Neural Machine Translation of Low-Resource and Similar Languages with Backtranslation

Michael Przystupa
University of British Columbia
<first>.<last>@gmail.com

Muhammad Abdul-Mageed
University of British Columbia
muhammad.mageed@ubc.ca

Abstract

We present our contribution to the WMT19 Similar Language Translation shared task. We investigate the utility of neural machine translation on three low-resource, similar language pairs: Spanish – Portuguese, Czech – Polish, and Hindi – Nepali. Since state-of-the-art neural machine translation systems still require large amounts of bitext, which we do not have for the pairs we consider, we focus primarily on incorporating monolingual data into our models with backtranslation. In our analysis, we found Transformer models to work best on Spanish – Portuguese and Czech – Polish translation, whereas LSTMs with global attention worked best on Hindi – Nepali translation.

1 Introduction

We present our contribution to the WMT 2019 Similar Language Translation shared task, which focused on translation between similar language pairs in low-resource settings (Barrault et al., 2019). Similar languages have advantages that can be exploited when building machine translation systems. In particular, languages that come from the same language family (or that come from related language families) may have in common a multitude of information such as lexical or syntactic structures. This commonality has been exploited in a number of previous works for similar language translation (Haji et al., 2003; Goyal and Lehal, 2009, 2011; Pourdamghani and Knight, 2017).

In this work, we are primarily concerned with neural machine translation (NMT). NMT is a language agnostic framework where language similarities could possibly be exploited to build scalable, state-of-the-art (SOTA) machine translation systems. For example, NMT systems have been used on a number of WMT translation tasks where they enabled highly successful modeling (Bahdanau et al., 2014; Luong et al., 2015; Koehn, 2017; Vaswani et al., 2017; Edunov et al., 2018). A weakness with NMT is its dependence on large bitext corpora. For this reason, researchers have considered ways to mitigate this specific issue.

A prominent approach meant to alleviate need for large parallel data is *backtranslation*. This technique generates synthetic bitext by translating monolingual sentences of the target language into the source language with a pre-existing target-to-source translation system. These noisy source translations are then incorporated to train a new source-to-target MT system (Sennrich et al., 2015a). This approach is instrumental in unsupervised machine translation where authors have shown that, up to a certain amount of bitext, better translation systems can be trained with these unsupervised approaches than supervised methods (Artetxe et al., 2017; Lample et al., 2017, 2018). Backtranslation research has also extended to scenarios of training supervised systems with just synthetic data (Edunov et al., 2018; Marie and Fujita, 2018). Given the success of this approach, it offers a promising avenue to leverage monolingual data for improving translation between similar languages.

Motivated by the success of backtranslation, we focus on leveraging monolingual data to improve NMT systems for similar language pairs. Hence, for our submissions to the shared task, we focus on investigating the effectiveness of synthetic bitext produced with *backtranslation*.

The rest of the paper is organized as follows: We discuss our methods in Section 2, including our NMT models and our decisions for backtranslation. Section 3 is where we describe our analysis of the shared task data. In Section 4, we present our experimental findings, discussing the effectiveness of backtranslation in terms of BLEU

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 224–235

score performance. We conclude in Section 5.

2 Methodology

Here, we outline our approach to improve translation quality for similar languages. This includes description of the two NMT models we considered in our analysis, and our procedure for backtranslating data.

2.1 Model Architectures

Sequence to sequence (seq2seq) models (Vinyals et al., 2015) have emerged as the most prominent architecture in the NMT literature. In seq2seq models, source sentences X are *encoded* as a series of latent representations capturing words in context information. A *decoder* utilizes these hidden states, such as for initialization, to help inform the decoding process for target sentences Y. For our work, we consider both a recurrent neural network (RNN) with *attention* and *Transformer* seq2seq models for our experiments. We briefly introduce each of these next.

Recurrent Neural Network Architecture

There are a number of variations of RNN architectures previously considered for NMT. The one we chose is the default model available in the OpenNMT-py toolkit (Klein et al., 2017). It is an implementation of one of several variations studied by Luong et al. (2015) which focused on understanding attention in depth. It follows the typical seq2eq architecture but includes an attention mechanism which combines the encoder hidden states as a context vector which is added as an additional input to the decoder. We include additional details of this particular model in the supplementary material, and otherwise only mention that both the encoder and decoder are Long Short Term Memory cells (Hochreiter and Schmidhuber, 1997). For the rest of the paper we shall refer to this model as LSTM+Attn when discussing it.

