Transformers for Low-resource Neural Machine Translation

Andargachew Mekonnen Gezmu[a] and Andreas Nürnberger[b]
Faculty of Computer Science, Otto von Guericke Universität Magdeburg, Universitätsplatz 2, Magdeburg, Germany

Keywords: Neural Machine Translation, Transformer, Less-resourced Language, Polysynthetic Language.

Abstract: The recent advances in neural machine translation enable it to be state-of-the-art. However, although there are significant improvements in neural machine translation for a few high-resource languages, its performance is still low for less-resourced languages as the amount of training data significantly affects the quality of the machine translation models. Therefore, identifying a neural machine translation architecture that can train the best models in low-data conditions is essential for less-resourced languages. This research modified the Transformer-based neural machine translation architectures for low-resource polysynthetic languages. Our proposed system outperformed the strong baseline in the automatic evaluation of the experiments on the public benchmark datasets.

1 INTRODUCTION

Machine translation is a helpful mechanism to tackle language barriers that potentially lead to frustration, isolation, and racism. The state-of-the-art machine translation, Neural Machine Translation (NMT), requires extensive training data to build competitive models despite its usefulness. There are significant improvements in NMT for a few high-resource languages. However, since the amount of training data significantly affects the quality of NMT models (Koehn and Knowles, 2017; Lample et al., 2018), performance levels are still low for less-resourced languages. Of the approximately 7000 languages spoken today, very few have adequate resources for NMT. If we neglect less-resourced languages, there will be grave consequences in integrating societies in today's globalized world. Even worse, in the long run, this leads to the danger of digital language death, a massive die-off caused by the digital divide (Kornai, 2013).

For less-resourced languages, tuning hyperparameters for low-data conditions, including modifying the system architecture in low-resource settings, is indispensable. Hyperparameters incorporate basic settings such as the learning rate, mini-batch size, system architecture, and regularization. Specifically, the system architecture provides the number of layers, the number of hidden nodes per layer, the choice of activation functions, and others.

Prior work has proposed different architectures for low-resource NMT (Östling and Tiedemann, 2017; Nguyen and Chiang, 2018; Sennrich and Zhang, 2019). Each architecture exhibits an extra level of performance based on training data size. The previous work primarily modified the Recurrent Neural Network (RNN) for NMT. While RNN-based architectures (Sutskever et al., 2014; Bahdanau et al., 2015) have been used for NMT to obtain good results, the Transformer-based ones are even attaining better successes in high-resource data conditions (Vaswani et al., 2017).

In this research, we adapted the Transformer-based NMT system for low-resource polysynthetic languages. We also developed a baseline phrase-based Statistical Machine Translation (SMT) system. We evaluated our proposed and baseline systems with public benchmark datasets of Amharic-English and Turkish-English. We chose these language pairs since they have different morphology and orthography features. Turkish is primarily an agglutinative language, in which a space-delimited word is a concatenation of several morphemes. Amharic is chiefly a fusion language, in which an orthographic word is a blending of several morphemes without clear boundaries. English has a relatively simple morphology while sharing both features. Besides, Amharic uses the Ethiopic script, whereas the other languages use Latin-based scripts.

[a] https://orcid.org/0000-0002-3424-755X
[b] https://orcid.org/0000-0003-4311-0624

Gezmu, A. and Nürnberger, A.
Transformers for Low-resource Neural Machine Translation.
DOI: 10.5220/0010971500003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 1*, pages 459-466
ISBN: 978-989-758-547-0; ISSN: 2184-433X

