Supervised and Unsupervised Machine Translation for Myanmar-English and Khmer-English

Benjamin Marie **Hour Kaing** **Aye Myat Mon** **Chenchen Ding**
Atsushi Fujita **Masao Utiyama** **Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{bmarie,hour_kaing,ayemyatmon,chenchen.ding}@nict.go.jp
{atsushi.fujita,mutiyama,eiichiro.sumita}@nict.go.jp

Abstract

This paper presents the NICT's supervised and unsupervised machine translation systems for the WAT2019 Myanmar-English and Khmer-English translation tasks. For all the translation directions, we built state-of-the-art supervised neural (NMT) and statistical (SMT) machine translation systems, using monolingual data cleaned and normalized. Our combination of NMT and SMT performed among the best systems for the four translation directions. We also investigated the feasibility of unsupervised machine translation for low-resource and distant language pairs and confirmed observations of previous work showing that unsupervised MT is still largely unable to deal with them.

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) in the WAT2019 (Nakazawa et al., 2019) Myanmar-English (my-en) and Khmer-English (km-en) translation tasks.[1] We present supervised systems built using the parallel data provided by the organizers and external additional monolingual data. For all the translation directions, we trained supervised NMT and SMT systems, and combined them through n-best list reranking using several informative features (Marie and Fujita, 2018a), as in our previous participation to WAT2018 (Marie et al., 2018). This simple combination method achieved the best results among the submitted MT systems for these tasks according to BLEU (Papineni et al., 2002). We also show that the use of monolingual data can dramatically improve translation quality and that an advanced cleaning and normalization of the data further boosts the translation quality. For contrastive experiments, and for investigating the feasibility of unsupervised machine translation (MT) for low-resource distant language pairs, we also present unsupervised MT systems that only use for training the development data provided for these tasks and our monolingual data.

The remainder of this paper is organized as follows. In Section 2, we introduce the data preprocessing, including cleaning and normalization steps. 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. In Section 6, we present our unsupervised MT system. Empirical results achieved by all our systems are showed and analyzed in Section 7. Section 8 concludes this paper.

2 Data preprocessing

To train our systems, we used all the bilingual data provided by the organizers. The provided bilingual data comprises different types of corpora: the training data provided by the ALT project[2] and additional training data. These additional data are the UCSY corpus, constructed by the University of Computer Studies, Yangon (UCSY),[3] for the my-en task, and the ECCC corpus, collected by National Institute of Posts, Telecoms & ICT (NIPTICT)

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

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

[3] Note that this corpus is not the same as last year and has been further cleaned by the organizers.

Proceedings of the 6th Workshop on Asian Translation, pages 68–75

and cleaned by NICT, for the km-en task.

For English, we used the monolingual corpora provided by the WMT18 shared News Translation Task (Bojar et al., 2018). For Khmer, we experimented with a monolingual corpus extracted from Common Crawl.[4] As for Myanmar, we experimented with two monolingual corpora: Myanmar Wikipedia and Myanmar Common Crawl. During our last year's participation in the task, we only observed slight improvements, or even a significant drop of the translation quality with the Common Crawl corpus, when using these Myanmar monolingual corpora that we assumed to be the consequence of the extreme noisiness of the data. This year, we introduce a new cleaning and normalization process (Section 2.1) to better exploit the monolingual data. The Wikipedia corpus was created from the entire Myanmar Wikipedia dumped on 2017/06/01. The Khmer and Myanmar Common Crawl corpora consist of sentences in their respective languages[5] from the first quarter of the Common Crawl data crawled during April 2018 for Myanmar, and April 2019 for Khmer. These monolingual corpora, especially the Common Crawl corpora crawled from various websites, contain a large portion of useless data that necessitates cleaning and normalization as presented in the Sections 2.1 and 2.2.

We tokenized and truecased English data respectively with the tokenizer and truecaser of Moses[6] (Koehn et al., 2007). The truecaser was trained on all our English monolingual data. Truecasing was performed on all the tokenized data. For Myanmar, the provided bilingual data were already tokenized. However, for the sake of consistency with our tokenizer we chose to reverse it and tokenized the bilingual and monolingual data by ourselves with an in-house tokenizer. We did not apply truecasing to the Myanmar data. We performed the same procedure for Khmer.

For cleaning, after pre-processing the Myanmar and Khmer monolingual data as described in the Sections 2.1 and 2.2, we segmented the text into sentences and removed 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 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.

