Samsung R&D Institute Poland submission to WMT20 News Translation Task

Mateusz Krubiński, Marcin Chochowski, Bartłomiej Boczek,
Mikołaj Koszowski, Adam Dobrowolski,
Marcin Szymański, Paweł Przybysz
Samsung R&D Institute Poland

Abstract

This paper describes the submission to the WMT20 shared news translation task by Samsung R&D Institute Poland. We submitted systems for six language directions: English to Czech, Czech to English, English to Polish, Polish to English, English to Inuktitut and Inuktitut to English. For each, we trained a single-direction model. However, directions including English, Polish and Czech were derived from a common multilingual base, which was later fine-tuned on each particular direction. For all the translation directions, we used a similar training regime, with iterative training corpora improvement through back-translation and model ensembling. For the En → Cs direction, we additionally leveraged document-level information by re-ranking the beam output with a separate model.

1 Introduction

Since the Transformer architecture became the standard model in Neural Machine Translation, recent advancements in the field have come from two techniques. The first one is deepening the model by adding more layers, mainly in the encoder part, in order to model more complex dependencies (Raganato and Tiedemann, 2018; Wang et al., 2019a; Wu et al., 2019). This, however, poses problems during the training – too deep models are much harder to train due to the gradient vanishing problem (Zhang et al., 2019). The second technique consists in improving the quality of training data by removing spurious translations (Koehn et al., 2019) and making the data easier to learn through the teacher-student methodology (Hinton et al., 2015; Kim and Rush, 2016; Tan et al., 2019).

In this submission, we decided to leverage both techniques. We deepened the model with a lexical-shortcuts transformer modification. We also iteratively improved the synthetic corpora by training better and better translation models, back-translating and distilling the data in each step.

The remainder of this paper is structured as follows: Section 2 introduces the data used for training, Section 3 shows baseline NMT models and our experiments. In Section 4 we describe our training regime and results. Section 5 is for conclusions.

2 Data

2.1 Data Filtering

We used all the parallel data available in the constrained settings. We filtered the parallel data with a two-step process. First, we used a simple heuristics for general clean-up:

- remove pairs where any of the sentences is longer than 1500 characters
- remove sentences with characters not in the Unicode range specific to a given language pair
- remove pairs based on a length-ratio threshold.

We then de-duplicated the data and used the fast-align[1] tool to filter out pairs basing on the alignment probability between the source and the target (Table 1). For monolingual data, we used only the general clean-up procedure.

2.2 Data Pre-Processing

We used the `normalize-punctuation.perl`[2] script from the Moses package on all the training data. For the En ↔ Iu directions, we used the alignment provided by the organizers, and

[1] github.com/clab/fast_align

[2] github.com/moses-smt/mosesdecoder/scripts/tokenizer/normalize-punctuation.perl

Proceedings of the 5th Conference on Machine Translation (WMT), pages 181–190
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

	Orig.	+ clean-up	+ fast-align
En ↔ Cs	62.5M	61.8M	43.4M
En ↔ Iu	2.6M	1.2M	1.1M
En ↔ Pl	11.2M	10.7M	8.6M

Table 1: Number of sentences in the parallel corpus originally, after simple rule-based cleaning-up, and after filtering out sentence pairs based on alignment probability.

decided to stick to the Inuktitut syllabics, without romanization.

For tokenization and segmentation, we used SentencePiece[3] (Kudo and Richardson, 2018). For the En ↔ Cs and En ↔ Pl directions, we started with a multilingual translation model that was later specialized towards each direction separately. For these 3 languages, we had to use a single, joint vocabulary with $32,000$ pieces and a unigram language model (ULM) tokenization scheme. For the En ↔ Iu directions, we used a joint vocabulary with the ULM tokenization scheme and $16,000$ pieces.

3 NMT System Overwiev

All of our systems are trained with the Marian NMT[4] (Junczys-Dowmunt et al., 2018) framework.

3.1 Baseline systems for En ↔ Cs and En ↔ Pl

We started with strong baselines, i.e. transformer models (Vaswani et al., 2017), which we will now refer to as *transformer-big*. This model consists of 6 encoder layers, 6 decoder layers, 16 heads, a model/embedding dimension of 1024 and a feed-forward layer dimension of 4096.

