 decoder transformer layers, where the number of attention heads, embedding dimension and inner-layer dimension are 4, 512, 2048, respectively. Each single model is trained on 4 Volta GPUs for 1.4 hours. We refer to this model as the ”Baseline” in our result tables.

4.1 System Trained on Parallel Data Only

We submitted a machine translation system that only uses the provided ALT and UCSY parallel datasets, without any additional monolingual data, results are reported in Tab. 3. The baseline system achieves 23.3 BLEU points for My→En and 34.9 for En→My . Ensembling 5 models yields +1.8 BLEU points gain for My→En and +1.0 point for En→My . To apply noisy channel reranking, we train language models *using data from the ALT and UCSY training set*. The language model architectures are the same for both languages, each has 5 transformer layers, 4 attention heads, 256 embedding dimensions and 512 inner-layer dimensions. Noisy channel ranking yields a gain of +1.2 BLEU points for My→En and +1.0 points for En→My on top of the ensemble models.

To further improve generalization, we also translated the source and target portion of the parallel dataset using the baseline system in order to collect forward-translations of source sentences and back-translations of target sentences. Based on our grid search, we then train a different model architecture than the baseline system, consisting of 4 layers in encoder and decoder, 8 attention heads, 512 embedding dimensions and 2048 inner-layer dimensions. Each model is trained on 4 Volta GPUs for 2.8 hours. In this case, we train only for one iteration and we ensemble 5 models for each direction followed by reranking.

By applying back-translation and self-training to the parallel data we obtain an additional gain of +0.7 points for My→En and +1.2 points for En→My over the baseline model. We also find that combining back-translation and self-training is beneficial for My→En direction, where we attain an increase of +0.5 BLEU compared to applying each method individually. The final BLEU scores on test set are 26.8 for My→En and 36.8 for En→My .

Description	My → En	En → My
Baseline (ensemble)	25.1	35.9
+ reranking	27.7	36.9
+ iter. 1 of ST + BT	35.5	40.1
+ iter. 2 of ST + BT	36.9	40.4
+ iter. 3 of ST + BT	37.9	40.6

Table 4: BLEU scores of systems trained using additional monolingual data.

4.2 System Using Also Monolingual Data

The results using additional monolingual data are reported in Tab. 4. Starting from the ensemble baseline of the previous section, noisy channel reranking now yields a bigger gain for My→En , +2.64 points, since the language model is now trained on much more in-domain target monolingual data.

Using the ensemble and the additional monolingual data, we apply back-translation and self-training for three iterations. For each iteration, we use the best model from the previous iteration to translate monolingual data with noisy channel reranking. As before, we combine the original parallel data with the two synthetic datasets, and train models from random initialization. We search over hyper-parameters controlling the model architecture whenever we add more monolingual data.

At the first iteration we back-translate 18M English sentences from Newscrawl and 23M Myanmar sentences from Commoncrawl. The best model architecture has 6 layers in the encoder and decoder, where the number of attention heads, embedding dimension and inner-layer dimension are 1024, 4096, 8, respectively. Each model is trained on 4 Volta GPUs for 17 hours. Ensembling two models for My→En and three models for En→My strikes a good trade-off between translation quality and decoding efficiency to generate data for the next iteration. The re-ranked

Description (My → En)	BLEU	Adequacy
FBAI	38.6	4.4
Team1	30.2	4.0
FBAI	26.8	-
Team2	24.8	2.8
Team3	19.6	1.3
Team4	18.5	-
Team5	14.9	-
Team6	10.7	-

Table 5: My→En leaderboard[5]. The values are BLEU score (second column) and Adequacy scores (third column). Rows highlighted in yellow identify systems that make use of additional monolingual data. Our system is tagged as FBAI.

ensemble improves by +7.78 BLEU points for My→En compared to best supervised model, and +3.18 points for En→My .

At the second iteration, we use the same amount of monolingual data of iteration 1 and repeat the same exact process. The model architecture is the same as in the first iteration. We ensemble two models for My→En and use a single model for En→My . We further improve upon the previous iteration by +1.41 points for My→En and +0.27 points for En→My .

