Results of the WMT17 Neural MT Training Task

Ondřej Bojar Jindřich Helcl
Tom Kocmi Jindřich Libovický Tomáš Musil

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
<surname>@ufal.mff.cuni.cz

Abstract

This paper presents the results of the WMT17 Neural MT Training Task. The objective of this task is to explore the methods of training a fixed neural architecture, aiming primarily at the best translation quality and, as a secondary goal, shorter training time. Task participants were provided with a complete neural machine translation system, fixed training data and the configuration of the network. The translation was performed in the English-to-Czech direction and the task was divided into two subtasks of different configurations—one scaled to fit on a 4GB and another on an 8GB GPU card. We received 3 submissions for the 4GB variant and 1 submission for the 8GB variant; we provided also our run for each of the sizes and two baselines. We translated the test set with the trained models and evaluated the outputs using several automatic metrics. We also report results of the human evaluation of the submitted systems.

1 Introduction

Neural machine translation (NMT) has recently replaced the "classical statistical machine translation" and became the dominant research paradigm. A large part of research on NMT is focused on architectural improvements of the neural networks or data preprocessing. However, in practice, the results of an NMT system depends not only on the architecture of the network, but also on the training techniques used to obtain the parameters.

The goal of NMT Training Task[1] is to compare the results of various training techniques applied to a fixed network architecture. We provided task participants with the model specification, training and validation data with a fixed way of data preprocessing. We also listed a few methods as an inspiration for the participants. These methods included (but were not limited to) the following:

Curricula. The basic idea behind this technique (Bengio et al., 2009) is inspired by the fact that humans learn more easily when examples are presented in an ordering from trivial to complex ones. Neural networks could potentially also benefit from such a strategy of increasing task difficulty. This technique includes modifications of the training data in order to converge faster, or more robustly, towards possibly better local optima. Data shuffling, reordering, or back-translation (Sennrich, Haddow, and Birch, 2016a), are all techniques that can have a positive impact on the training.

Optimization algorithms. There are many optimization algorithms that can be employed to training an NMT model, such as Adadelta (Zeiler, 2012) or Adam (Kingma and Ba, 2014). Each method differs in the number of inner trainable parameters and the approach it uses them to perform gradient descent optimization. Better optimization algorithms can improve both the convergence speed and model performance.

Reinforcement learning. A significant improvement in model performance can also be achieved by using variants of the REINFORCE algorithm (Williams, 1992). MIXER (Ranzato et al., 2015), self-critical training (Rennie et al., 2016), or minimum risk training (Shen et al., 2016) all optimize the model directly to maximize the sentence-level BLEU score (Chen and Cherry, 2014) or another sequence-based metric. These

[1] http://www.statmt.org/wmt17/nmt-training-task/

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 525–533
Copenhagen, Denmark, September 711, 2017.

methods deal with the exposure bias problem the traditional cross-entropy approach suffers from.

Multi-task training methods improve the model by training it to perform many tasks at once. Eriguchi, Tsuruoka, and Cho (2017) show that teaching the model how to parse helps the translation. A similar result was achieved by Elliott and Kádár (2017) who teach the network to predict the visual features of an image when translating its caption.

Knowledge distillation. The goal of knowledge distillation is to reduce the size of large trained models to smaller models while maintaining the good performance. There are two ways for employing this technique. First, train a large model and then reduce its size by removing unimportant units (Cun, Denker, and Solla, 1990; He et al., 2014; Han, Mao, and Dally, 2015). Second, train a large "teacher" model (or ensemble of models) and train a smaller "student" network on its outputs (Buciluǎ, Caruana, and Niculescu-Mizil, 2006; Hinton, Vinyals, and Dean, 2015; Kim and Rush, 2016). Both of these methods showed promising results, not only in NMT but in the deep learning field in general.

The rest of the paper is organized as follows. Section 2 describes the software and the model architectures. Details on the used dataset are given in Section 3. We summarize the submitted systems in Section 4. Section 5 presents the results of the submissions and Section 6 discusses them. We conclude in Section 7.

