Iterative Data Augmentation for Neural Machine Translation: a Low Resource Case Study for English–Telugu

Sandipan Dandapat and Christian Federmann
Microsoft AI & Research
{sadandap,chrife}@microsoft.com

Abstract

Telugu is the fifteenth most commonly spoken language in the world with an estimated reach of 75 million people in the Indian subcontinent. At the same time, it is a severely low resourced language. In this paper, we present work on English–Telugu general domain machine translation (MT) systems using small amounts of parallel data. The baseline statistical (SMT) and neural MT (NMT) systems do not yield acceptable translation quality, mostly due to limited resources. However, the use of synthetic parallel data (generated using back translation, based on an NMT engine) significantly improves translation quality and allows NMT to outperform SMT. We extend back translation and propose a new, iterative data augmentation (IDA) method. Filtering of synthetic data and IDA both further boost translation quality of our final NMT systems, as measured by BLEU scores on all test sets and based on state-of-the-art human evaluation.

1 Introduction

In the past two decades, machine translation (MT) has shown very promising results, most of which have been achieved using data-driven techniques. In recent years, the data-driven paradigm of MT is largely dominated by neural machine translation (NMT) and showing significant success over its predecessor statistical machine translation (SMT) (Bahdanau et al., 2014; Bojar et al., 2017).

The performance of any data-driven approach to MT mostly depends on the amount of parallel corpora available to train them. This problem is exacerbated by NMT, which generally needs larger quantities of parallel data and is less robust to noisy data. Unfortunately, large amounts of readily available parallel resources exist only for a small number of languages, e.g., OPUS (Tiedemann and Nygaard, 2004) and Europarl (Koehn, 2005), with only very few sources of Indic language data.

Indic language MT is difficult due to complex linguistic structure and lack of good quality data. Most of the Indic languages are leading languages of the world in terms of number of speakers but are very poorly resourced (i.e., only very little machine-readable parallel text exists) so building a general domain data-driven MT system is a challenging problem. Also, Indic languages do not have enough comparable resources to explore extraction of useful parallel content from the same (Irvine and Callison-Burch, 2013). Lastly, due to the usage of multiple fonts and encodings, a significant portion of the web data cannot be used to extract parallel data for training. Telugu is no exception to this. Lack of large, high quality parallel resources makes the development of general purpose MT systems much harder for Telugu compared to other, resource rich languages, more specifically when building NMT-based models.

One of the major problems with training an NMT system on little data, especially when training an engine for general usage (i.e., not domain specific), is overfitting. Deep neural networks have large parameter spaces and need ample amounts of data in order to generalize adequately; with small amounts of data they tend not to generalize well. We address this issue by learning the optimizer over a smaller number of training steps.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 287–292
Alacant, Spain, May 2018.

In this paper, we describe our English–Telugu (En–Te) general purpose MT system. First, we describe the baseline SMT-based and NMT-based systems trained on 750k parallel sentences. Telugu is a morphologically rich language and, as such, suffers from a high out-of-vocabulary (OOV) rate in a low data scenario. We address data sparsity by augmenting a large amount of synthetic training data (Sennrich et al., 2015), generated using back translation, to iteratively improve the NMT systems. The iterative process uses synthetic data to improve the MT engine and (implicitly) the quality of the synthetic data using the improved MT engine in the reverse direction.

Secondly, we use sub-word representations to reduce the data sparsity problem. This essentially handles Telugu's rich morphology. Furthermore, as translation quality varies across sentences while generating synthetic data, we filter poor quality translation pairs to augment the system only with high quality synthetic parallel data. We observe improved translation quality as a result. The main finding of this work is that the use of iterative data augmentation and filtering of the synthetic data help to improve the translation quality.

The rest of the paper is organized as follows. Section 2 describes the data sets used to build the systems. In Section 3, we describe the baseline SMT and NMT models and their quality. Section 4 provides details on improving NMT models using synthetic data. Section 5 reports the experimental setup and results. We conclude in Section 6.

2 Data Sets

In this work, we use two types of training data: true parallel data, and synthetically generated parallel data using back-translation (Sennrich et al., 2015). In this section, we describe the true English–Telugu parallel data used for system training. The generation of synthetic data is explained in Section 4.1. The full training data contains 750k true parallel sentences along with a larger set of synthetic data (15.4M and 8.2M for En→Te and Te→En, respectively).