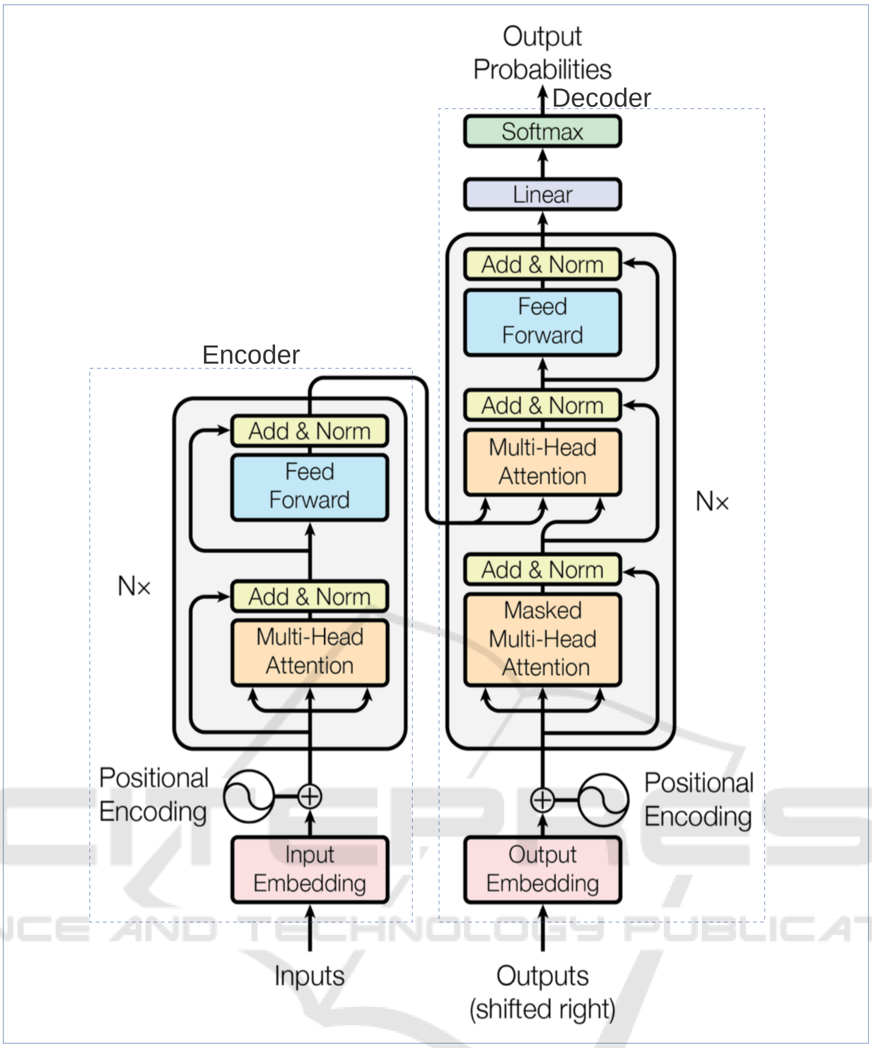

Figure 1: The Transformer architecture. Adapted from Vaswani et al. (2017).

2 SYSTEM ARCHITECTURE

To train NMT models, we used the encoder-decoder architecture implemented with Transformers. The encoder, shown in the left half of Figure 1, comprises a stack of N identical Transformer blocks. Each Transformer block contains a multi-head self-attention layer followed by a fully-connected feed-forward layer with residual connections and layer normalizations. The decoder, shown in the right half of Figure 1, is similar to the encoder, except it includes a masked multi-head self-attention layer, which is a modification of multi-head self-attention to prevent positions from interfering in subsequent computations.

It would be difficult for a single Transformer block to learn to capture all of the different kinds of complex relations among its inputs. For example, words in a sentence can simultaneously relate to each other in many different ways. Likewise, distinct syntactic, semantic, and discourse relationships can hold between verbs and their arguments in a sentence. Transformers model these complex relations with multihead self-attention layers. These are sets of self-attention layers, called heads, that reside in parallel layers at the same depth, each with its own set of parameters. Given these distinct sets of parameters, each head can learn different aspects of the relationships among inputs at the same level of abstraction.

With Transformers, information about the order of the inputs is not an integral part of the network. Therefore, nothing would allow them to use information about the relative or absolute positions of the elements of an input sequence. Thus, positional en-

Table 1: Differences between TransformerShallow1, TransformerShallow2, and TransformerDeep.

Hyperparameter	TransformerShallow1	TransformerShallow2	TransformerDeep
Batch size	1024	4096	1024
Filter size	512	512	2048
Hidden size	128	128	512
Number of heads	4	4	8
Transformer blocks	2	2	6

coding combines Transformer inputs to each specific position in an input sequence. Vaswani et al. (2017) discuss further details about the overall architecture.

Because of the long training times of NMT models, we followed the best practices of previous research in low-resource settings instead of working with all possible hyperparameters, including various architectures. In RNN-based NMT systems, there are mixed findings on the size of training batch sizes in low-data conditions. While Morishita et al. (2017) and Neishi et al. (2017) are using large batch sizes, Sennrich and Zhang (2019) recommend small batch sizes. There is also a trend to use smaller and fewer layers (Nguyen and Chiang, 2018).