The model is regularized with a dropout between transformer layers of 0.2 and a label smoothing of 0.1. We also used layer normalization (Lei Ba et al., 2016) and tied the weights of the target-side embedding and the transpose of the output weight matrix, as well as source- and target-side embeddings (Press and Wolf, 2017). Optimizer delay was used to simulate bigger batches, updating weights every 16 batches, Adam (Kingma and Ba, 2015) was used as an optimizer, parametrized with a learning rate of 0.0003 and linear warm-up for the initial $32,000$ updates with subsequent inverted squared decay.

For each language pair, we trained both uni- and bi-directional models. We also examined the effect of using multilingual data to train a quadro-directional model on concatenated En ↔ Cs and En ↔ Pl corpora. The En ↔ Pl corpora were up-sampled 5 times to match size. *<2XX>* tokens were appended to each sentence to indicate the target language. The results on newsdev2020 are presented in Table 2.

	Uni	Bi	Quadro	Quadro- huge
En → Cs	26.1	25.4	24.4	26.0
Cs → En	32.4	31.4	30.5	32.7
En → Pl	26.1	25.4	26.2	27.3
Pl → En	30.0	30.3	31.0	32.3

Table 2: SacreBLEU scores on newsdev2020 for baseline trainings, for various model capacities: uni-directional models, bi-directional models and quadro-directional transformer-big. Quadro-huge stands for the quadro-directional model with the transformer-huge parameters.

	Corpora size	Pre-training	BLEU
En → Pl	0.25M	√	20.4
	0.25M	-	16.5
	0.5M	√	21.8
	0.5M	-	19.4
	8.6M	√	25.1
	8.6M	-	26.1
Pl → En	1M	√	27.1
	1M	-	25.7
	8.6M	√	29.1
	8.6M	-	30.0

Table 3: SacreBLEU scores on newsdev2020 for En ↔ Pl trainings, with and without pre-training. 8.6M corpora size means all the available training data was used.

3.2 Baseline system for En ↔ Iu

As the parallel corpora for En ↔ Iu are significantly smaller than for the other pairs, we decided to start with a transformer model with a smaller number of parameters i.e. *transformer-base*. All our base models were bi-directional.

The model consists of 6 encoder layers, 6 decoder layers, 8 heads, a model/embedding dimension of 512 and a feed-forward layer dimension of 2048. We examined the effect of vocabulary size on the model quality, and obtained the best results for the vocabulary size of $16,000$ (Table 4). Basing on our previous experience, we also examined an

[3] github.com/google/sentencepiece

[4] github.com/marian-nmt/marian

	Vocab size	BLEU
En → Iu	16k	15.1
	32k	15.1
	64k	15.0
Iu → En	16k	28.3
	32k	27.9
	64k	27.6

Table 4: SacreBLEU scores on newsdev2020 for En ↔ Iu bi-directional trainings, for different sizes of the sentencepiece vocabulary.

unbalanced encoder/decoder configuration with a deeper encoder (8 layers) and a more shallow decoder (4 layers). The result was 28.3 (+0.0) for Iu → En and 15.3 (+0.2) for En → Iu, compared to the base case. We used this model as a reference for the following experiments.

3.3 Multilingual Denoising Pre-training

Liu et al. (2020) recently proposed a method for pre-training sequence-to-sequence models with an auto-encoder-based denoising objective. Pre-training a complete encoder-decoder model allows for later direct fine-tuning on the translation objective, with parallel corpora. In our experiment, we sampled 250M sentences from CommonCrawl for Czech, English and Polish (i.e. 750M in total). During training, we randomly cropped up to 25% tokens from each sentence, and taught the model to predict the original sequence. We used the same architecture as in baseline trainings. Next, we used the best checkpoint to warm-start training on the parallel data. Table 3 presents our results for varying sizes of the training corpus (the smaller corpus is a random subset of the parallel data). We observe that, although our implementation works well for low-resource setting, it leads to quality drop when all the parallel data is used. Accordlingly, we used this pre-training method only for the En ↔ Iu directions.