Transformer

The Transformer is a model that uses intra-attention (*self-attention*) instead of sequential hidden states. For translation, it has been shown to train faster compared to RNN-based seq2seq architectures (Vaswani et al., 2017). For brevity, we exclude discussing this model in detail, and instead refer readers to the original paper Vaswani et al. (2017), or alternatively the tutorial by Rush (2018) which provides a step-by-step guide on the implementation.

2.2 Backtranslation Decisions

Applying backtranslation in practise generally requires a number of decisions such as the amount of synthetic text to add and decoding scheme choice. Both of these considerations have previously been studied by Edunov et al. (2018) which can be applied as general backtranslation guidelines. We largely based our choices off of their findings, but with one discrepancy. In their work, the emphasis was on the number of available training sentence pairs when making backtranslation choices as the key factor.

However, Edunov et al. (2018) do not discuss other aspects of bitext such as sentence length variation, number of words, or even initial bitext quality. This makes it difficult to apply their findings to other bitext corpora based solely on number of sentences. Our assumption when applying findings from Edunov et al. (2018) is that the translation system's BLEU score is more reflective of the expected synthetic sentence quality than the number of sentences used. Our final results suggest this assumption is fairly reasonable. Our Hindi – Nepali translation models, despite having the smallest bitext corpus, performed better on the test sets compared to our Polish – Czech systems following this choice.

Before backtranslating any data, we trained both the Transformer and LSTM+Attn NMT systems with only the provided bitext corpora and calculated the BLEU score on the validation set. Based on our *bitext only* model performances, we then chose the appropriate backtranslation scheme for each language pair. For the Spanish – Portuguese systems we sampled the synthetic source sentences because Edunov et al. (2018) found that for resource rich language pairs this could provide better training signal. For our work, this corresponded to randomly picking each word x_i from the probability distribution for the current position $x_i \sim p(x_i|\mathbf{y}, x_{<i})$. For both Czech – Polish and Hindi – Nepali synthetic sentences, the synthetic source sentences were deterministically produced with greedy decoding, as their validation BLEU scores were much lower. This again was in line with translation behavior of backtranslation found by Edunov et al. (2018).

We used these decoding schemes to backtrans-

late the available monolingual data with the best corresponding *bitext only* NMT system (either the Transformer or LSTM+Attn model) for each language direction. The two exceptions were Spanish and Hindi, for each of which we had significantly more monolingual data. For Spanish, we only used ∼3.3M sentences at most, and for Hindi we only used ∼2.4M sentences.

For our experiments, the best performing bitext only systems produced 2 sets of backtranslated text. The first set (which we will refer to as *Synth 1*) included only parts of all the considered monolingual data for a subset of the translation directions. The second set (henceforth referred to as *Synth 2*) consisted of backtranslating all Czech, Polish, Hindi, and Nepali monolingual data and larger portions of the Portuguese and Spanish data. As part of the *Synth 2* data set, we increased the frequency bitext was trained on compared to synthetic bitext. This meant that for every synthetic sentence our models trained on, the model was trained on several sentences of the bitext. This decision was due to the performances we found on our *Synth 1* datasets where several language pairs did not perform as well. In most cases, with the exception of few of our Spanish – Portuguese models, systems trained with these *synthetic* datasets outperformed our bitext only models.

At this point, we had produced 24 models trained on synthetic and real bitext. [1] From these 24 models, we again chose the best performing ones to perform a 3rd round of backtranslation. This 3rd set of backtranslated data (which we refer to as *Synth 3*) followed the same decoding schemes for each language pair as previously discussed. The amount of backtranslation was mostly the same except for the synthetic Portuguese to Spanish data where we backtranslated the largest amount of the available Spanish monolingual data. Exact counts are available in Tables 2,3,4. In the work we report here, we only followed this procedure once. In the future, our goal will be to follow the iterative backtranslation approach proposed by Hoang et al. (2018).

3 Dataset Analysis

In this section, we present an analysis of the shared task data. For additional information, such as our pre-processing of the data, refer to the supplementary material.

To get an understanding of the provided data, we collect statistics including the word and sentence counts, sentence length variation, and token overlap. Table 1 contains information on the approximate sentence and word counts after cleaning the data. Based on the size of the datasets, we hypothesize that our most successful NMT system would be for Spanish – Portuguese (∼3.5M sentences), followed by, Czech – Polish (∼1.7M sentences), and Hindi – Nepali being the most difficult (∼68K sentences).