At the third and last iteration, we use more monolingual data for both languages, 28M Myanmar sentences and 79M English sentences. We found beneficial (Ng et al., 2019) at this iteration to increase FFN dimension to 8192 and the number of heads to 16. Each model is trained on 8 Volta GPUs for 30 hours. After training models on the parallel and synthetic datasets, we fine-tune each of them on the ALT training set, followed by ensembling. We ensemble 5 models for both directions and apply noisy channel re-ranking as our final submission. Compared to iteration 2 models, the final models yield +0.94 points gain for My→En and +0.26 points for En→My . The BLEU scores of this system on the test set are 38.59 for My→En and 39.25 for En→My .

4.3 Final Evaluation

Tables 5 and 6 report the leaderboard results provided by the organizers of the competition. For each direction, they selected the best system of the four teams that scored the best according to BLEU, and they performed a JPO adequacy human evaluation (Nakazawa et al., 2018). These evaluations are conducted by professional translations who assign a score between 1 and 5 to each translation based on its adequacy. A score equal to 5 points means that all the important information is correctly reported while a score equal to 1 point means that almost all the important information is missing or incorrect.

Description (En → My)	BLEU	Adequacy
FBAI	39.3	3.9
FBAI	36.8	-
Team A	31.3	2.4
Team B	30.8	2.7
Team C	30.8	-
Team D	28.2	-
Team F	25.9	-
Team G	22.5	-
Team H	20.9	1.1
Team I	19.9	-

Table 6: En→My leaderboard[6]. The values are BLEU score (second column) and Adequacy scores (third column). Rows highlighted in yellow identify systems that make use of additional monolingual data. Our system is tagged as FBAI.

First, we observe that our system achieves the best BLEU and adequacy score in both directions, with a gain of more than 8 BLEU points over the second best entry for both directions. The average adequacy score is 0.4 point and 1.2 point higher than the second best entry for My→En and En→My , respectively. Among the rated sentences, more than 30% of sentences translated by our system are rated with 5 points in En→My , compared to 6.3% of the second best system. For My→En , 48% of our translated sentences are rated with 5 points while the second best system has only 24.5%. See Fig 1 for the percentage of each score obtained by the best systems which participated in the competition.

Second, our submission which does not use additional monolingual data is even stronger than all the other submissions in En→My in terms of BLEU score, including those that do make use of additional monolingual data (see second row of Tab. 6).

If we consider submissions that only use the provided parallel data (see rows that are not highlighted), our submission improves upon the sec-

[5] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=70&o=4

[6] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=71&o=9

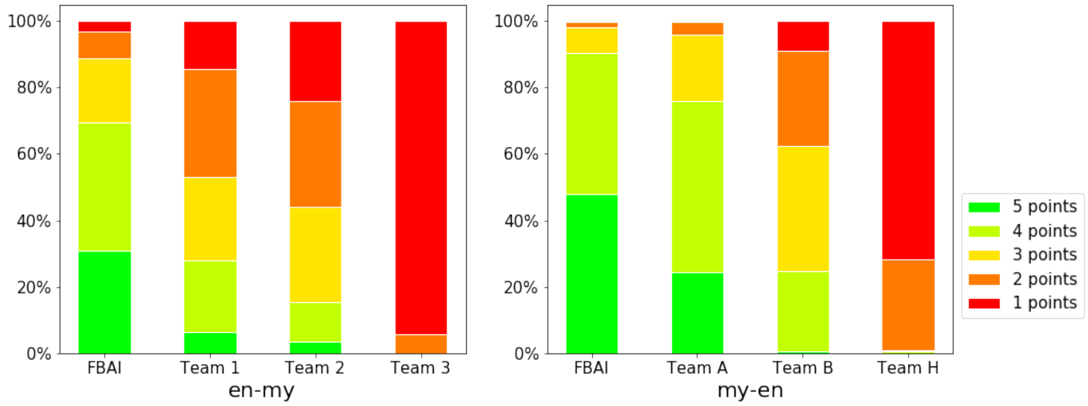

Figure 1: Percentage of each adequacy score obtained by the best systems which participated in the competition. Our system is tagged as FBAI.

ond best system by 7.2 BLEU in My→En and 10.9 BLEU in En→My . This suggests that our baseline system is very strong and that applying ST and BT to the parallel dataset is a good way to build even stronger baselines, as demonstrated also in Tab. 3.