3.4 Lexical Shortcuts

Since our quadro-directional model showed promising results, we decided to try to examine the effect of deepening and enlarging the model. We increased the feed-forward layer dimension to 8192, and the number of encoder layers to 12. The rest of the parameters is the same as in *transformer-big*. He et al. (2019) demonstrated that, with a fixed number of layers, it was more efficient to have a deeper encoder than decoder. It also makes decoding for back-translation much faster. To help with gradient propagation, we implemented Lexical Shortcuts (Emelin et al., 2019) in the encoder. We used the feature-fusion version of the gating mechanism. The results are summarized in the Quadro-huge column in Table 2. This model outperformed the baseline in all the directions, except one. We decided to use this system in further trainings.

	Pl	En	Cs
Newscrawl	3.7M	230M	80.5M
+ Moore-Lewis	96.5M	-	-

Table 5: Number of sentences in monolingual datasets after clean-up and domain-based filtering.

3.5 Back-Translation with Language Model

Back-translation (Sennrich et al., 2016) is a common strategy of utilizing monolingual data in training NMT systems. For English and Czech, the amount of monolingual in-domain data in the Newscrawl data set is big enough, so for this language pair we used only the monolingual set. Yet for Polish, the Newscrawl is very limited in size, hence we decided to use Moore-Lewis filtering (Moore and Lewis, 2010) to extract in-domain data from CommonCrawl.

With this additional monolingual corpus, we had over 80M in-domain news sentences for each language (Table 5). We used those monolingual datasets to train an in-domain RNN-style language model for each of the three languages, using the same common vocabulary as the one in the translation models. This allowed us to easily ensemble this language model with a translation model during decoding, as described in Gulcehre et al. (2015). For each iteration of the back-translation, we used an ensemble of the top 4 NMT models available w.r.t. the dev-set score for the particular direction and the in-domain language model. The weights of the models were optimized through a grid-search.

3.6 Noisy Channel Model Reranking

Re-ranking the beam output is a method used to improve translation quality by the re-scoring hypothesis from a forward model. The noisy channel model (Yee et al., 2019) approach was used with success by Facebook in their submission to the WMT19 news translation task (Ng et al., 2019). Based on the Bayes' rule, given a target sequence y and a source sentence x, for every hypothesis from the beam output, we calculate

$$\log P(y \mid x) + \lambda_1 \log P(x \mid y) + \lambda_2 \log P(y)$$

and use this score to re-rank the beam outputs. We model $P(y \mid x)$ with the forward model, $P(x \mid y)$ with the backward model and $P(y)$ with the language domain model. The weights λ_1 and λ_2 are tuned on the dev-set.

When we used this method for our baseline unidirectional systems, we noticed significant BLEU improvements: 26.9 (+0.8) on newsdev2020 for the En → Pl direction. However, there was no improvement when applied to translations produced with strong ensembles of both the domain language models and the translation models, trained on the back-translated data. In our final submission, we used this method only for the Iu → En and En → Iu directions.

3.7 Multi-Agent Dual Learning

In our submission, we used the simplified version of Multi-Agent Dual Learning (MADL) (Wang et al., 2019b), proposed in Kim et al. (2019), to generate additional training data from the parallel corpus. We generated n-best translations of both the source and the target sides of the parallel data, with strong ensembles of, respectively, the forward and the backward models. Next, we picked the best translation from among n candidates w.r.t. the sentence-level BLEU score. Thanks to these steps, we tripled the number of sentences by combining three types of datasets:

1. original source – original target,
2. original source – synthetic target,
3. synthetic source – original target,

where the synthetic target is the translation of the original source with the forward model, and the synthetic source is the translation of the original target with the backward model.