Therefore, we proposed three different architectures: TransformerShallow1, TransformerShallow2, and TransformerDeep. All systems use Adam optimizer (Kingma and Ba, 2015) with varied learning rate over the course of training, dropout (Srivastava et al., 2014) rate of 0.1, and label smoothing (Szegedy et al., 2016) of value 0.1. Table 1 details the differences between the three architectures. TransformerShallow1 and TransformerShallow2 differ only in training batch sizes. Here, training batch sizes are the source and target language tokens. We give all the common hyperparameters shared among the three systems in the appendix.

We used Google's tensor2tensor[1] (Vaswani et al., 2018) library to implement our system. The preconfigured hyperparameters in tensor2tensor are the basis for the aforementioned three architectures.

3 BASELINE SYSTEM

Our phrase-based SMT baseline system had settings that were typically used by Ding et al. (2016), Williams et al. (2016), and Sennrich and Zhang (2019). We used the Moses (Koehn et al., 2007) toolkit to train phrase-based SMT models. First, we used GIZA++ (Och, 2003) and the grow-diag-final-and heuristic for symmetrization for word alignment. Then, we used the phrase-based reordering model (Koehn et al., 2003) with three different orientations: monotone, swap, and discontinuous in backward and forward directions conditioned on the source and target languages.

We used five-gram language models smoothed with the modified Kneser-Ney (Kneser and Ney, 1995). The system applied KenLM (Heafield, 2011) language modeling toolkit for this purpose. Initially, we have not used big monolingual corpora for language models. This is because they are no longer the exclusive advantages of phrase-based SMT, as NMT can also benefit from them (Sennrich and Zhang, 2019). Afterward, to prove this claim, we used the Contemporary Amharic Corpus[2] (CACO) (Gezmu et al., 2018) for English-to-Amharic translation.

The feature weights were tuned using Minimum Error Rate Training (MERT) (Och, 2003). We also used the k-best batch Margin Infused Relaxed Algorithm (MIRA) for tuning (Cherry and Foster, 2012) by selecting the highest-scoring development run with a return-best-dev setting.

In decoding, we applied cube pruning (Huang and Chiang, 2007), a distortion limit of six, and the monotone-at-punctuation (do not reorder over punctuation) heuristic (Koehn and Haddow, 2009).

4 EXPERIMENTS AND EVALUATION

We evaluated the performance of our baseline and proposed systems. The experiments used the same datasets for each system; preprocessing, training, and evaluation steps were similar.

Table 2: The number of sentence pairs in each dataset.

Language pair	Dataset	Sentence pairs
Amharic-English	Test	2500
	Development	2864
	Training	140000
Turkish-English	Test	3010
	Development	3007
	Training	207373

[1] Available at: https://github.com/tensorflow/tensor2tensor

[2] Available at: http://dx.doi.org/10.24352/ub.ovgu-2018-144

Table 3: Performance results of TransformerShallow1, TransformerShallow2, and TransformerDeep.

Translation Direction	NMT System	BLEU	BEER	CharacTER
English-to-Amharic	TransformerShallow1	17.8	0.485	0.639
	TransformerShallow2	18.9	0.498	0.614
	TransformerDeep	26.7	0.552	0.523
	BaselineMERT	20.2	0.502	0.646
	BaselineMIRA	19.4	0.485	0.702
Amharic-to-English	TransformerShallow1	24.0	0.523	0.629
	TransformerShallow2	25.4	0.530	0.614
	TransformerDeep	32.2	0.570	0.539
	BaselineMERT	25.8	0.508	0.633
	BaselineMIRA	23.3	0.497	0.701
English-to-Turkish	TransformerShallow1	7.8	0.418	0.764
	TransformerShallow2	9.4	0.439	0.719
	TransformerDeep	12.6	0.485	0.613
	BaselineMERT	7.8	0.442	0.775
	BaselineMIRA	7.6	0.437	0.796
Turkish-to-English	TransformerShallow1	10.2	0.434	0.803
	TransformerShallow2	11.6	0.444	0.773
	TransformerDeep	16.3	0.498	0.665
	BaselineMERT	10.7	0.465	0.737
	BaselineMIRA	9.1	0.446	0.804

Table 4: Performance results of English-to-Amharic translation using the CACO corpus.

