Combination of Statistical and Neural Machine Translation for Myanmar–English

Benjamin Marie **Atsushi Fujita** **Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{bmarie, atsushi.fujita,eiichiro.sumita}@nict.go.jp

Abstract

This paper presents the NICT's machine translation system combining neural and statistical machine translation for the WAT2018 Myanmar–English translation task. For both translation directions, we built state-of-the-art statistical (SMT) and neural (NMT) machine translation systems and combined them to improve translation quality. Our NMT systems were trained with the Transformer architecture using the provided parallel data. Our systems combining SMT and NMT are ranked first for this task according to BLEU. This paper also describes the impact of using a small quantity of back-translated monolingual data.

1 Introduction

This paper describes neural (NMT) and statistical machine translation systems (SMT) built for the participation of the National Institute of Information and Communications Technology (NICT) to the WAT2018 Myanmar–English translation task (Nakazawa et al., 2018).[1] We present systems built using only the parallel data provided by the organizers. For contrastive experiments, we also present systems that use monolingual data not provided by the organizers. For both translation directions, we trained NMT and SMT systems, and combined them through n-best list reranking using several informative features (Marie and Fujita, 2018). This simple combination method achieved the best results among the submitted MT systems for this task according to BLEU (Papineni et al., 2002). We also show that the use of monolingual data can dramatically improve translation quality.

The remainder of this paper is organized as follows. In Section 2, we introduce the data preprocessing. In Section 3, we describe the details of our NMT and SMT systems. The back-translation of monolingual data used by some of our systems is described in Section 4. Then, the combination of NMT and SMT is described in Section 5. Empirical results achieved by our systems are showed and analyzed in Section 6, and Section 7 concludes this paper.

2 Data preprocessing

To train our systems, we used all the bilingual data provided by the organizers. The provided bilingual data comprise two different corpora: the training data provided by the ALT project[2] and additional training data, UCSY corpus, constructed by the University of Computer Studies, Yangon (UCSY). Since no monolingual corpus was provided, we submitted for human evaluation the outputs of systems trained only on the parallel data. For contrastive experiments, we also built systems using monolingual data. For English, we used the monolingual corpora provided by the WMT18 shared News Translation Task. As for Myanmar, we experimented with two monolingual corpora: Myanmar Wikipedia and Myanmar CommonCrawl. The Wikipedia corpus was created from the entire Myanmar Wikipedia dumped on 2017/06/01. The CommonCrawl cor-

[1] The team ID of our participation is "NICT-4".

[2] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/index.html

pus consists of sentences in the Myanmar language[3] from the first quarter of the CommonCrawl data crawled during April 2018. These Myanmar monolingual corpora, especially the CommonCrawl corpus crawled from various websites, contain a large portion of useless data. For instance, many lines in the corpus are made of long sequences of numbers or punctuation marks. For cleaning, we decided to remove lines in both corpora that fulfill at least one of the following conditions:

- more than 25% of its tokens are numbers or punctuation marks.
- contains less than 4 tokens
- contains more than 80 tokens

For contrastive experiments, we also prepared CommonCrawl and Wikipedia corpora for English and cleaned them in the same manner. For the CommonCrawl corpus, we sampled 2M lines from the entire CommonCrawl corpus provided by WMT18, while for the Wikipedia corpus we sampled 1M lines from the entire dump of the English Wikipedia of 2017/06/01.

We tokenized and truecased English data respectively with the tokenizer and truecaser of Moses (Koehn et al., 2007). The truecaser was trained on the English side of the parallel data. Truecasing was performed on all the tokenized data. For Myanmar, the provided bilingual data were already tokenized into writing units and Romanized.[4] However, we were not able to take advantage of this preprocessing and chose to reverse it and tokenize the bilingual and monolingual data by ourselves with an in-house tokenizer. We did not apply truecasing on the Myanmar data. Note that for the en→my task, the outputs generated in the Myanmar language must be processed as done by the organizers before submission.