3.8 Document Level Reranking

For the En ↔ Cs translation directions, the training data is aligned on the document level. To make use of this information, we implemented the method presented in Voita et al. (2019). The method assumes one has access to consecutive tuples of sentences in the target language. Using the backward and forward models, one should translate the tuples with the sentence-level based systems, and then train the model to predict the original tuple, basing on the two-way translated data. As we already have access to the document-level aligned translations from the CzEng 2.0 corpus (Kocmi et al., 2020), we could do the translation just once. We experimented only with the En → Cs direction. We selected tuples of 4 consecutive sentences in English, translated each sentence independently, and glued the translations back together. We used a special token to indicate the end of the sentence. See Table 9 in the Appendix for an example of the training data. However, when we utilized this model to "repair" the newsdev2020 dev-set translations, we noticed a quality drop. We decided to try a different approach, and used the document-level repair model to re-rank the beam output. The procedure is similar to a greedy search for the best path through n-best lists of forward model translations. It is described with Algorithm 1.

Algorithm 1 Document Level Reranking

```
Input: $\{trn^b(s_j)\}$ - $n$-best list ($b = 1..N$) with
    translations of sentence $s_j$
Input: $L_{repair}(\{sa_j\}_{j=1..4}, \{sb_j\}_{j=1..4})-$ likeli-
    hood computed with repair model for two 4-
    sentence sequences
Output: Re-ranked translations $rep$
for all paragraph in test-set do
    $i = 0$
    for all sentence $s_i$ in paragraph do
        if $i < 4$ then
            $rep_i = trn^1(s_i)$
        else
            $seq_1 = rep_{i-3}, \ldots, rep_{i-1}, trn^1(s_i)$
            $seq_b = rep_{i-3}, \ldots, rep_{i-1}, trn^b(s_i)$
            $rep_i = \arg\max_{b=1..N} L_{repair}(seq_1, seq_b)$
        end if
        $i += 1$
    end for
end for
```

Although on the dev-set we didn't see much difference in the BLEU score, manual inspection showed some promising results. We decided to apply this method to our best-scoring system and saw a 0.1 improvement in the BLEU score on the test-set.

3.9 Post-Processing

For all the translation directions we participated in, we normalized the system outputs with a series of regular expressions:

- substitute English quotation marks (“ ... ”) with Czech/Polish ones („ ... ”),
- if a source starts/ends with a quotation mark, we make sure so does the translation,
- remove word repetitions,
- replace consecutive sequences of white-spaces with a single one,
- if a source ends with a punctuation mark (e.g. ?.!), we substitute the last character of the translation with it,
- replace three consecutive dots with an ellipsis,
- replace hyphens with en dashes.

4 Results

4.1 English → Polish

The model for the English → Polish direction was derived from the multilingual quadro-huge model – similarly to the other models for directions with Polish, Czech or English. The successive steps and respective BLEU scores are reported in Table 6.

We started with fine-tuning the quadro-directional model on the parallel data for the specific direction. Next, we used an ensemble of our best models to back-translate Newscrawl 2018 and 2019, we filtered it (3.5M sentences) and merged with the parallel corpus (8.6M). The fine-tuning gave us +1.5 BLEU improvement. We were able to achieve an additional +0.9 BLEU with the rule-based post-processing (see above). In the next step, we used the MADL procedure to generate additional data. To further increase the amount of data and its variability, we picked the top 2 best translations, according to the sentence-level BLEU in the distillation process – instead of choosing just one. Again, we up-sampled the original parallel corpus twice. This procedure gave additional 52M sentences (a 6-fold increase).

We back-translated all the monolingual in-domain data (i.e. 89M after filtration) and used both corpora to fine-tune the next generation model. We augmented the data by randomly masking up to 10% of the input tokens with a random punctuation mark, and observed yet another performance boost. Using all these corpora, we trained another model from scratch, hoping to get a less correlated model. Although the fresh model performance was poorer than the previous best (30.2 BLEU vs. 31.6 BLEU), the grid-search ensemble optimization included it in the best ensemble. As a final step, we used Moore-Lewis filtering to choose 1M Newscrawl sentences that were closest to the concatenated newsdev2020 and newstest2020. We translated them with the best ensemble, and used it to fine-tune our best-performing model. Again, we ran ensemble optimization including this model into the models reservoir. The optimal ensemble was the one we submitted as the primary system.

	newsdev2020	
System	**En → Pl**	**Pl → En**
Quadro-huge	27.3	32.3
+ finetune	27.4	32.8
+ ensemble	28.7	32.9
+ BT	28.9	33.7
+ post-process	29.8	-
+ ensemble	30.7	34.1
+ BT2 & MADL	31.4	34.4
+ masking	31.6	-
FRESH	30.2	32.9
+ ensemble	32.2	34.9
+ post-process	-	35.0
+ test-dev tune	32.2	35.1
+ ensemble	32.4	35.4
	newstest2020	
WMT’20 SUBMISSION	**27.6**	**34.3**

Table 6: Successive improvements in the BLEU scores on the English → Polish and Polish → English directions, computed with SacreBLEU.