2 The NMT System

We used Neural Monkey (Helcl and Libovický, 2017) as the NMT system for the task. Since the software is still in development, the participants were instructed to use version 0.1.0.[2]

Neural Monkey is an open-source sequence-to-sequence learning toolkit implemented in TensorFlow[3] with simple configuration and great extensibility. Besides the basic attentive NMT pipeline (Bahdanau, Cho, and Bengio, 2014), the toolkit implements a growing collection of techniques related to sequence-to-sequence learning in general.

[2] https://github.com/ufal/neuralmonkey/releases/tag/0.1.0

[3] https://www.tensorflow.org/

GPU Memory	4 GB	8 GB
embedding size	350	600
encoder state size	(2x) 350	(2x) 600
decoder state size	350	600
max sentence length	50	50
BPE merges	30,000	30,000

Table 1: Configuration for 4 and 8 GB models.

Neural Monkey conceptualizes problems of sequence-to-sequence learning as a generic encoder-decoder pipeline, with many types of individual encoders and decoders. In our task, we used the so-called sentence encoder, which maps the input sequence of tokens to a sequence of distributed representations of the tokens, and runs a bidirectional GRU (Cho et al., 2014) network over these vectors. We used the basic recurrent decoder with conditional GRU (Firat and Cho, 2016) units and attention over the encoder.

The used toolkit implements the whole training functionality, including converting token types to indices to the vocabulary, batching, and automatic validation after a specified number of training steps. It also comes with a simple configuration interface which allows the users to design their models without the requirement of writing any code.

We prepared two configurations of the models, one that fits to a GPU with 8GB of memory and a smaller one that fits into 4GB of memory. For specific details about the configuration, refer to Table 1.

3 Data

The dataset used for the NMT Training Task was a subset of the CzEng 1.6 corpus (Bojar et al., 2016). The experiments were to be executed in a constrained fashion, i.e. the participants were not allowed to augment the training corpus by additional data. However, filtering or automatically modifying the provided corpus as well as adding synthetic data (obtained using only this corpus) was permitted.

Prior to the distribution of the corpus, we removed the parts of CzEng 1.6 containing the largest amounts of noise. Specifically we removed the sections named *eu*, *navajo*, *pdfs*, *techdocs*, and *tweets*. We also removed all sentence pairs where one of the sentences contained more

than 40 tokens. The final training dataset contains 48.6 millions sentence pairs. We provided it pre-processed: tokenized and truecased by applying the casing of the lemma identified by Morphodita (Straková, Straka, and Hajič, 2014)[4] to the word form; we did not provide the lemmas to the participants. The corpus was shuffled at the level of sentences, i.e. directly suitable for training with Neural Monkey (that itself does not perform any shuffling unless the whole training data would be loaded to memory). A label file was included with the corpus indicating the original source of each sentence pair, allowing to distinguish e.g. *news* from *subtitles*.

For validation, we used the data from the WMT 2016 news test (*newstest2016*). As the test set, this year's WMT news test (*newstest2017*) was announced and used.

We provided the devset pre-processed in the same way as the training data, i.e. tokenized and truecased by applying the casing of the lemma to the word form.

The test set was not disclosed at all prior to the submission deadline.

The training corpus was analyzed to obtain the byte-pair encoding (BPE; Sennrich, Haddow, and Birch, 2016b) merge file, jointly for English and Czech. The participants were expected to use this BPE merge file in their training. (Neural Monkey, unlike other toolkits, applies BPE splitting internally, to be able to report various scores based on original tokenization and not only based on BPE tokens.) The merge file consisted of 30,000 BPE merges.

4 Training Task Participants

Including secondary and revised versions, we collected six submissions from three external participants: the Air Force Research Laboratory (AFRL), Pavel Denisov, and our students Mostafa Abdou and Vladan Glončák. Additionally, we submitted two of our systems and two baseline runs.

The following paragraphs describe the baseline systems and summarize the techniques used in the submissions for the task.

4.1 Baseline Systems

The baseline systems used the default configurations and datasets as provided to training task participants. The 4GB and 8GB baselines were trained for 60 days, each on a single Nvidia GeForce 1080 GPU.

Among other things, the baseline configuration specifies that tokens appearing only once in the training data are replaced with a special OOV token with probability 0.5.