In addition to this, the sentence length variations in the box-plots of Figure 1 highlight how for Spanish – Portuguese, and Czech – Polish the sentences are generally longer in the bitext compared to Hindi – Nepali. In our experimental results, we reason that part of the success for the LSTM+Attn models on Hindi – Nepali is due to the short sentence lengths. A cited advantage of the Transformer (Vaswani et al., 2017) is its ability to encode longer dependencies, but also see Tang et al. (2018), which on the Hindi – Nepali corpus would not be as much of a requirement due to the shorter bitext.

We also wanted to understand from which perspective each of the language pairs might be considered similar, so we analyzed the overlap between tokens in each language pairs bitext. We tokenized on our cleaned data with the *Tok-Tok Tokenizer* available through the NLTK toolkit.[2] We then calculated the percentage of shared tokens compared to the total tokens at increasingly higher thresholds by token frequency.

Figure 2 shows our findings for the percentage of shared tokens at different thresholds of token frequency. These plots would suggest that although Spanish – Portuguese and Czech – Polish have larger over all token overlap, the most frequent tokens are where much of the language discrepancy is. Czech and Polish in particular, seem to have significantly fewer shared tokens which could suggest a smaller lexical overlap. This could partially be because of differences in alphabets between Czech and Polish. By contrast, Hindi and Nepali seem to share much more in common as we see an increase of overlap for more frequent tokens, but we note this could be an artefact of the small size of the Hindi and Nepali data. We now present our experimental findings.

[1] 24 = 2 (Transformer vs. RNN) x 2 (*Synth 1* vs. *Synth 2*) x 6 (translation pairs).

[2] https://www.nltk.org/

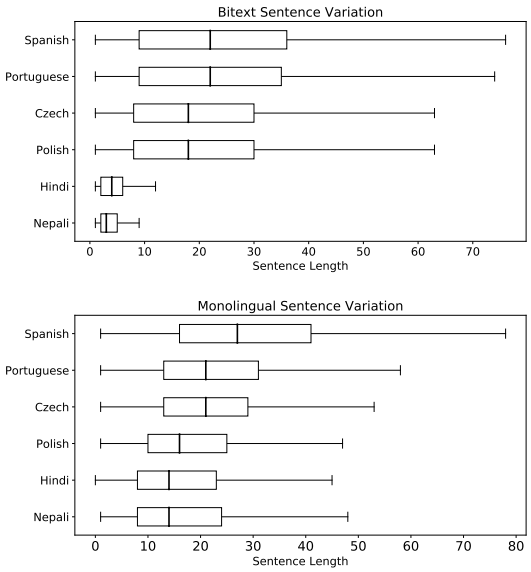

Figure 1: Boxplots showing the variation in sentence lengths between language pairs.

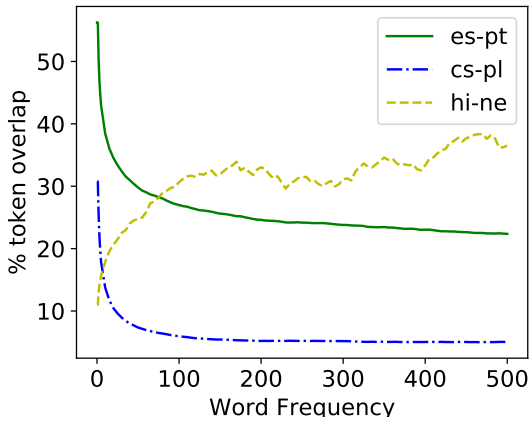

Figure 2: Lexical overlap between language pairs at different thresholds for word frequency.

	Bitext		Monolingual	
Lg.	Sentences	Words	Sentences	Words
Es	∼3.5M	∼90M	∼46M	∼1.3B
Pt		∼87M	∼10M	∼241M
Cs	∼1.7M	∼36M	∼920K	∼20.8M
Pl		∼37M	∼1.1M	∼22M
Hi	∼68K	∼360K	∼44M	∼890K
Ne		∼337K	∼551K	∼11M

Table 1: Approximate sentence and word counts for bitext and monolingual data after cleaning the data.