The true parallel data includes automatically extracted parallel sentences from the web and from OPUS (Tiedemann and Nygaard, 2004). Many web pages feature content available in multiple languages. Such content includes both sentence or paragraph aligned parallel data (e.g., TED talks' transcriptions) and comparable or noisy-parallel corpora (e.g., cross-lingually linked Wikipedia documents). Once such potential parallel pages between Telugu and English are extracted from the web, a sentence aligner is used to extract sentence aligned parallel text, based on a modified Moore Sentence Aligner (Moore, 2002).

Test Data To the best of our knowledge, there are no publicly available test sets for evaluating Te–En MT systems. Thus, we have created two different test sets to evaluate our systems. Our first test set was created by selecting sentences from news articles. The English source sentences were manually translated into Telugu and validated by human experts. We shall refer this test set as **News**.

In order to understand the performance of our systems w.r.t. state-of-the-art test sets, we have created our second test set using a subset of the WMT 2009 (Callison-Burch et al., 2009) test set for English–French. 1,000 English sentences were randomly selected and manually translated into Telugu by human experts. We call this test set **WMT**. Table 1 summarizes the different data used for training and testing.

Parallel Data	#sentences	#En	#Te
Train	751,609	13.6	10.4
News (test set)	5,000	14.4	10.9
WMT (test set)	1,000	22.8	16.4
Dev	2,500	20.4	14.3
Monoligual Data			
English	8.2m	15.7	–
Telugu	15.4m	–	8.6

Table 1: Number of sentences (#sentences) and average sentence lengths (#En, #Te) for data sets used in this work.

Note that we have created our test sets with a single reference translation. We intend to publicly release the test sets. Monolingual data mentioned in Table 1 is used to build the language models for SMT systems and to generate synthetic parallel data used to train the NMT systems.

3 Baseline Models

The baseline SMT models use a vanilla **phrasal** (Koehn et al., 2003) and a **treelet**[1] (Quirk et al., 2005; Bach et al., 2009) translation model for Te→En and En→Te systems, respectively. We do not use treelet translation system in the Te→En

[1]Extracts treelet translation pairs using source language dependency parse tree and an unsupervised alignment algorithm. This is used for tree-based reordering.

direction due to lack of a Telugu parser. For both phrasal and treelet systems, word alignment is done using GIZA++ (Och and Ney, 2003). We use the target side of the parallel corpus along with additional monolingual target language data to train a 5-gram language model using modified Kneser–Ney smoothing (Kneser and Ney, 1995). Finally, we use MERT (Och, 2003) to estimate the lambda parameters using the held out *Dev* data with a single reference translation.

The baseline NMT model is developed based on the architecture described in (Devlin, 2017). The encoder uses a 3-layer bi-directional RNN (consists of 512 LSTM units). The decoder uses an LSTM layer in the bottom to capture the context and the attention. The LSTM layer is then followed by 5 fully-connected layers applied in each timestep using a ResNet-style skip connection (He et al., 2016). The details of the model and equations are described in (Devlin, 2017). All the models are trained using ADAM optimizer (Kinga and Adam, 2015) with a dropout rate of 0.25. The optimizer uses 100k and 500k steps with a batch size of 1024 for En→Te and Te→En baseline NMT systems, respectively. In the case of Te→En NMT system, source-side Telugu sentences are represented using byte-pair encoding (BPE) (Sennrich et al., 2015) to reduce the data sparsity problem, which uses 50,000 merging operations.

Table 2 summarizes the baseline accuracy of the MT systems on different test sets. We use BLEU (Papineni et al., 2002) score for automatic evaluation of all the systems. It is interesting to note that the baseline SMT systems in general have higher scores for most of the test scenarios compared to the NMT baselines (except for the News test set in the Te→En direction). This essentially indicates that 750k parallel data is not enough to build NMT-based systems with better quality translation compared to corresponding SMT-based systems due to large parameter space of the NMT-based systems. In addition, the absolute BLEU scores achieved by the baseline systems (either NMT or SMT) are quite low, especially in the En→Te direction. We observe that En→Te has much lower BLEU scores compared to Te→En, irrespective of the MT techniques used. This is often the case for morphologically rich, free word order target languages when using automated metrics based on single references.