The Adam optimizer (Kingma and Ba, 2014) with the learning rate of 10^{-4} and mini-batch size of 60 sentences are used. We used L2 regularization with weight of 10^{-8} and gradient clipping with the threshold gradient norm of 1.

The baseline model for 4 GB GPUs achieved the highest validation score after 7.5 epochs of training (47 days). The 8 GB baseline model obtained the highest score after 6.6 epochs (53 days).

4.2 AFRL

The AFRL system is described in another WMT paper by Erdmann, Young, and Gwinnup (2017). They participated in both 4GB and 8GB setups. They used knowledge distillation from an ensemble of models.

The teacher systems were enriched with factors (domain, casing, and subword position information) and trained on a cleaned dataset.

The final (student) system was trained on the news-domain data from the teacher systems dataset, output of ten teacher systems on the same dataset and data from the task training set selected to be most suitable for training a news-domain system.

The original submitted systems trained for about 5 days. We asked the participants to also submit systems trained longer that were not ready in time for the manual evaluation. The AFRL-4GB-REVISED system trained for about 11 days, and the AFRL-8GB-REVISED system trained for about 6 days.

4.3 Pavel Denisov

The system submitted by Pavel Denisov was the default 4GB system trained on 10 million longest sentences in the training dataset. The idea was to make training dataset closer to the validation dataset in the sense of sentence length. The batch size was increased from the default 60 to 90 which is possible when the 4GB model is trained on a larger GPU card. It gave promising validation BLEU score for shorter training duration (approximately 12 hours). The submitted model was trained for 4 days.

[4] http://ufal.mff.cuni.cz/morphodita

4.4 CUNI-4GB-BATCH-DECR

Our students submitted two systems, as described in the paper by Abdou, Glončák, and Bojar (2017). One of the submissions was however using a different BPE file and could not be evaluated among other systems and the other submission (CUNI-4GB-BATCH-DECR) was unfortunately left out from the manual evaluation. We therefore provide at least its automatic scores.

The submission CUNI-4GB-BATCH-DECR uses essentially the baseline configuration but it decreases the batch size from their initial value of 100 by 20 every 48 hours down to the batch size of 20. The motivation is that smaller batch sizes have been shown to converge to flatter optima, i.e. less prone to overfitting, while larger batches make a better use of the GPU. The gradual reduction could theoretically benefit from both: fast training and avoidance of local optima.

4.5 CUNI-4GB-CURRIC

The 4GB submission we provided (CUNI-4GB-CURRIC) is one instance of curriculum learning, namely learning first on short target (Czech) sentences only and gradually adding also longer sentences to the batches as the training progresses. Importantly, the batches in later stages of the training also have to include the short sentences. As a contrastive experiment, we have *only* sorted sentence pairs by the length and the training spectacularly failed.

After one epoch of curriculum learning, we continued the training on the official corpus, keeping its shuffling fixed, for 7M sentence pairs with a relatively small batch size of 20.

More details and further experiments on curriculum learning within one epoch are available in Kocmi and Bojar (2017), who document that curriculum learning can be somewhat helpful according to automatic scoring.

4.6 CUNI-8GB-DOMAIN

The CUNI-8GB-DOMAIN submission is a run forked from the BASELINE-8GB after 3.38 epochs (30.8 days) of training and trained further for 1.5 epochs (9.1 days) on a domain-adapted corpus.

The domain-adapted corpus contains 32.8M parallel lines in total and it was created by concatenating and repeating different types of extracts from the provided training corpus as listed in Table 2.

# Sents	Copies	Corpus
0.25M	4×	News section of training data
2.43M	1×	Top 5% selected by 2-grams
2.43M	1×	Top 5% selected by 4-grams
0.25M	1×	News section again
4.86M	1×	Top 10% selected by 2-grams
2.43M	1×	Top 5% selected by 4-grams
9.72M	1×	Top 20% selected by 2-grams
9.73M	1×	Top 20% selected by 4-grams

Table 2: Composition of the domain-adaptation corpus used in CUNI-8GB-DOMAIN.