To tune/validate and evaluate our systems, we used the official development and test sets designated for the tasks: 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.

2.1 Cleaning of Myanmar Data

Many lines in the Common Crawl corpus are made of long sequences of numbers and/or punctuation marks, and 80% of Myanmar lines are not written in a standard Unicode format. It also contains foreign languages, such as English, Thai, and Chinese sentences. In the Wikipedia corpus, a standard Unicode format is used but the text is also very noisy. The most common issues in these corpora are spelling errors. From these observations, we applied the following steps for cleanings:

- Encoding normalization
- Noisy sentence removal
- Spelling error correction

First, we used the UCSY encoding converter to convert Zawgyi font to Unicode.[7] Second, we manually removed 22% of noisy sentences in the Common Crawl corpus and 15% of noisy sentences in the Wikipedia corpus.

There are many spelling errors in the corpora. The spell and pronunciation of a word

[4] https://commoncrawl.org/

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

[6] https://github.com/moses-smt/mosesdecoder

[7] This step requires three minutes of computational time for processing one thousand sentences.

Data set	#sent. pairs (#tokens)	
	my-en	km-en
Train	221.1k (my: 4.1M, en: 3.2M)	122.7k (km: 4.1M, en: 3.3M)
Development	1,000 (my: 36,688, en: 25,538)	1,000 (km: 33,604, en: 25,538)
Test	1,018 (my: 37,519, en: 26,236)	1,018 (km: 34,238, en: 26,236)

Table 1: Statistics of our preprocessed parallel data.

Corpus	#lines	#tokens
WMT (English)	338.7M	7.5B
CommonCrawl (English)	2.0M	44.5M
Wikipedia (Myanmar)	268.7k	5.5M
CommonCrawl (Myanmar)	3.0M	67.5M
CommonCrawl (Khmer)	882.9k	30.1M

Table 2: Statistics of our preprocessed monolingual data.

Order	From	To	Graph
1	◌្ + ដ	◌្ + ត	◌
2	◌ំ + ◌	◌ + ◌ំ	◌ាំ
3	◌ុ + ◌ាំ ◌ាំ + ◌ុ ◌ាំ + ◌ឹ	◌ឹ + ◌ាំ	◌ុាំ
4	V + S[S]	S[S] + V	-
5	WS + SS	SS + WS	-

Table 3: Khmer Text normalization rules, where "V" is Vowel, "S" is subscript (subscript sign + a consonant) and [S] refer to one or zero subscript, WS is west subscript, and SS is south subscript.

may lead to misspelling because there are complex orthographic rules and a large gap between the script and the pronunciation in the Myanmar language. One type of spelling errors results in words that do not exist in the Myanmar language. They can be detected easily by a spell checker and a dictionary look-up. Another type of errors happened when the writer uses existing words but wrongly or ambiguously in context. Those errors are difficult to automatically detect as these words exist in a Myanmar dictionary but are incorrect according to the context. There are two types of errors: phonetic errors and context errors. Context error is a subset of phonetic error (e.g., "I saw three trees in the park" as "I saw tree trees in the park").

We performed a dictionary[8] look-up to match the word in the given text with the word in the dictionary. If a word is not there then it is considered as an error. We also measured the Levenshtein distance at the character level to find the closest word in a large Myanmar dictionary. After generating a list of suggestions, we used a bigram language model to select and apply the best correction in context.

2.2 Cleaning of Khmer Data

We clean the Common Crawl corpus for Khmer in two steps, spelling disambiguation and over-tokenization recovery. In our context, over-tokenization refers to dependant characters that should never be separated by a space.

The Khmer corpus is in Unicode format and it is very common for spelling ambiguities where multiple character sequences can represent word with the same graphical representation. We solve this problem by replacing the spelling ambiguities into one form which basically follows the way of Khmer native speakers' spelling. The replacement rules are simply in the order as in Table 3.

As our in-house tokenizer works at character level, over-tokenization is unavoidable when out-of-vocabularies (OOVs) appear. We reverted the over-tokenization by removing spaces as follows:

- before `[U+17B6 - U+17D3]`
- before `.?[U+17CB - U+17CD]`
- before and after `U+17D2`
- after `[U+17A5 U+17A7 U+17AB U+17AD]`

However, recovering from over-tokenization did not result in improvements of translation quality according to BLEU. Consequently, for the sake of simplicity, we did not use this step when building our MT systems.