System	Te→En		En→Te	
	News	WMT	News	WMT
SMT	9.12	8.76	4.99	3.98
NMT	9.13	7.59	4.04	3.26

Table 2: BLEU scores of the baseline systems

4 Improved NMT Models

The baseline experiments in the previous section clearly associate with the fact that NMT models require massive amount of parallel data in order to generalize over the large parameter space of the model (Gu et al., 2018). Researchers have tried different data augmentation techniques (Gulcehre et al., 2015; Cheng et al., 2016) to improve NMT models. Most of the data augmentation techniques try to leverage the use of monolingual data. We adopt the *back-translation* technique proposed by (Sennrich et al., 2015) to improve the quality of the MT system, which has shown notable success in the past. In this direction, we use an iterative data augmentation and filtering strategy to improve translation quality.

4.1 Back-Translation

To improve our models, first, we use back-translation (Sennrich et al., 2015) to increase the use on parallel data. Back-translation uses a reverse translation engine to translate target-side monolingual data and essentially produced the synthetic data to train the system in forward direction. For example, let e_i be an English sentence, and $t'_i = MT_{En \rightarrow Te}(e_i)$ is the translation produced by the $En \rightarrow Te$ MT system. Then the $Te \rightarrow En$ system is trained on $\{t'_i, e_i\}$ data.

We use the monolingual data mentioned in Table 1 to generate the back-translated data. Table 3 summarizes the detail of the synthetic data used to train the NMT systems. Note, after adding synthetic data, we train the ADAM optimizer with 200k steps with a batch size of 4,096.

Corpus	#sentences	#En	#Te
En_{synth}, Te_{mono}	15.4m	11.4	8.6
Te_{synth}, En_{mono}	8.2m	15.7	12.6

Table 3: Synthetic data

4.2 Iterative Data Augmentation

A good quality baseline system (i.e., reverse translation engine) is required to produce good quality

synthetic data. The quality of the synthetic data affects the quality of the MT system. Due to the low quality of the baseline systems (cf. Table 2), we plan to improve the quality of the synthetic data iteratively through iterative data augmentation. The detail of our algorithm is given in Algorithm 1. In line 1 and 2 of the algorithm, we build the baseline reverse translation engines ($M^{(0)}$) using only true parallel data (D_{bi}). Line 4 of the algorithm uses the baseline $M^{(0)}_{En\rightarrow Te}$ to produce synthetic parallel data $\langle D'_{Te}, D_{En}\rangle$ which is further used to improve the MT quality in the other direction ($M^{(t)}_{Te\rightarrow En}$) in line 5. Instead of using the baseline engines ($M^{(0)}_{Te\rightarrow En}$), we use the modified $M^{(t-1)}_{Te\rightarrow en}$ engine in line 6 to produce synthetic data $\langle D'_{en}, D_{Te}\rangle$. Finally, in line 7, we improve the $M^{(t)}_{En\rightarrow Te}$ system using the synthetic data produced in line 6. We continue the process until there is no overall gain (average over Δ_{BLEU} in $Te \rightarrow En$ and $Te \rightarrow En$ directions) in BLEU score. This is ensured in line 8 by measuring the change in BLEU score in the dev set between two successive iterations.

Algorithm 1 iterativeAugment(D_{En}, D_{Te}, D_{bi})

In: Monolingual English corpus D_{En},
Monolingual Telugu corpus D_{Te},
English–Telugu parallel corpus D_{bi}
Out: Translation models $M^{(t)}_{Te\rightarrow En}$ and $M^{(t)}_{En\rightarrow Te}$

1: $M^{(0)}_{En\rightarrow Te} \leftarrow$ baseline En-to-Te NMT system using D_{bi}
2: $M^{(0)}_{Te\rightarrow En} \leftarrow$ baseline Te-to-En NMT system using D_{bi}
3: **for** $t := 1$ **to** T **do**
4: $\quad D'_{Te} \leftarrow$ Translate D_{En} to Telugu using $M^{(t-1)}_{En\rightarrow Te}$
5: $\quad M^{(t)}_{Te\rightarrow En} \leftarrow D_{bi}$ + $\{D'_{Te}$,$D_{En}\}$
6: $\quad D'_{En} \leftarrow$ Translate D_{Te} to English using $M^{(t-1)}_{Te\rightarrow En}$
7: $\quad M^{(t)}_{En\rightarrow Te} \leftarrow D_{bi}$ + $\{D'_{En}$,$D_{Te}\}$
8: $\quad$ **if** $\frac{1}{2}(\Delta_{BLEU}(\text{dev},M^{(t)}_{Te\rightarrow En})+\Delta_{BLEU}(\text{dev},M^{(t)}_{En\rightarrow Te}))$ ≤ 0 **then**
9: $\quad\quad$ return $M^{(t-1)}_{Te\rightarrow En}$, $M^{(t-1)}_{En\rightarrow Te}$
10: $\quad$ **end if**
11: **end for**