4 Experiments

For all of our experiments, we use OpenNMT-py (Klein et al., 2017) to handle training and build our models. For our LSTM+Attn model, we used the default parameters provided in the OpenNMT-py toolkit. For the Transformer, we used the recommended settings provided by the OpenNMT-py toolkit, with the exception of using 2 layers in the Transformer encoder and decoder instead of 6. We changed the number of Transformer layers because we found in our preliminary results on the bitext only systems that this worked well for each language direction. We did not investigate model architecture and hyperparameter tuning further, and hence we note additional work in this context could lead to better performance (Chen et al., 2018). The exact parameters are listed in the supplemental material. For our final evaluation, we also perform ensemble decoding by using different checkpoints in the optimization process and further details can be found in the supplement material.

We represented the vocabulary for each language with a joint byte-pair encoding (BPE) model (Sennrich et al., 2015b) trained on all available bitext and monolingual data shared between the languages motivated by the work of Lample et al. (2018). Our BPE models were trained with the SentencePiece API and consisted of 20,000 merge operations.[3] The reader may notice that, based on our discussion in Section3, Czech and Polish may not have necessarily benefited from a joint vocabulary. This indeed may be the case, especially as our final results for Czech – Polish translation were the lowest-performing among all our final systems.

We present our findings for each respective language pair on the validation data provided by task organizers. [4] We measure performance on the validation data with the BLEU score based on the BPE representations of sentences using the script that comes with the OpenNMT-py toolkit. Note that for our test data, BLEU score is measured on the detokenized input sequences (i.e., word tokens rather than BPE).

4.1 Spanish ↔ Portuguese Results

Table 2 shows validation results with various amounts of backtranslated text, as well as infor-

[3] https://github.com/google/sentencepiece

[4] We provide the formal task evaluation on the TEST data in Section 4.4.

Language	Model	Bitext Only	Synth 1	Synth 2	Synth 3
Es - Pt	Transformer	50.26	47.69	52.63	**52.83**
	LSTM+Attn	48.81	46.08	41.91	
Pt - Es	Transformer	51.72	54.01	53.91	**55.64**
	LSTM+Attn	49.9	50	50.5	

	Synth 1		Synth 2		Synth 3	
	Bitext	*Synthetic*	*Bitext*	*Synthetic*	*Bitext*	*Synthetic*
Es - Pt	3,517,035	2,486,960	3,517,035	3,399,936	7,034,070	3,600,928
Pt - Es		1,597,856		1,940,736		4,033,824

Table 2: Validation BLEU scores from varying quality and amount of backtranslated text for both directions for Spanish – Portuguese translation.

Language	Model	Bitext Only	Synth 1	Synth 2	Synth 3
Cs - Pl	Transformer	13.5	13.59	16.04	**16.32**
	LSTM+Attn	9.91	9.36	11.24	
Pl - Cs	Transformer	13.34	13.84	15.1	**15.57**
	LSTM+Attn	10.01	9.65	10.9	

	Synth 1		Synth 2		Synth 3	
	Bitext	*Synthetic*	*Bitext*	*Synthetic*	*Bitext*	*Synthetic*
Cs - Pl	1,713,570	874,240	3,427,140	1,194,737	3,427,140	1,194,737
Pl - Cs		921,097		921,097		921,097

Table 3: Validation BLEU scores from varying quality and amount of backtranslated text for Czech – Polish translation.

Langauge	Model	Bitext Only	Synth 1	Synth 2	Synth 3
Hi - Ne	Transformer	6.38	6.39	7.74	8.96
	LSTM+Attn	10.71	10.93	9.89	**11.72**
Ne - Hi	Transformer	5.58	13.31	12.21	13.83
	LSTM+Attn	9.48	**14.7**	11.5	14.07

	Synth 1		Synth 2		Synth 3	
	Bitext	*Synthetic*	*Bitext*	*Synthetic*	*Bitext*	*Synthetic*
Hi - Ne	304,955	278,720	304,955	452,304	487,928	452,304
Ne - Hi	609,910	647,360	609,910	2,622,219	2,439,640	2,622,219

Table 4: Validation BLEU scores from varying quality and amount of backtranslated text for both directions of Hindi – Nepali translation.

mation on the size of the training data used for each model. Note that we did not evaluate the *Synth 3* dataset on the LSTM+Attn model which was due to our previous findings and compute resource limitations.