For cleaning bilingual data, we only applied the Moses script `clean-n-corpus.perl` to remove lines in the parallel data containing more than 80 tokens and escaped characters forbidden by Moses. Note that we did not perform any punctuation normalization.

Data set	#sent. pairs	#tokens (my)	#tokens (en)
Train	226.6k	4.4M	3.4M
Development	993	37.8k	25.4k
Test	1,007	38.8k	25.9k

Table 1: Statistics of our preprocessed parallel data.

Corpus	#lines	#tokens
WMT (English)	338.7M	7.5B
Wikipedia (English)	1M	11.8M
CommonCrawl (English)	2M	44.5M
Wikipedia (Myanmar)	1.2M	13.0M
CommonCrawl (Myanmar)	1.5M	56.6M

Table 2: Statistics of our preprocessed monolingual data.

To tune/validate and evaluate our systems, we used the official development and test sets chosen for the task: the ALT test data consisting of translations of English texts sampled from English Wikinews.

Tables 1 and 2 present the statistics of the parallel and monolingual data, respectively, after preprocessing.

3 MT Systems

3.1 NMT

To build competitive NMT systems, we chose to rely on the Transformer architecture (Vaswani et al., 2017) since it has been shown to outperform, in quality and efficiency, the two other mainstream architectures for NMT known as deep recurrent neural network (deep RNN) and convolutional neural network (CNN). We chose Marian[5] (Junczys-Dowmunt et al., 2018) to train and evaluate our NMT systems since it supports state-of-the-art features and is one of the fastest NMT framework publicly available. In order to limit the size of the vocabulary of the NMT models, we further segmented tokens in the parallel data into sub-word units via byte pair encoding (BPE) (Sennrich et al., 2016b) using 8k operations for both languages.[6] All

[3] We used `fasttext` and its pretrained models for language identification: `https://fasttext.cc/blog/2017/10/02/blog-post.html`

[4] The preprocessing was performed, and can be reversed with this script: `http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/myan2roma.py`

[5] `https://marian-nmt.github.io/`, version 1.6

[6] The number of operations was chosen among 8k,16k,32k according to the best BLEU score obtained on the development set. We observed around 2 BLEU points of difference between

our NMT systems were consistently trained on 4 GPUs,[7] with the following parameters for Marian:

```
--type transformer --max-length 80
--mini-batch-fit --valid-freq 5000
--save-freq 5000 --workspace 10000
--disp-freq 500 --beam-size 12
--normalize 1 --valid-mini-batch
16 --overwrite --early-stopping
5 --cost-type ce-mean-words
--valid-metrics ce-mean-words
perplexity translation --keep-best
--enc-depth 4 --dec-depth 4
--transformer-dropout 0.1
--learn-rate 0.001 --dropout-src
0.1 --dropout-trg 0.1 --lr-warmup
16000 --lr-decay-inv-sqrt 16000
--lr-report --label-smoothing 0.1
--devices 0 1 2 3 --dim-vocabs
8000 8000 --optimizer-params
0.9 0.98 1e-09 --clip-norm 5
--sync-sgd --exponential-smoothing.
```

3.2 SMT

We also trained SMT systems using Moses. Word alignments and phrase tables were trained on the tokenized parallel data using `mgiza`. Source-to-target and target-to-source word alignments were symmetrized with the `grow-diag-final-and` heuristic. We trained phrase-based SMT models and `MSLR` (monotone, swap, discontinuous-left, discontinuous-right) lexicalized reordering models. We also used the default distortion limit of 6. We trained two 4-gram language models, one on the WMT monolingual data for English, and on the Wikipedia data for Myanmar, concatenated to the target side of the parallel data, and another one on the target side of the parallel data only, using `LMPLZ` (Heafield et al., 2013). To tune the SMT model weights, we used `kb-mira` (Cherry and Foster, 2012) and selected the weights giving the best BLEU score for the development data during 15 iterations.