NMT Model	BLEU	BEER	CharacTER
BaselineMERT	20.2	0.502	0.646
BaselineMERT + CACO	21.4	0.503	0.640
TransformerDeep	26.7	0.552	0.523
TransformerDeep + 1×authentic	26.2	0.554	0.517
TransformerDeep + 2×authentic	27.3	0.562	0.505
TransformerDeep + 3×authentic	27.8	0.563	0.501
TransformerDeep + 4×authentic	27.2	0.563	0.504

4.1 Datasets and Preprocessing

We trained our models on the benchmark datasets of the Amharic-English and Turkish-English parallel corpora. We used an Amharic-English parallel corpus provided by the Data and Knowledge Engineering Group at the University of Magdeburg[3] (Gezmu et al., 2021a). For Turkish-English translation, we used the datasets provided by the Conference on Machine Translation[4]. Turkish-English datasets have already been preprocessed with standard Moses tools (Koehn et al., 2007) and are ready for machine translation training. Table 2 shows the number of sentence pairs in each dataset.

We tokenized the English datasets with Moses' tokenizer script; we modified Moses' script to tokenize the Amharic datasets. Next, the Amharic datasets were transliterated with a transliteration scheme, Amharic transliteration for machine translation[5], which is fully discussed in (Gezmu et al., 2021b). Finally, all but the Amharic datasets were true-cased with Moses' true-caser script.

We removed sentence pairs with extreme length ratios of more than one to nine and sentences longer than eighty tokens for the phrase-based SMT baseline. For open vocabulary NMT, the tokens were split into a 32000 word-piece vocabulary as Wu et al. (2016) recommended. We used the word-piece implementation in Google's tensor2tensor library.

4.2 Training and Decoding

The situation of training NMT models is complex because the training of NMT models is usually nondeterministic and hardly ever converges (Popel and

[3] Available at: http://dx.doi.org/10.24352/ub.ovgu-2018-145

[4] Available at: http://data.statmt.org/wmt18/translation-task/preprocessed/

[5] Available at: https://github.com/andmek/AT4MT

Bojar, 2018). Most research in NMT does not specify any stopping criteria. Some mention only an approximate number of days elapsed to train the models (Bahdanau et al., 2015) or the exact number of training steps (Vaswani et al., 2017).

We trained, thus, each NMT model for 250000 steps following Vaswani et al. (2017). For decoding, we used a single model obtained by averaging the last twelve checkpoints. Following Wu et al. (2016), we used a beam search with a beam size of four and a length penalty of 0.6.

4.3 Evaluation

Eventually, translation outputs of the test sets were detokenized, detruecased, and evaluated with a case-sensitive BiLingual Evaluation Understudy (BLEU) metric (Papineni et al., 2002). For consistency, we used the metric's implementation made by Post (2018), sacreBLEU[6]. To fill the limitations of BLEU (Callison-Burch et al., 2006; Reiter, 2018), we also used BEtter Evaluation as Ranking (BEER) (Stanojevic and Sima'an, 2014) and Translation Edit Rate on Character Level (CharacTER) (Wang et al., 2016) metrics. Unlike BLEU and BEER, the smaller the CharacTER score, the better. Moreover, the Amharic outputs were not back transliterated to use these automatic metrics effectively.

5 RESULTS

Table 3 shows the performance results of the three systems plus the baseline system with BLEU, BEER, and CharacTER metrics. The TransformerShallow1 model is the least performing model. Note that the only difference between TransformerShallow1 and TransformerShallow2 is their training batch sizes. The system gained more than one BLEU score by increasing the training batch size from 1024 to 4096. BEER and CharacTER scores also reflect similar improvements. The TransformerDeep system produced the best NMT models.

The baseline system achieved better scores when feature weights were tuned using MERT than batch MIRA. Thus, we took the phrase-based SMT system tuned with MERT as our strong baseline.

The TransformerDeep models outperform the baseline models by more than six BLEU scores in the Amharic-English translation; they gained approximately five more BLEU scores than the baseline models in Turkish-English translation.