Specifically, we used the annotation of the originating domain to extract all news-like sentences. This subset was rather small, only 250k sentence pairs. We therefore used the bilingual cross-entropy difference selection (Axelrod, He, and Gao, 2011) implemented in XenC (Rousseau, 2013) to select 5, 10 and 20% of the original corpus similar in terms of 2-grams and 4-grams to the news section. Presumably, the small news section made it also to these extracts and smaller extracts were probably included in larger extracts, so considering our corpus composition, the same sentences could be reused in the training corpus up to 11 times.

5 Results

The configuration file for translation was provided with the NMT system, to evaluate the model on the devset. The same configuration was used to translate the test set, with the model variables provided by the participants. Except for the chrF3 and METEOR metrics, we detokenized the output of the NMT system using the standard Moses detokenizer[5] and capitalized the first character of the sentence.

5.1 Automatic Scoring of Training Task Systems

For the results of the automatic evaluation, see Table 3.

Since the training time is an important factor in NMT, we suggested that task participants further train their systems and submit new models for automatic scoring. Two more submissions are thus

[5] http://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl

	System	BLEU-dev	BLEU-test	chrF3	METEOR	BEER 2.0	CharacTER
8GB	CUNI-8GB-DOMAIN	**19.18**	**15.2**	**42.59**	**21.60**	**0.487**	**0.683**
	AFRL-8GB-REVISED	18.30	14.8	41.54	20.71	0.478	0.701
	AFRL-8GB	18.15	14.7	41.53	20.95	0.477	0.698
	BASELINE-8GB	17.47	13.8	40.75	20.44	0.472	0.704
4GB	AFRL-4GB-REVISED	**18.37**	**15.2**	**41.90**	**20.92**	**0.480**	**0.693**
	AFRL-4GB	17.58	14.2	40.97	20.64	0.474	0.702
	BASELINE-4GB	16.74	13.7	40.61	20.23	0.472	0.704
	CUNI-4GB-CURRIC	16.24	13.1	39.54	19.93	0.464	0.716
	DENISOV-4GB	15.98	12.6	40.22	20.06	0.452	0.713
	CUNI-4GB-BATCH-DECR	12.98	10.5	36.29	17.85	0.441	0.751

Table 3: Automatic scores for submissions to the WMT17 NMT Training Task.

included in the table, AFRL-4GB-REVISED and AFRL-8GB-REVISED.

BLEU scores for the development set are computed internally by Neural Monkey. For the test set, BLEU was measured on the EuroMatrix evaluation server[6] (we use the BLEU-cased variant of BLEU) as well as BEER 2.0 (Stanojević and Sima'an, 2014) and CharacTER (Wang et al., 2016) scores. We also measured chrF3 (Popović, 2015) and METEOR (Denkowski and Lavie, 2014) scores, both with the same tokenization as in the training data and our NMT system output.

5.2 Learning Curves

We asked participants to provide us with the detailed "events" file as collected by TensorFlow which logs the performance on the common validation set at a fine resolution.

For some techniques, the learning curves cannot be provided, but Figure 1 is a valuable complement to the automatic scoring above. The scores were measured internally by Neural Monkey on the devset after every 2000 batches.

Specifically, we see that the 4GB and 8GB baselines are clearly separated by about the same margin throughout the training and that CUNI-4GB-BATCH-DECR loses a little from the performance later in the training.

Interestingly, DENISOV-4GB seems to very closely follow the performance of BASELINE-8GB, i.e. a much larger setup, but it was unfortunately stopped too early so the obtained score is ultimately worse than both of the baselines. It should be however noted that the learning curves are based on the number of training *sentences* processed, not the number of words. The longer sentences used by DENISOV-4GB have provided the model with more material to learn from, so the score could be artificially inflated.

5.3 Manual Evaluation of Training Task Systems

As announced, the official evaluation of the NMT training task is the manual scoring of the systems submitted at the deadline according to the submission instructions.

We designed training task so that it was in fact subsumed by the WMT17 News Translation Task (Bojar et al., 2017): the training data was a subset of the training data provided for English-to-Czech news task participants and the testset we used the official newstest2017 of WMT. All training task submissions can be therefore seen as regular news task submissions, with the additional constraint of a fixed MT system and further constrained training data.