Back-translation	# backtr.	my→en	en→my
None (baseline)	0	19.0	27.6
Wikipedia	300k	20.1	29.0*
	1M	23.3	27.9
CommonCrawl	300k	23.2	22.5
	1M	25.1*	17.6

Table 3: BLEU scores for our NMT systems on the official test set of the tasks. The “Corpus” column denotes the origin of the back-translated data and the “#backtr.” column denotes the number of back-translated sentences mixed with the bilingual data for training. “$*$” indicates the best configuration for each task used for experiments with back-translated data presented in Section 6.

4 Back-Translation of Monolingual Data for NMT

Parallel data for training NMT can be augmented with synthetic parallel data, generated through a so-called back-translation, to significantly improve translation quality (Sennrich et al., 2016a). To perform back-translation, we used an NMT system, trained on the parallel data provided by the organizers, to translate target monolingual sentences into the source language. Then, the back-translated sentences were simply mixed with the original parallel data to train from scratch a new source-to-target NMT system. However, since our monolingual data were not provided by the organizers, we did not use back-translation to generate our primary submissions for human evaluation.

We compared systems using back-translations of either Wikipedia or CommonCrawl. We also experimented using 300k or 1M back-translated sentences for training. The results are reported in Table 3. For my→en, the use of back-translations significantly improved the translation quality, especially in the configurations where we used 1M back-translated sentences. Since the improvements are significantly larger with CommonCrawl data, we chose this corpus for additional experiments presented in Section 6. For en→my, using only 300k back-translated sentences from Wikipedia led to the best results but with only 1.6 BLEU points of improvements over the baseline system, which did not use back-translated data. Using CommonCrawl corpus systematically decreased the translation quality with up to 10 BLEU points from the baseline system. We

8k and 32k.

[7] NVIDIA® Tesla® P100 16Gb.

Feature	Description
L2R (4)	Scores given by each of the 4 left-to-right Marian models
R2L (1)	Score given by each the right-to-left Marian model
LEX (4)	Sentence-level translation probabilities, for both translation directions
LM (1 or 2)	Scores given by the language models used by the Moses baseline systems
LEN (2)	Difference between the length of the source sentence and the length of the translation hypothesis, and its absolute value

Table 4: Set of features used by our reranking systems. The "Feature" column refers to the same feature name used in Marie and Fujita (2018). The numbers between parentheses indicate the number of scores in each feature set.

speculate that our Myanmar CommonCrawl corpus is too noisy to be useful to train an NMT model.

5 Combination of NMT and SMT

Our primary submissions for the task were the results of a simple combination of NMT and SMT. As demonstrated by Marie and Fujita (2018), and despite the simplicity of the method used, combining NMT and SMT makes MT more robust and can significantly improve translation quality, even though SMT greatly underperforms NMT. Following Marie and Fujita (2018), our combination of NMT and SMT works as follows.

5.1 Generation of n-best Lists

We first independently generated the 100-best translation hypotheses with 4 NMT models, independently trained, and also with the ensemble of these 4 NMT models. We also generated 100-best translation hypotheses with our SMT system. We then merged all these 6 lists generated by different systems, without removing duplicated hypotheses, which resulted in a list of 600 diverse translation hypotheses for each source sentence. Finally, we rescored all the hypotheses in the list with a reranking framework using features to better model the fluency and the adequacy of each hypothesis. This method can find a better hypothesis in these merged n-best lists than the one-best hypothesis originated by the individual systems.

5.2 Reranking Framework and Features

We chose `kb-mira` as a rescoring framework and used a subset of the features proposed in Marie and Fujita (2018). All the following features we used are described in details by Marie and Fujita (2018). As listed in Table 4, it includes the scores given by 4 left-to-right NMT models independently trained. We also used as features the scores given by one right-to-left NMT model. We computed sentence-level translation probabilities using the lexical translation probabilities learned by `mgiza` during the training of our SMT systems. The two language models trained for SMT for each translation direction were also used to score the n-best translation hypotheses. We used only one language model trained on the target side of the parallel data for our primary submission. To account for hypotheses length, we added the difference, and its absolute value, between the number of tokens in the translation hypothesis and the source sentence.