We found that too much of the sampled backtranslated text did not necessarily improve translation quality. Between the *Synth 1* and *Synth 2* synthetic sets, we can see a small drop of performance particularly for Spanish to Portuguese translation where we had much more available monolingual data to backtranslate. In our best performing model, part of this improvement is likely due to us doubling the number of times the bitext was looked at with respect to the synthetic sentences. This is in alignment with previous research findings on the importance of bitext over synthetic sentence pairs (Sennrich et al., 2015a; Edunov et al., 2018).

4.2 Czech ↔ Polish Translation

Table 3 shows our Czech – Polish validation BLEU scores and, like our Spanish – Portuguese systems, excludes results of the LSTM+Attn model on *Synth 3* dataset. Similar to our Spanish – Portuguese models, we found that the most useful change is doubling the amount of times the bitext is trained on. One difference with our Czech – Polish data was that we had upsampled bitext sooner having tried it on the *Synth 2* dataset instead of waiting till *Synth 3*. This discrepancy allowed us to isolate improvements on the *Synth 3* dataset to the quality of synthetic sentences instead of having result confounded with upsampling like with Spanish – Portuguese. As we see in our results from *Synth 2* to *Synth 3*, where the only difference is synthetic sentence quality, we again achieve an improvement in BLEU score.

4.3 Hindi ↔ Nepali Translation

Table 4 show's our results for Hindi – Nepali translation. As our initial models on this particular pair were performing relatively poorly, we decided to train even more frequently on the bitext compared to the amounts considered on the previous language pairs. This decision was in part motivated by the results of Edunov et al. (2018) where upsampling bitext with deterministically backtranslating data in low resource language pairs seemed most effective.

Initially we believed that maintaining a close to 1-to-1 ratio of synthetic to real bitext would always be necessary to achieve better results. For the *Synth 1* dataset, we upsampled the training corpus by 5x's for Hindi to Nepali translation and 10x's for Nepali to Hindi translation. This lead to large improvements for both models when translating from Nepali to Hindi, although it did not provide quite as noticeable improvements for translating Hindi to Nepali. The most likely explanation is the noticeable difference in the amount of synthetic sentences. At least for Nepali to Hindi this choice to maintain the 1-to-1 ratio seemed to work best for Nepali to Hindi as we achieved our best performance on *Synth 1* for this translation direction.

Although generally maintaining close to a *1-to-1 ratio* seems to be important, we note one discrepancy for Hindi to Nepali results. Between the *Synth 1* to *Synth 2* Hindi to Nepali dataset we kept the upsampled bitext fixed while increasing the

	Model	Dataset	Ensemble	Val BLEU	Test BLEU
Es - Pt	Transformer	Synth 3	True	46.41	46.1
Pt - Es	Transformer	Synth 3	True	52.4	52.3
Cs - Pl	Transformer	Synth 3	False	7.88	2.3
Pl - Cs	Transformer	Synth 3	True	8.18	6.9
Hi - Ne	LSTM + Attn	Synth 3	True	10.19	8.2
Ne - Hi	LSTM + Attn	Synth 1	True	10.66	9.1

Table 5: Final BLEU scores on the detokenized translations for the best performing models across all our experiments.

amount of synthetic sentences to closer to a *2 to 3 ratio* of real to synthetic bitext. In the Transformer case, this increase in data seemed beneficial as the BLEU score for the Transformer improved, but seemed to negatively impact the LSTM+Attn model. This raises a potential question on whether considerations of backtranslation could be model dependent. We leave investigating this question as future work.

We further found that there is a limitation to the benefit of upsampling the amount of bitext despite having even more synthetic bitext. For the *Synth 3* datasets, we again returned to maintaining a 1-to-1 ratio of real to synthetic bitext. This lead to upsampling the data 10x's for translating Hindi to Nepali, and 20x's for Nepali to Hindi. This upsampling, along with higher quality synthetic data did seem to benefit both the Transformer ad LSTM+attn model for Hindi to Nepali translation which achieved our best performances. In contrast, as the amount of synthetic data increased for Nepali to Hindi translation, we observed this to negatively impact performance compared to those on the *Synth 1* datasets. Even though the synthetic sentences were produced with a better translation systems, the *Synth 3* dataset performance was still worse.

4.4 Shared Task Evaluation

Official, shared task results for our primary submissions are presented in Table 5 along with a number of important choices we made as to which models to submit. There are a number of interesting behaviors we see in terms of performance from our validation to test sets. In the Spanish – Portuguese translation systems, we can see that the relative BLEU scores between the two directions are fairly stable. This is likely in part due to the sampling process used for backtranslation we used in comparison for the other language pairs which used greedily decoded sentences. As for the other language pairs, although we originally hypothesized that Czech – Polish would produce better systems than Hindi – Nepali our results seem to suggest the opposite and that we might have overfit the Czech – Polish validation set compared to Hindi – Nepali translation.