[8]We used a Myanmar dictionary that contains a list of unique 41,343 Myanmar words from https://github.com/chanmratekoko/Awesome-Myanmar.

```
--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 4 5
6 7 --dim-vocabs 8000 8000
--optimizer-params 0.9 0.98
1e-09 --clip-norm 5 --sync-sgd
--exponential-smoothing
```

Table 4: Parameters of Marian used for training our NMT systems.

3 Supervised MT Systems

3.1 NMT

To build competitive NMT systems, we relied on the Transformer architecture (Vaswani et al., 2017). We chose Marian[9] (Junczys-Dowmunt et al., 2018) to train and evaluate our NMT systems. 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 each language. All our NMT systems were consistently trained on 8 GPUs,[10] with the parameters presented in Table 4.

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, on the Common Crawl corpus for Khmer, 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, 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.

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). We used an NMT system, trained on the parallel data provided by the organizers, to translate target monolingual sentences into the source language. Then, these back-translated sentences were simply mixed with the original parallel data to train from scratch a new source-to-target NMT system.

We back-translated 2M sentences randomly sampled from WMT18 English data for my→en and km→en, our Myanmar Wikipedia corpus for en→my, and our Khmer Common Crawl corpus for en→km.

5 Combination of NMT and SMT

Our primary submissions for the tasks were the results of a simple combination of NMT and SMT. As demonstrated by Marie and Fujita (2018a), and despite the simplicity of the method used, combining NMT and SMT makes MT more robust and can significantly improve translation quality, even when SMT greatly underperforms NMT. Following Marie and Fujita (2018a), 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 7 NMT models, independently trained, and also with the ensemble of these 7 NMT models. We also generated 100-best translation hypotheses with our

[9] https://marian-nmt.github.io/, version 1.7.6
[10] NVIDIA® Tesla® V100 32Gb.

Feature	Description
L2R (7)	Scores given by each of the 7 left-to-right Marian models
LEX (4)	Sentence-level translation probabilities, for both translation directions
LM (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 5: Set of features used by our reranking systems. The "Feature" column refers to the same feature name used in Marie and Fujita (2018a). The numbers between parentheses indicate the number of scores in each feature set.

SMT system. We then merged all these 9 lists generated by different systems, without removing duplicated hypotheses, which resulted in a list of 900 diverse translation hypotheses for each source sentence.

5.2 Reranking Framework and Features

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. We chose `kb-mira` as a rescoring framework and used a subset of the features proposed in Marie and Fujita (2018a). All the following features we used are described in details by Marie and Fujita (2018a). As listed in Table 5, it includes the scores given by 7 left-to-right NMT models independently trained. 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 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 Unsupervised SMT

We also built an SMT system, without any supervision, i.e., using only our monolingual data for training. We chose unsupervised SMT (USMT) over unsupervised NMT (UNMT) since previous work (Lample et al., 2018) has shown that USMT significantly outperforms UNMT for distant languages.

We built USMT systems using a framework similar to the one proposed in Marie and Fujita (2018b). The first step of USMT is the induction of a phrase table from the monolingual corpora. We first collected phrases of up to six tokens from the monolingual corpora[11] using `word2phrase`.[12] As phrases, we also considered all the token types in the corpora. Then, we selected the 300k most frequent phrases in the monolingual corpora to be used for inducing a phrase table. All possible phrase pairs are scored, as in Marie and Fujita (2018b), using bilingual word embeddings, and the 300 target phrases with the highest scores were kept in the phrase table for each source phrase. As a result, the induced phrase table contains a total of 90M (300k×300) phrase pairs. For this induction, bilingual word embeddings of 300 dimensions were obtained using word embeddings trained with `fastText`[13] and aligned in the same space using unsupervised `Vecmap` (Artetxe et al., 2018a). This alignment is the most critical step for unsupervised MT since it is used for initializing the training. It is expected to be extremely difficult for distant languages such as Myanmar, Khmer, and English, as reported by previous work (Søgaard et al., 2018). For each phrase pair, a total of four scores, to be used as features in the phrase tables were computed to mimic phrase-

[11] Since our Myanmar Wikipedia corpus is significantly smaller than the Myanmar Common Crawl corpus, we concatenated both corpora and used the resulting corpus in all the subsequent steps of USMT training.