[6]Signature BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.9

Though big monolingual corpora are not integral components of NMT, both SMT and NMT can benefit from them. Table 4 shows the results of English-to-Amharic translation using the CACO corpus for language modeling of the baseline phrase-based SMT and back-translating (Sennrich et al., 2016; He et al., 2016; Cheng et al., 2016; Qin, 2020) of the TransformerDeep to produce synthetic training data. Both models gained more than one BLUE score by using CACO. The TransformerDeep model attained the optimum result when we randomly drew three times the size of the original training data from the CACO corpus and translated it to English. Then we mixed the synthetic data with the original (authentic) data to train the new model. Likewise, Deng et al. (2018) reported exciting results using back-translation of huge monolingual corpus for English-Turkish translation.

6 CONCLUSIONS AND FUTURE WORK

Based on the best practices of prior research in this line of work, we conducted NMT for low-resource polysynthetic languages. We used the Transformer-based NMT architecture by tuning the hyper-parameters for low-data conditions. In low-data conditions, using smaller and fewer layers degrades the performance of the Transformer-based systems. Furthermore, unlike RNN-based systems, smaller batch sizes demote their performance. On the other hand, the TransformerDeep models outperform all other models, including the baseline models whether using an extensive monolingual corpus or not.

We suggest the experiments be done for additional language pairs. We also recommend future research for the adaptation of the universal Transformer-based architecture (Dehghani et al., 2019) to low-resource settings.

REFERENCES

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluation the role of bleu in machine translation research. In McCarthy, D. and Wintner, S., editors, *EACL 2006, 11st Conference of the European Chapter of the*

Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy. The Association for Computer Linguistics.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Cherry, C. and Foster, G. F. (2012). Batch tuning strategies for statistical machine translation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 427–436. The Association for Computational Linguistics.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, L. (2019). Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Deng, Y., Cheng, S., Lu, J., Song, K., Wang, J., Wu, S., Yao, L., Zhang, G., Zhang, H., Zhang, P., Zhu, C., and Chen, B. (2018). Alibaba's neural machine translation systems for WMT18. In Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Jimeno-Yepes, A., Koehn, P., Monz, C., Negri, M., Névéol, A., Neves, M. L., Post, M., Specia, L., Turchi, M., and Verspoor, K., editors, *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 368–376. Association for Computational Linguistics.

Ding, S., Duh, K., Khayrallah, H., Koehn, P., and Post, M. (2016). The JHU machine translation systems for WMT 2016. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 272–280. The Association for Computer Linguistics.

Gezmu, A. M., Nürnberger, A., and Bati, T. B. (2021a). Extended parallel corpus for Amharic-English machine translation. *CoRR*, abs/2104.03543.

Gezmu, A. M., Nürnberger, A., and Bati, T. B. (2021b). Neural machine translation for Amharic-English translation. In Rocha, A. P., Steels, L., and van den Herik, H. J., editors, *Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART 2021, Volume 1, Online Streaming, February 4-6, 2021*, pages 526–532. SCITEPRESS.

Gezmu, A. M., Seyoum, B. E., Gasser, M., and Nürnberger, A. (2018). Contemporary Amharic corpus: Automatically morpho-syntactically tagged Amharic corpus. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 65–70, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., and Ma, W. (2016). Dual learning for machine translation. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 820–828.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O., editors, *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 187–197. Association for Computational Linguistics.

Huang, L. and Chiang, D. (2007). Forest rescoring: Faster decoding with integrated language models. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '95, Detroit, Michigan, USA, May 08-12, 1995*, pages 181–184. IEEE Computer Society.

Koehn, P. and Haddow, B. (2009). Edinburgh's submission to all tracks of the WMT 2009 shared task with reordering and speed improvements to moses. In Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J., editors, *Proceedings of the Fourth Workshop on Statistical Machine Translation, WMT@EACL 2009, Athens, Greece, March 30-31, 2009*, pages 160–164. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In Luong, T., Birch, A., Neubig, G., and Finch, A. M., editors, *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In Hearst, M. A. and Ostendorf, M., editors, *Human Language Technology Conference of the North American Chapter of the Association* *for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

Kornai, A. (2013). Digital language death. *PLOS ONE*, 8(10):1–11.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. Association for Computational Linguistics.

Morishita, M., Oda, Y., Neubig, G., Yoshino, K., Sudoh, K., and Nakamura, S. (2017). An empirical study of mini-batch creation strategies for neural machine translation. In Luong, T., Birch, A., Neubig, G., and Finch, A. M., editors, *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 61–68. Association for Computational Linguistics.