The reranking framework was trained on n-best lists generated by decoding of the development data that we used to validate the training of NMT systems and to tune the weights of SMT models.

6 Results

Our results are presented in Table 5. As expected, SMT performed significantly worse than NMT for both translation directions (#1 vs #3), especially for my→en with 9.5 BLEU points difference. Introducing a larger language models trained on monolingual data improved translation quality (#1 vs #2), especially for my→en, owing to the very large monolingual data. Even though our monolingual data for en→my were significantly smaller and noisier, we could still obtain an improvement of 0.5 BLEU points.

For NMT without back-translation, ensembling 4 models for decoding was very effective with 3.0 and 2.0 BLEU points of improvements (#3 vs #4), respectively for my→en and en→my. As discussed in Section 4, introducing back-translation significantly improved the translation quality.

ID	System	my→en	en→my
1.	Moses	9.5	23.1
2.	Moses w/ big LM	11.4	23.6
3.	Marian single	19.0	27.6
4.	Marian ensemble of 4	22.0	29.6
5.	Marian single w/ backtr.	25.1	29.0
6.	Marian ensemble of 4 w/ backtr.	27.8	31.8
7.	Moses (#1) + Marian ensemble of 4 (#4)	22.5	30.5
8.	Moses w/ big LM (#2) + Marian ensemble of 4 w/ backtr. (#6)	29.1	32.3

Table 5: BLEU scores for our MT systems on the official test set of the tasks. “big LM” denotes the use of a language model trained on large monolingual data. “backtr” denotes the use or not of back-translated monolingual data. “Moses + Marian” denotes our n-best list combination described in Section 5: #7 combines systems trained only on the parallel data provided by the organizers, while #8 does so the best SMT and the best NMT systems realized using additional monolingual data. We submitted systems #4 and #7 for human evaluation.

Combining SMT and NMT, without using large monolingual data, slightly improved the translation quality (#4 vs #7) by 0.5 and 0.9 BLEU points for my→en and en→my, respectively. Combining “Moses big LM” and “Marian ensemble of 4 w/ backtr.” further improved translation quality (#6 vs #8) by 1.3 and 0.5 BLEU points, respectively.

While our combination of SMT and NMT (#7) achieved the best BLEU scores among the submitted systems, it consistently underperformed our best NMT system (#4) according to human evaluation. We speculate that this is the consequence of the adoption of some SMT outputs that may be more adequate to the given source sentence but less fluent than the NMT outputs. In future work, we will perform further analysis to better understand the results, given the translation task, the specificities of the Myanmar–English language pair, and the methodology of the human evaluation used for WAT.

7 Conclusion

In this paper, we showed that combining SMT and NMT can further improve the translation quality over a very strong NMT system, even though SMT largely underperforms NMT. Moreover, we showed that the use of monolingual data significantly improved the translation quality for Myanmar–English. In order to allow participants to build state-of-the-art MT systems, we strongly encourage WAT organizers to provide monolingual data for future editions of the workshop. The gap in translation quality between NMT systems that use and do not use monolingual data has been constantly enlarging every year and it is expected to be even more significant in the near future (Edunov et al., 2018).

Acknowledgments

The method used in this participation was developed under the program “Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology” of the Ministry of Internal Affairs and Communications (MIC), Japan.

References

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, August. Association for Computational Linguistics.

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, July. 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, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase-based and neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 111–124, Boston, USA. Association for Machine Translation in the Americas.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Win Pa Pa, Isao Goto, Hideya Mino, Katsuhito Sudoh, and Sadao Kurohashi. 2018. Overview of the 5th workshop on asian translation. In *Proceedings of the 5th Workshop on Asian Translation (WAT2018)*, Hong Kong, China, December.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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, Berlin, Germany, August. 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 *Proceedings of the 30th Neural Information Processing Systems Conference (NIPS)*, pages 5998–6008.