[12] https://code.google.com/archive/p/word2vec/

[13] https://github.com/facebookresearch/fastText

ID	System	my→en	en→my	km→en	en→km
1.	Moses	10.3	20.5	19.8	40.4
2.	Marian single	15.7	25.2	17.0	37.8
3.	Marian single w/ backtr.	19.1	28.8	24.9	42.9
4.	Marian ensemble of 4 w/ backtr.	22.4	29.7	25.9	43.0
5.	#1 + #4	24.8	31.3	27.5	43.9
6.	Unsupervised SMT	< 1.0	< 1.0	< 1.0	< 1.0

Table 6: Official BLEU scores for our MT systems on the official test set of the tasks. "backtr" denotes the use of back-translated monolingual data. #5 denotes our n-best list combination described in Section 5: a combination of the best SMT and the best NMT systems realized using monolingual data. We submitted systems #5 for human evaluation.

based SMT: forward and backward phrase and lexical translation probabilities. Finally, the phrase table and the language models were plugged into a Moses system that was tuned on the development data using `KB-MIRA`.

We performed four refinement steps to improve the system, using at each step synthetic parallel sentences generated from one third of the monolingual corpus, by the forward and backward translation systems, instead of using only either forward (Marie and Fujita, 2018b) or backward translations (Artetxe et al., 2018b). We report on the performance of the systems obtained after the fourth refinement step.

7 Results

Table 6 presents the results for different versions of our SMT and NMT systems. We can observe that NMT (#2) is significantly better than SMT (#1) for my-en while we can observe the reverse for km-en. Our assumption for explaining this difference is that my-en has a much larger training data while km-en may not have enough to train a better NMT systems. The extreme noisiness of the training data for km-en, that we assessed by a native Khmer speaker, may also explain the large gap between SMT and NMT since it is well-known that SMT is much more robust than NMT when trained on noisy data.

Exploiting monolingual data through back-translation (#3) consistently improves all our NMT systems by a large margin, from 3.4 (my→en) to 7.9 (km→en) BLEU points. This highlights the importance of using monolingual data in low-resource scenarios, even when the NMT system used for generating back-translations deliver a translation of a low quality.

Our results are more contrasted when ensembling 7 NMT models during decoding (#4). While we observe an improvement of 3.3 BLEU points for (my→en), the improvements for the other directions were limited to 1.0 BLEU points or less. Considering the cost of independently training 7 NMT models and the cost of decoding with 7 models, ensembling does not seem to offer a cost-effective solution.

Finally, combining SMT and NMT (#5) provides the best results with improvements over #4 ranging from 0.9 (en→km) to 2.4 BLEU points (my→en).

Our results for unsupervised SMT (#6) follow the same trend as the results presented by Marie et al. (2019) for English-Gujarati and English-Kazakh at WMT19: while unsupervised MT has shown promising results for European languages, it is far from being useful for real-world applications, i.e., truly low-resource distant language pairs. We assume that training useful bilingual weakly-supervised/unsupervised bilingual word embeddings for initializing the system remains one of the main challenges.

8 Conclusion

In this paper, we showed that exploiting cleaned and normalized noisy monolingual data significantly helps in improving the translation quality for my-en and km-en. Furthermore, as in our previous participation in WAT2018, we showed that combining NMT and SMT can further improve the translation quality over a very strong NMT system. In order to allow participants to build state-of-

the-art MT systems, we encourage, even more than last year, WAT organizers to provide monolingual data for future editions of the workshop.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

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. 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. 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. 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 Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018a. 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.

Benjamin Marie and Atsushi Fujita. 2018b. Unsupervised neural machine translation initialized by unsupervised statistical machine translation. *CoRR*, abs/1810.12703.

Benjamin Marie, Atsushi Fujita, and Eiichiro Sumita. 2018. Combination of statistical and neural machine translation for Myanmar-English. 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.

Benjamin Marie, Haipeng Sun, Rui Wang, Kehai Chen, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2019. NICT's unsupervised neural and statistical machine translation systems for the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 294–301, Florence, Italy. Association for Computational Linguistics.

Toshiaki Nakazawa, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Nobushige Doi, Yusuke Oda, Anoop Kunchukuttan, Shantipriya Parida, Ondřej 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.

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. 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. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. 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. Curran Associates, Inc.