Finally, the gains brought by monolingual datasets is striking only in My→En (+11.8 BLEU points in My→En compared to only +2.5 BLEU points in En→My , for our submissions). The reason is because the ALT test set originates from English news and the target English monolingual data is high quality and in-domain with the test set. Moreover, the source originating Myanmar sentences are translationese of English news sentences, a setting which is particularly favorable to BT. Instead, Myanmar monolingual data is out-of-domain and noisy which makes BT much less effective. ST helps improving BT performance as shown in Tab. 2 but the gains are still limited.

5 Conclusion

We described the approach we used in our submission to the WAT 2019 Myanmar-English machine translation competition. Our approach achieved the best performance both with and without the use of additional monolingual data. It is based on several methods which we combine together. First, we use back-translation to help regularizing and adapting to the test domain, particularly in the Myanmar to English direction. Second, we use self-training as a way to better leverage in-domain source-side monolingual data, particularly in the English to Myanmar direction. Third, given the complementary nature of these two approaches we combined them in an iterative fashion. Fourth, we improve decoding by using noisy-channel reranking and ensembling.

We surmise that there is still quite some room for improvement by better leveraging noisy parallel data resources, by better combining together these different sources of additional data, and by designing better approaches to leverage source side monolingual data.

Acknowledgements

The Authors wish to thank Sergey Edunov for sharing precious insights about his experience participating in WMT competitions and Htet Linn for feedback on how spacing is used in Burmese and for checking a handful of translations during early development.

References

Mikel Artetxe and Holger Schwenk. 2018. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *arXiv preprint arXiv:1812.10464*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della-Pietra, Vincent J. Della-Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the common crawl. In *Proceedings of the Language Resources and Evaluation Conference*, Reykjavik, Iceland.

Christian Buck and Philipp Koehn. 2016. Quick and reliable document alignment via tf/idf-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. *arXiv preprint arXiv:1906.06442*.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Chenchen Ding, Hnin Thu Zar Aye, Win Pa Pa, Khin Thandar Nwet, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. 2019. Towards Burmese (Myanmar) morphological analysis: Syllable-based tokenization and part-of-speech tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):5.

Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. 2018. NOVA: A feasible and flexible annotation system for joint tokenization and part-of-speech tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(2):17.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. In *arXiv:1803.05567*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv:1909.13788*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan M. Pino. 2019. Findings of the wmt 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy. 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. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Toshiaki Nakazawa, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Nobushige Doi, Yusuke Oda, Anoop Kunchukuttan, Shantipriya Parida, Ondej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th workshop on Asian translation. In *Proceedings of the 6th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Toshiaki Nakazawa, Katsuhito Sudoh, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, and Sadao Kurohashi. 2018. Overview of the

5th workshop on Asian translation. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine. In *Proceedings of NAACL-HLT*.

Jiajun Shen, Peng-Jen Chen, Matt Le, Junxian He, Jiatao Gu, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. The source-target domain mismatch problem in machine translation. *arXiv:1909.13151*.

Ye Kyaw Thu, Win Pa Pa, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Introducing the asian language treebank (alt). In *LREC*.

Nicola Ueffing. 2006. Using monolingual source-language data to improve mt performance. In *IWSLT*.

P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*.

David Yarowski. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Annual Meeting of the Association for Computational Linguistics*.

Kyra Yee, Nathan Ng, Yann N. Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv:1908.05731*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Empirical Methods in Natural Language Processing*.

A Hyper-Parameter Search

In this section we report the set of hyper-parameters and range of values that we used in our random hyper-parameter search. For each experiment we searched using $N = 30$ hyper-parameter configurations.

Notice that the actual range of hyper-parameters searched in each experiment may be smaller than reported below; for instance, if a model shows signs of overfitting we may search up to 5 layers as opposed to 6 at the next iteration.

- Layers: $\{4, 5, 6\}$
- Embedding dim: $\{128, 256, 512, 1024\}$
- FFN dim: $\{128, 256, 512, 1024, 2048, 4096, 8192\}$
- Attention heads: $\{1, 2, 4, 8, 16\}$
- Dropout: $\{0.1, 0.2, 0.3, 0.4, 0.5\}$
- Batch size (number of tokens): $\{1, 2, 4, 8, 12, 16, 24, 32\}$ (multiply by 16000)
- Label smoothing: $\{0.1, 0.2, 0.3\}$
- Learning rate: $\{1, 3, 5, 7, 10, 30, 50, 100, 300, 500\}$ (multiply by 1e-4)
- Seed: $\{1, 2, 3, ... , 30\}$
- Data upsampling ratio
  - bitext: $\{1, 2, 3, 4, 6, 8, 12, 16, 20, 32, 40, 64\}$
  - forward-translated: $\{1, 2, 3, 4, 6, 8, 9\}$
  - back-translated: $\{1, 2, 3, 4, 6, 8, 9\}$

When applying noisy-channel reranking, we tune the hyper-parameters λ_1 and λ_2 on the validation set. The ranges of the two hyper-parameters are between 0 and 3.