With the help of WMT17 news task organizers, we included the outputs of training task submissions among the MT outputs of other MT systems for the common manual scoring.[7] Please see Bojar et al. (2017) for details on the judgment technique (direct assessment, DA) and its interpretation.

Table 4 is an extract of the official WMT17 news task results, i.e. Table 7 in Bojar et al. (2017), renaming the systems to match the naming in this paper. The horizontal lines between the systems indicate clusters according to Wilcoxon

[6] http://matrix.statmt.org

[7] Unfortunately, the submission CUNI-4GB-BATCH-DECR, despite being submitted in time, slipped through and was not included in time in the manual evaluation.

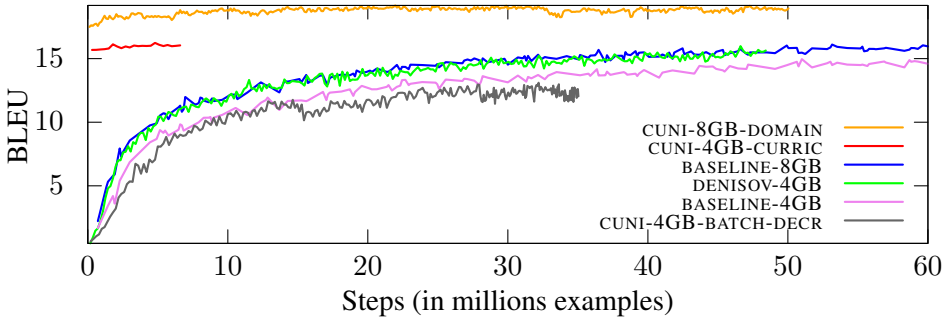

Figure 1: Learning curves for training task submissions (where available). The 8GB and 4GB baseline runs actually ran much longer, to 300M and 380M training steps, resp. CUNI-4GB-CURRIC and CUNI-8GB-DOMAIN curves are only continuations and therefore start higher.

#	Ave %	Ave z	System
~~*1*~~	~~*42.2*~~	~~*-0.141*~~	~~BASELINE-4GB~~
2	44.9	-0.236	CUNI-8GB-DOMAIN
3	42.2	-0.315	AFRL-4GB
	40.7	-0.373	BASELINE-8GB
	40.5	-0.376	AFRL-8GB
4	36.5	-0.486	CUNI-4GB-CURRIC
	36.6	-0.493	DENISOV-4GB

Table 4: Manual evaluation of the training task submissions. For the crossed-out BASELINE-4GB see the text.

rank-sum test at p-level $p \leq 0.05$, the column "#" is the rank of the cluster. The "Ave %" is the average DA score over all evaluated translations by the given system and it reflects the average quality as assessed by human judges against the reference translation on an absolute scale between 0 and 100. The "Ave z" first standardizes each annotator's scores and then averages them. Please see the original paper for a detailed discussion.

The manual evaluation was affected by an unfortunate omission: namely, the baseline-4GB outputs were not included in the standard batches, among other outputs, but they were scored only later, in annotation batches of their own. While the direct assessment annotation technique *in theory* evaluates translation quality on an absolute scale and such evaluations could be in principle comparable among different annotation runs, we see that this does not really work in practice. It is rather unlikely that the 4GB baseline would be significantly better than the 8GB baseline, also taking into account the big difference in BLEU. We thus asked WMT17 news task organizers to remove baseline-4GB from their paper and we do not consider this result in our discussion below.

6 Discussion

Despite the fact that baseline-4GB was not correctly manually evaluated, the manual evaluation allows us to draw some reliable conclusions.

CUNI-8GB-DOMAIN significantly surpassed BASELINE-8GB, confirming that domain adaptation can be very helpful for NMT even with relatively simple adaptation techniques.

AFRL-8GB performed comparably to BASELINE-8GB, and based on the description of the submission, AFRL-8GB was trained for 5 days as 10 models in parallel, which could roughly correspond to the training time of the baseline. While we cannot compare AFRL-4GB and BASELINE-4GB, which would be a very interesting contrastive pair, we know that AFRL-4GB performed equally well (better, but not significantly) as AFRL-8GB. That alone is a good achievement, in line with automatic scoring.