5 Conclusion

Our findings are congruent with previous work showing the efficacy of backtranslation as a strategy for improving NMT systems. However, we couch this conclusion with caution. The reason is that tuning the correct amount of included synthetic data is still much dependent on the size of data at hand (which can be limited). Further work is needed before we can reach a more definitive recommendation as to how to perform backtranslation in different contexts, with varying degrees of resource availability.

Acknowledgments

Thank you to Pawel Przystupa and Arun Rajendran for helping evaluate sentence translation quality in Polish and Hindi respectively. Thank you to the organizers, particularly Dr. Marta Costa-Jussa, who helped us through the shared task. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and Compute Canada (www.computecanada.ca).

References

Nepali parallel corpus. https://web.archive.org/web/20160802134929/http://www.cle.org.pk/software/ling_resources/UrduNepaliEnglishParallelCorpus.htm.

News commentary parallel corpus v11 (2016). http://www.casmacat.eu/corpus/news-commentary.html.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv e-prints*, abs/1409.0473.

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*, Florence, Italy. Association for Computational Linguistics.

Sabin Bhatta. 2017. Nepali news classifier. https://github.com/sndsabin/Nepali-News-Classifier.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindMonoCorp 0.5. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. *CoRR*, abs/1804.09849.

Christos-C. 2017. Bible corpus tools. https://github.com/christos-c/bible-corpus-tools.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Vishal Goyal and Gurpreet Lehal. 2009. Hindi-punjabi machine transliteration system (for machine translation system).

Vishal Goyal and Gurpreet Singh Lehal. 2011. Hindi to punjabi machine translation system. In *Information Systems for Indian Languages*, pages 236–241, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jan Haji, Petr Homola, and Vladislav Kubo. 2003. A simple multilingual machine translation system. In *In In: Proceedings of the MT Summit IX*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Philipp Koehn. 2017. Neural machine translation. *CoRR*, abs/1709.07809.

Philipp Koehn. 2018. Global voices parallel corpus. http://casmacat.eu/corpus/global-voices.html.

Anjinkya Kulkarni. 2016. Ted parallel corpus. https://github.com/ajinkyakulkarni14/TED-Multilingual-Parallel-Corpus.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

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

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2513–2518, Copenhagen, Denmark. Association for Computational Linguistics.

Rudolf Rosa. 2018. Plaintext wikipedia dump 2018. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Alexander Rush. 2018. The annotated transformer. http://nlp.seas.harvard.edu/2018/04/03/attention.html.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. *arXiv preprint arXiv:1808.08946*.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781.

Bitext Word Counts

	Cs	PL	Es	PT	Hi	Ne
Europarl v9	1,4340,556	14,408,072	52,655,739	51,631,991		
Wiki Titles v1	552,136	554,080	1,577,829	1,546,923		
JRC-Acquis	21,465,448	22047909	34513834	32,601,655		
News Commentary v14			1406962	1358467		
Other					306,178	284,419
Dev	59,316	53,710	69,377	67,898	56,465	53,374
Total	36,417,456	37,063,771	90,223,741	87,206,934	362,643	337,793

Table 6: Sentence counts for each dataset after cleaning procedure.

Supplementary Material

A Data Sources

Submissions to the shared task were asked to only use the data provided data from the organizers. This included bitext from a number of different sources of varying utility to training translation systems. For the Spanish – Portuguese and Czech – Polish bitext corpora included the latest JRC-Acquis (Steinberger et al., 2006), Europarl (Koehn, 2005), News Commentary (new) data sets, as well as the Wiki Titles corpus (Bojar et al., 2018). The Hindi – Nepali corpus consists of the KDE, Ubuntu, and Gnome data sets available through Tiedemann (2012).[5] There was also a bilingual dictionary included for Hindi - Nepali language pair but we did not include it in our analysis because they were largely word to word translations. By the same argument, we likely should not have included the Wiki titles data set either as this corpus was also largely word to word translations. An interesting observation from our results is that our Czech – Polish systems ended up doing much worse then our Hindi – Nepali systems suggesting perhaps fewer, longer sentences are indeed more valuable then shorter, near word to word translations.