4.3 Data Filtering

Although the quality of the synthetic data improves through the iterative process in the Algorithm 1, we found that the back-translation quality varies widely across sentences. Thus, we filter poor quality back-translated sentences using a pseudo fuzzy match (PFS) score (He et al., 2010) to rank all the back-translated output. For example, in line 6, once the synthetic parallel data (e.g., $\langle D'_{en}, D_{te}\rangle$) is produced using reverse translation engine (e.g., $M^{(t)}_{Te\rightarrow En}$), we further translate the back-translated D'_{en} into Telugu (D''_{te}) using forward translation engine $M^{(t)}_{En\rightarrow Te}$. We measure the PFS between t ($\in D_{te}$) and t'' ($\in D''_{te}$) as shown in Equation 1.

$$PFS = 1 - \frac{EditDistance(t, t'')}{max(|t|, t''|)} \quad (1)$$

This essentially helps ranking each pair in the synthetic parallel data with higher scores corresponding to better translation quality.

5 Experiments and Results

First, we conducted one experiment to see the effect of choosing SMT and NMT system as the reverse translation engine to produce back-translated data (line 1 and 2 in Algorithm 1). Note that our baseline SMT system has better quality compared to the baseline NMT system (cf. Table 2). However, we found that the use of NMT as the reverse translation engine has better improvement in translation quality compared to using SMT system for back-translation. Table 4 shows the effect of SMT and NMT system as reverse translation engine. In this process we rely on the baseline $M^{(0)}$ (as shown in line 1 and 2 of the Algorithm 1) and do not use any iterative augmentation of data.

System	Te→En		En→Te	
	News	WMT	News	WMT
SMT	12.78	12.26	5.29	4.14
NMT	14.21	13.26	5.71	4.55

Table 4: Effect of MT system type on back translation. NMT achieves higher quality gains compared to SMT.

The accuracies in Table 4 show that the use of synthetic parallel data significantly improves the baseline translation quality (cf. Table 2). The use on SMT as back-translation system gives an average improvement of 3.58 and 4.16 absolute BLEU points for Te→En system over the baseline SMT and NMT system, respectively. Similar observations are found in En→Te directions with 0.23 and 1.07 absolute BLEU point improvement over the baseline SMT and NMT system, respectively.

Furthermore, we found an absolute average BLEU score improvement of 1.22 and 0.41 using NMT for generating back-translated data compared to the SMT reverse translation system, respectively for Te→En and En→Te systems.

We conduct a second experiment based on the iterative data augmentation technique described in Algorithm 1. We shall refer this as **IDA**. Here we do not filter any data based on PFS value (i.e $PFS \geq 0$). Figures 1 and 2 shows the effect

PFS	#data	Te → En	#data	En → Te
≥0	8.2m	15.05	15.4m	6.49
≥0.3	7.3m	15.14	9.8m	6.66
≥0.5	6.3m	**15.22**	6.7m	**6.77**
≥0.7	4.1m	15.20	3.1m	6.57

Table 5: The effect of PFS on News test set

of IDA over the baselines and non-iterative data augmentation (NMTBT) on different test sets for En→Te and Te→En. We found that the algorithm has no improvement after 2nd iteration in both the directions.

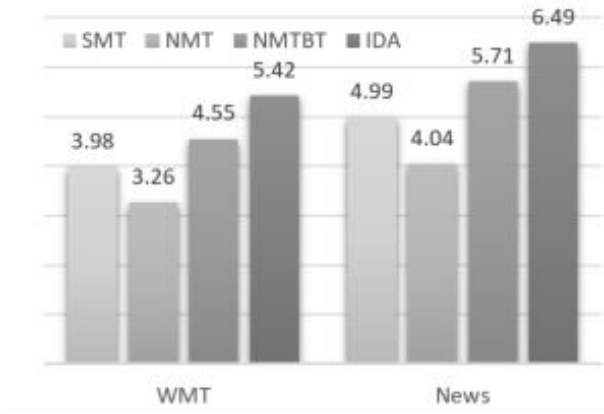

Figure 1: Comparison of BLEU scores for $En \rightarrow Te$. SMT and NMT are baseline systems, NMTBT refers to NMT system with baseline synthetic data.