We already knew from automatic scores that the curriculum technique tested by CUNI-4GB-CURRIC is not very effective. We cannot really compare it to BASELINE-4GB but we are not surprised by the relatively low score.

The submission DENISOV-4GB was very interesting, since it achieved the score of the 8GB baseline with just a 4GB model throughout its training, see Figure 1. We hypothesize the reason for the seemingly faster training was that while being presented longer sentences, the system is actually presented more words during training. Nevertheless, the experiment shows that the system is able

to generalize to short sentences from long sentences which does not hold vice versa. Concerning the manual evaluation of DENISOV-4GB, we know that it was trained only for 4 days, so the final quality it reached was not good according to automatic scores. Manual scores in Table 4 confirm this result but it would be very interesting to see what quality would be reached if the training ran much longer.

The point of NMT training task was not to find a single winner but rather to see which techniques are more promising and important for the final performance as well as throughout the training. The short answer is domain adaptation because both CUNI-8GB-DOMAIN and AFRL used it and scored high. Further conclusions are hard to draw because the underlying data and training times differed too much.

For future similar tasks, we recommend to provide already domain-adapted training data and to attempt to keep track of further details about the training, e.g. the number of tokens processed and floating point operations needed.

7 Conclusion

We presented the results of WMT17 Neural MT Training Task, a shared task in optimizing parameters of a given NMT system when translating from English to Czech.

The best results were obtained by a standard domain adaptation technique applied before the training. Ensembling and knowledge distillation is also valuable but current results are not sufficient to assess whether the effort put into the development pays off.

Acknowledgments

This study was supported in parts by the grants SVV 260 453, GAUK 8502/2016, H2020-ICT-2014-1-645442 (QT21) and Charles University Research Programme "Progres" Q18 – Social Sciences: From Multidisciplinarity to Interdisciplinarity.

This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

References

Abdou, Mostafa, Vladan Glončák, and Ondřej Bojar. 2017. Variable mini-batch sizing and pre-trained embeddings. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Axelrod, Amittai, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA. ACM.

Bojar, Ondřej, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. Czeng 1.6: enlarged czech-english parallel corpus with processing tools dockered. In *International Conference on Text, Speech, and Dialogue*, pages 231–238. Springer.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM.

Chen, Boxing and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Cun, Yann Le, John S. Denker, and Sara A. Solla. 1990. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pages 598–605. Morgan Kaufmann.

Denkowski, Michael and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Elliott, Desmond and Ákos Kádár. 2017. Imagination improves multimodal translation. *CoRR*, abs/1705.04350.

Erdmann, Grant, Katherine Young, and Jeremy Gwinnup. 2017. The AFRL WMT17 neural machine translation training task submission. In *Proceedings of the Second Conference on Machine Translation (WMT17)*, Copenhagen, Denmark.

Eriguchi, Akiko, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. *CoRR*, abs/1702.03525.

Firat, Orhan and Kyunghyun Cho. 2016. Conditional gated recurrent unit with attention mechanism. https://github.com/nyu-dl/dl4mt-tutorial/blob/master/docs/cgru.pdf. Published online, version `adbaeea`.

Han, Song, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR'16 best paper award)*.

He, Tianxing, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. 2014. Reshaping deep neural network for fast decoding by node-pruning. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 245–249. IEEE.

Helcl, Jindřich and Jindřich Libovický. 2017. Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, (107):5–17.

Hinton, Geoffrey E., Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Kim, Yoon and Alexander M. Rush. 2016. Sequence-level knowledge distillation. *CoRR*, abs/1606.07947.

Kingma, Diederik P. and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kocmi, Tom and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Recent Advances in Natural Language Processing 2017*.

Popović, Maja. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Rennie, Steven J., Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563.

Rousseau, Anthony. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82.

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

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shen, Shiqi, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Stanojević, Miloš and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206, Doha, Qatar. Association for Computational Linguistics.

Straková, Jana, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland. Association for Computational Linguistics.

Wang, Weiyue, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *WMT*, pages 505–510.

Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Zeiler, Matthew D. 2012. Adadelta: an adaptive learning rate method. *CoRR*, abs/1212.5701.