Additionally, the organizers provided monolingual datasets for Spanish, Portuguese, Czech and Polish. They all largely came from the same sources including the Europarl, JRC-Acquis, New Crawl, and News Commentary datasets. For Hindi and Nepali, we were allowed to use any monolingual data we found. For Hindi monolingual data, we only used the corpora collected by Bojar et al. (2014) which consisted of several million sentences collected from the internet. For Nepali, we largely used corpora provided in the WMT19 Parallel Corpus Filtering shared task which included a filtered Wikipedia dump of Nepali sentences, Global Voices Corpus (Koehn, 2018), the Nepali tagged corpus (nep), and a bible corpus (Christos-C, 2017). Externally, we found 3 additional Nepali corpora including one called the Nepali News corpus (Bhatta, 2017), the Ted Multilingual corpus (Kulkarni, 2016),and an additional Wikipedia dump corpus (Rosa, 2018).

A.1 Data Set Cleaning Information

To clean the datasets, we removed white spaces and re-tabulated the sentence pairs because of formatting errors. Additionally, we removed any pairs which were less than 4 characters long excluding leading and trailing white spaces. Table 6,8 contain the number of word counts per data set considered in this work. Table 7, 9 contain the sentence counts per dataset after the cleaning process.

[5]The actual Hi – Ne sources were never disclosed but were confirmed by organizers

Bitext Sentence Counts

	CS – PL	ES – PT	HI – NE
Europarl v9	615,115	1,791,082	
Wiki Titles v1	244,028	614,600	
JRC-Acquis	859,382	1,067,198	
News Commentary v14		46,850	
Other			65,506
Dev	3,051	3,001	3,001
Total	1,721,576	3,522,731	68,507

Table 7: Sentence counts for each dataset after cleaning procedure.

Monolingual Datasets Word Counts

	Cs	PL	Es	PT	Hi	Ne
Europarl v9	15,129,685	8,117,153	57,499,268	56,486,759		
New commentary v14	5,699,897		11,879,901	1,611,655		
News Crawl 2007 - 2018		14,348,031	1,311,839,007	183,746,078		
Hindi Monolingual					890,209,442	
Ted Multilingual						32,078
Filtered Wikipedia Dump						2,939,682
Wikipedia Dump						3,477,956
Global Voice						86,703
Nepali Tagged Corpus						51,276
Nepali NewsCorpus						4,616,548
Bible Corpus						769,344
Total	20,829,582	22,465,184	1,381,218,176	241,844,492	890,209,442	11,973,587

Table 8: Sentence counts for each dataset after cleaning procedure.

Monolingual Datasets Sentence Counts

	Cs	Pl	Es	Pt	Hi	Ne
Europarl v9	661,426	380,336	2,004,495	2,004,629		
New commentary v14	259,666		412,791	58,002		
News Crawl 2007 - 2018		814,397	43,807,883	8,299,115		
Hindi Monolingual					44,486,496	
Ted Multilingual Corpora						4,345
Filtered Wikipedia Dump						92,296
Wikipedia Dump						118,519
Global Voice						2,892
Nepali Tagged Corpus						4,287
Nepali NewsCorpus						298,151
Bible Corpus						30,547
Total	921,092	1,194,733	44,486,496	10,361,746	47,108,715	551,037

Table 9: Sentence counts for each dataset after cleaning procedure.

B Model Information

B.1 Details on RNN with Attention Model

As mentioned in the paper, our RNN architecture is a one of several studied in the work of Luong et al. (2015). The particular model we use can be described with the following equations.

$$z_i = \text{Encoder}(x_i, z_{i-1}) \ , \ \forall i \ \epsilon \ T \tag{1}$$

$$\text{score}(z_i, s_j) = z_i W^g s_j \ , \ \forall i \ \epsilon \ T \tag{2}$$

$$\alpha_i = \text{softmax}(\text{score}(z_i, s_j)) \ , \ \forall i \ \epsilon \ T \tag{3}$$