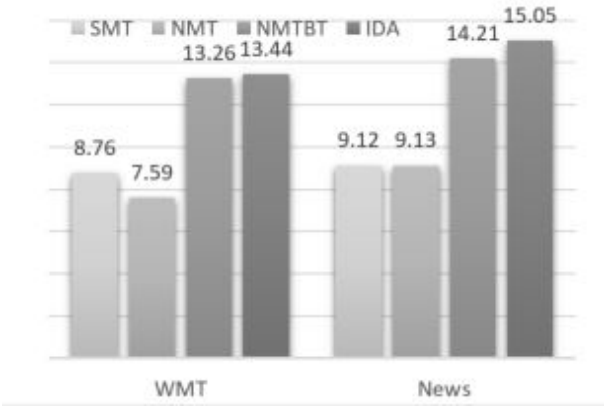

Figure 2: Comparison of BLEU scores for $Te \rightarrow En$

Finally, in our last experiment we show the effect of different PFS threshold for data filtering and their effective impact on BLEU score. Table 5 shows the effect of data filtering using PFS on the News test set. We found that the filtering of data generally improves the translation quality. The best accuracy is achieved when the synthetic data is selected with $PFS \geq 0.5$.

System	Te → En	En → Te
SMT	27.9	35.3
$\text{NMT}_{IDA,PFS \geq 0.5}$	56.7	49.6

Table 6: Human evaluation scores on News test set. Based on source-based Direct Assessment. Differences are statistically significant according to Wilcoxon rank sum test with p-level $p \leq 0.05$. Human perceived quality indicates that the NMT system may be good enough for actual general domain use.

5.1 Human Evaluation

In addition to the above automatic evaluations, we performed a manual evaluation of the MT output for both language directions to understand the translation quality from a human perspective. Human evaluation for this research is based on direct assessment. We follow WMT17 (Bojar et al., 2017) and use Appraise (Federmann, 2012), modified to show source sentences instead of reference translations. This adopts the evaluation strategy implemented for IWSLT17 (Cettolo et al., 2017).

For each language direction, five independent annotators evaluated 350 candidate translations on the News test set, randomly drawn from both the baseline SMT (cf. Table 4) and the final NMT system (using IDA and PFS ≥ 0.5). Following direct assessment as implemented at IWSLT17, annotators see the source text and a corresponding candidate translation and are asked to assign a quality score $x \in \{0, 100\}$.

After filtering out annotations used for quality control, we collected an average number of 402 segment scores for SMT, and 399 for NMT. Table 6 shows the average absolute translation quality of the two approaches in both directions. The human evaluation shows statistically significant improvement of 103% and 41% in the absolute scale for Te→En and En→Te NMT systems, respectively, compared to the SMT baseline. We use Wilcoxon rank sum test (Wilcoxon, 1945) with p-level $p \leq 0.05$ to determine statistical significance. All collected data points will be released publicly.

6 Conclusion

We have demonstrated that we can build good quality NMT models with limited resources for a morphologically rich language pair. Contributions of this paper are the definition of iterative data augmentation (IDA) and empirical results showing the effectiveness of back translation and PFS-based data filtering for English–Telugu NMT. The proposed IDA method is much more effective than using baseline back translation by itself.

References

Bach, N., Gao, Q., and Vogel, S. (2009). Source-side dependency tree reordering models with subtree movements and constraints. *Proceedings of the MTSummit-XII, Ottawa, Canada, August. International Association for Machine Translation*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proc. of the 2nd Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J. (2009). Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. (2017). Overview of the iwslt 2017 evaluation campaign. In *Proc. of IWSLT*, Tokyo, Japan.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596*.

Devlin, J. (2017). Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu. *arXiv preprint arXiv:1705.01991*.

Federmann, C. (2012). Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Gu, J., Hassan, H., Devlin, J., and Li, V. O. (2018). Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, Y., Ma, Y., Way, A., and Van Genabith, J. (2010). Integrating n-best smt outputs into a tm system. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 374–382. Association for Computational Linguistics.

Irvine, A. and Callison-Burch, C. (2013). Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Kinga, D. and Adam, J. B. (2015). A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54. Association for Computational Linguistics.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting of the ACL*, pages 160–167. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal smt. In *Proc. of the 43rd Annual Meeting of the ACL*, pages 271–279. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Tiedemann, J. and Nygaard, L. (2004). The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC*.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83.