Neishi, M., Sakuma, J., Tohda, S., Ishiwatari, S., Yoshinaga, N., and Toyoda, M. (2017). A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In Nakazawa, T. and Goto, I., editors, *Proceedings of the 4th Workshop on Asian Translation, WAT@IJCNLP 2017, Taipei, Taiwan, November 27- December 1, 2017*, pages 99–109. Asian Federation of Natural Language Processing.

Nguyen, T. Q. and Chiang, D. (2018). Improving lexical choice in neural machine translation. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 334–343. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In Hinrichs, E. W. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan*, pages 160–167. ACL.

Östling, R. and Tiedemann, J. (2017). Neural machine translation for low-resource languages. *CoRR*, abs/1708.05729.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Popel, M. and Bojar, O. (2018). Training tips for the transformer model. *Prague Bull. Math. Linguistics*, 110:43–70.

Post, M. (2018). A call for clarity in reporting BLEU scores. In Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Jimeno-Yepes, A., Koehn, P., Monz, C., Negri, M., Névéol, A., Neves, M. L., Post, M., Specia, L., Turchi, M., and Verspoor, K., editors, *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Qin, T. (2020). *Dual Learning*. Springer.

Reiter, E. (2018). A structured review of the validity of BLEU. *Comput. Linguistics*, 44(3).

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 211–221. Association for Computational Linguistics.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Stanojevic, M. and Sima'an, K. (2014). BEER: better evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 414–419. The Association for Computer Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, L., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., and Uszkoreit, J. (2018). Tensor2tensor for neural machine translation. In Cherry, C. and Neubig, G., editors, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 193–199. Association for Machine Translation in the Americas.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wang, W., Peter, J., Rosendahl, H., and Ney, H. (2016). Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 505–510. The Association for Computer Linguistics.

Williams, P., Sennrich, R., Nadejde, M., Huck, M., Haddow, B., and Bojar, O. (2016). Edinburgh's statistical machine translation systems for WMT16. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 399–410. The Association for Computer Linguistics.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

APPENDIX

Common Hyperparameters

Activation data type: float32
Attention dropout: 0.1
Batch shuffle size: 512
First kernel size: 3
Dropout: 0.2
Evaluation frequency in steps: 1000
Evaluation steps: 100
Evaluation timeout minutes: 240
FFN layer: dense relu dense
Initializer: uniform unit scaling
Initializer gain: 1
Kernel height: 3
Kernel width: 1
Label smoothing: 0.1
Layer prepostprocess dropout: 0.1
Learning rate: 0.2
Learning rate cosine cycle steps: 250000
Learning rate decay rate: 1
Learning rate decay scheme: noam
Learning rate decay steps: 5000
Length bucket step: 1.1
Max area height: 1
Max area width: 1
Max length: 256
Memory height: 1
Min length bucket: 8
Mixed precision optimizer init loss scale: 32768
Mixed precision optimizer loss scaler: exponential
MOE hidden sizes: 2048
MOE k: 2
MOE loss coef: 0.001
MOE number experts: 16
MOE overhead evaluation: 2
MOE overhead train: 1
Multiply embedding mode: sqrt depth
Multiproblem label weight: 0.5
Multiproblem mixing schedule: constant
Multiproblem schedule max examples: 10000000
Multiproblem schedule threshold: 0.5
NBR decoder problems: 1
Norm epsilon: 0.000001
Norm type: layer
Optimizer: adam
Optimizer adafactor beta2: 0.999
Optimizer adafactor clipping threshold: 1
Optimizer adafactor decay type: pow
Optimizer adafactor memory exponent: 0.8
Optimizer adam beta1: 0.9
Optimizer adam beta2: 0.997
Optimizer adam epsilon: 0.000000001
Optimizer momentum: 0.9
Position embedding: timing
ReLu dropout: 0.1
Sampling method: argmax
Sampling temp: 1
Schedule: continuous train and evaluate
Scheduled sampling gold mixin prob: 0.5
Scheduled sampling method: parallel
Scheduled sampling number passes: 1
Scheduled sampling warmup schedule: exp
Scheduled sampling warmup steps: 50000
Self attention type: dot product
Split targets max chunks: 100
Standard server protocol: grpc
Symbol modality number shards: 16
Training steps: 250000
Vocabulary divisor: 1
Weight data type: float32