B Things We Tried But Did Not Use

This section details attempts that did not significantly improve the overall performance of our translation system and which were therefore left out of the final system.

B.1 Out-of-domain parallel data

Similarly to Guzmán et al. (2019) we added out-of-domain parallel data from various sources of the OPUS repository[7], namely GNOME/Ubuntu, QED and GlobalVoices. This provides an additional 38,459 sentence pairs. We also considered two versions of Bible translations from the bible-corpus[8] resulting in additional 61,843 sentence pairs. Adding this data improved the baseline system by +0.17 BLEU for My→En and +0.26 BLEU for En→My .

B.2 Pre-training

We pre-trained our translation system using a cross-lingual language modeling task (Lample and Conneau, 2019) as well as a Denoising Auto-Encoding (DAE) task (Vincent et al., 2008). They both did not provide significant improvements; in the following, we report our results using DAE.

In this setting, we have a single encoder-decoder model which takes a batch of monolingual data, encodes it with the model's encoder, prepends the encoded representation with a language-specific token, and then tries to reconstruct the original input using the model's decoder. Additionally, the source sentences are corrupted using three different types of noise: word dropping, word blanking, and word swapping (Lample et al., 2018a,b). The goal is to encourage the model to learn some kind of common representation for both languages.

We found some gains, particularly for the En→My direction, however, doing backtranslation on top of DAE pretraining did worse or did not improve compared to backtranslation without DAE pretraining. For this reason, we decided to leave this technique out of our final system.

B.3 PBSMT

We also train a phrase based system using Moses with a default setting. We preprocessed the data using moses tokenizer for English sentences. For Myanmar sentences, we use BPE instead. We train a count-based 5-gram English and Myanmar language models on the monolingual data we collect. We tune the system using MERT on the ALT validation set. However, the phrase based system does not perform as good as our NMT baseline.

[7] http://opus.nlpl.eu/

[8] https://github.com/christos-c/bible-corpus/

The phrase based system we train on the parallel data only yields 10.98 BLEU for My→En and 21.89 BLEU for En→My , which are 12.32 and 13.05 BLEU points lower than our supervised single NMT model.

B.4 Weak Supervision

For augmenting the original training data with a noisy set of parallel sentences, we mine bitexts from Commoncrawl. This is achieved by first aligning the webpages in English and Myanmar and then extracting parallel sentences from them. To align webpages, we perform sentence alignment using the IBM1 sentence alignment algorithm (Brown et al., 1993), trained on the provided parallel data to obtain bilingual dictionaries from English to Myanmar and Myanmar to English. Using these dictionaries, unigram-based Myanmar translations are added to the English web documents and Myanmar translations are added to the English documents. The similarity score of a document pair a and b is computed as:

$$sim(a, b) = Lev(url_a, url_b) \times Jaccard(a, b) \tag{2}$$

where $Lev(url_a, url_b)$ is the Levenshtein similarity between the url_a and url_b and $Jaccard(a, b)$ is the Jaccard similarity between documents a and b. Finally, a one-to-one matching between English and Myanmar documents is enforced by applying a greedy bipartite matching algorithm as described in Buck and Koehn (2016). The set of matched aligned documents is then mined for parallel bitexts.

We align sentences within two comparable webpages by following the methods outlined in the parallel corpus filtering shared task for low-resource languages (Koehn et al., 2019). One of the best performing methods for this task used the LASER model (Artetxe and Schwenk, 2018) to gauge similarity between sentence pairs (Chaudhary et al., 2019). Since the open-source LASER model is only trained with 2,000 Myanmar-English bitexts, we retrained the model using the provided UCSY and ALT corpora. For tuning, we use similarity error on the ALT validation dataset and observe that the model performs rather poorly as the available training data was substantially lower than the original setup.