$$c = \sum_{i=1}^{T} \alpha_i * z_i \tag{4}$$

$$\tilde{s}_{j-1} = W^s[c; s_{j-1}] \tag{5}$$

$$s_j = \text{Decoder}(\tilde{s}_{j-1}, y_j, s_{j-1}) \tag{6}$$

$$p(y_j | y_{<j}, \mathbf{x}) = \text{Generator}(\tilde{s}_j) \tag{7}$$

The *encoder* and *decoder* are Long Short Term Memory (LSTM) RNNs (Hochreiter and Schmidhuber, 1997), where the encoder produces latent representations z_i for each word embedding x_i in the source sentence of length T. Equation 2 refers to *general* attention proposed by Luong et al. (2015), where W^g is learned and Equations 3 and 4 show the application of this global attention mechanism. The *decoder* LSTM then produces hidden states s_j using as input the word embedding y_j, context vector $\tilde{s}_{j-1}$, and previous hidden state s_{j-1}. The context hidden states $\tilde{s}_j$ are how the log-probability of target words are determined and are calculated on the concatenation of context c and previous hidden state s_{j-1} with learned parameters W^s.

B.2 Ensemble Decoding

As a way to further improve translation system quality, previous research has shown that an ensemble of models can improve translation performance (Koehn, 2017). For our work this meant using a window around the best performing single models that we found on the evaluation set. By window we mean we translated the test and evaluation sets with the single best model along with the n checkpoint models before, and n checkpoint models after the single best model.

For our final evaluations this involved either $n = 1$ or $n = 2$ windows around the best performing models. We did not find much difference between the two choices of n as both generally gave only minute improvements to performance. Our checkpoints were saved after every 10,000 mini-batch updates. As an example, generally we found the Transformer worked well with around 50,000 or 60,000 updates. Supposing we found 50,000 steps the best along with picking $n = 1$, we then included the checkpoint at 40,000 updates and 60,000 updates to translate the final model.

B.3 Hyperparameter Information

Table 10 contains the specific parameters for the models used in our analysis. One parameter left out of the tables was the number of updates which in OpenNMT-py is counted per batch update. For the RNN model we found 150,000 steps generally sufficient for our best performances on the Hindi – Nepali data, and at most 60,000 or 50,000 steps with the Transformer sufficient for Spanish – Portuguese and Czech – Polish even with the backtranslated data.

B.4 Tuning results

In Table 11 shows the full results of tuning our models. As a reminder, the BLEU scores were calculated on the byte-pair encoding representations of the sentences instead of the detokenized translations. This is in part why the scores, particularly in some cases, are much higher than the final validation scores reported in the paper.

LSTM Model	
Embed Dim	*500*
RNN Type	LSTM
Num Layers	2
Hidden Dim	500
Input Feeding	True
Attention	Global
Attenion Type	General
Dropout	0.3
Optimization	
Batch Size	32
Batch type	Sentences
Optimizer	SGD
Init Learning Rate	1.0
Learning Rate Schedule	
# Steps before Decay	50,000
Decay Frequency	10,000 steps
Decay Schedule	lr_{curr} * 0.5

Transformer	
Embed Dim	*512*
RNN Type	Transformer
Num Layers	2
Hidden Dim	512
Num Heads	8
Attenion Type	Multi-Head
Fully Connected Hidden Size	2048
Dropout	0.1
Position Encoding	Sinusoidal
Optimization	
Batch Size	4096
Batch type	Tokens
Optimizer	Adam
β_2	0.998
Init Learning Rate	2.0
Label Smoothing	0.1
Gradient Accum. Count	2
Learning Rate Schedule	
# Steps before Decay	8000
Decay Schedule	Noam

Table 10: The parameters used for the RNN Model and the Transformer model. Parameters are largely from the OpenNMT-py toolkit suggested parameters.

	Model	Decoding Type	Bitext Only	Bitext + Synth 1	Bitext + Synth 2	Bitext + Synth 3
Es - Pt	Transformer	Sampling	50.26	47.69	52.63	**52.83**
Pt - Es			51.72	54.01	53.91	**55.64**
Cs - Pl		Greedy	13.5	13.59	16.04	**16.32**
Pl - Cs			13.34	13.84	15.1	**15.57**
Hi - Ne			6.38	6.39	7.74	**8.96**
Ne - Hi			5.58	13.31	12.21	**13.83**
Es - PT	LSTM+Attn	Sampling	**48.81**	46.08	41.91	
Pt - Es			49.9	50	**50.5**	
Cs - Pl		Greedy	9.91	9.36	**11.24**	
Pl - Cs			10.01	9.65	**10.9**	
Hi - Ne			10.71	10.93	9.89	**11.72**
Ne - Hi			9.48	**14.7**	11.5	14.07

BLEU Score

Table 11: BLEU scores on the validation set. These scores were calculated on the BPE tokens.