4.2 Polish → English

For the Polish to English direction, we proceeded similarly to our solution for English to Polish. We started with the quadro-huge model. We back-translated Newscrawl 2018, filtered it (12M sentences) and merged with the parallel corpus (8.6M). We kept on training the fine-tuned quadro-huge model, increasing the performance by 0.9 BLEU. We used the same MADL procedure as before, distilling 2 best translations for each source sentence. We also back-translated Newscrawl 2007-2017 (144M) and merged it with the MADL corpus. With this corpus, single model performance increased by +0.7 BLEU. We used the same corpus to train a fresh model. Similarly to the English to Polish direction, the fresh model performed poorer

(32.9 BLEU) than the fine-tuned one (34.4 BLEU), but – again – in ensemble it gave additional improvement. Finally, we fine-tuned on the 1M corpora filtered out from Newscrawl in the domain of the concatenated newsdev2020 and newstest2020, and ensembled for the final submission.

4.3 English → Czech

We started with fine-tuning the quadro-huge model with only English to Czech parallel data and ensembling several models into one. This model specialization gave us +1.4 BLEU. Next, we back-translated Newscrawl 2018 and 2019 in two flavors: normally, and with adding Gumbel noise (`--output-sampling` in Marian). Then, we filtered the result (see section 2.1), obtaining 35M sentences. With this additional corpus, the single model performance improved by 0.9 BLEU. In contrast to the Cs → En direction, using back-translations from CzEng 2.0 seemed to hurt the model performance.

In the next iteration, we produced the MADL corpus (120M) and merged it with back-translated Newscrawl 2009-2017 (102M, with and without noise) and used this data to train yet another model. Finally, we ensembled this model with models trained from scratch and fine-tuned on the 1M from Newscrawl common with the concatenated newsdev2020 and newstest2020. Before the last step – document level re-ranking – we used the sentence-splitter from NLTK (Bird et al., 2009) to pre-process the testset. It was required because of our systems being trained with sentence-level data and in newstest2020 some of the segments contain multiple sentences. We translated the pre-processed testset with the best ensemble, re-ranked on the document level and finally glued back the translations together. The document level re-ranking gave us -0.1 BLEU on the dev-set but +0.1 on the test-set.

4.4 Czech → English

Again, the specialization of the quadro-huge model with only Czech to English data gave us almost 1 BLEU gain in performance. Next, we back-translated Newscrawl 2018 and 2019 and filtered it with our pipeline, obtaining 49M sentences. We added the Newscrawl translations from CzEng 2.0 (79M) and the original filtered parallel corpus (43M), ending up with 171M parallel sentences as our training set. Using this data, we improved the single model performance by 3.6 BLEU. Fine-tuning on the MADL corpus (120M) and the back-translated Newscrawl 2017 (25M) gave additional +1.2 BLEU on the single model. Finally, we ensembled them with a model trained from scratch and fine-tuned on the 1M sentences from Newscrawl that were similar to the concatenated newsdev2020 and newstest2020 w.r.t. the Moore-Lewis score. For sentence splitting, we used the same splitter as for En → Cs. Results on the previous test-sets of the final systems for the En ↔ Cs directions, without document level re-ranking, in the Appendix (Table 10).

	newsdev2020	
System	**En → Cs**	**Cs → En**
Quadro-huge	26.0	32.7
+ finetune	26.5	33.5
+ ensemble	27.3	33.8
+BT	27.4	37.4
+ ensemble	28.5	37.7
+ post-process	-	37.8
+ BT2 & MADL	28.8	38.6
FRESH	27.0	35.6
+ ensemble	29.1	38.7
+ test-dev tune	29.1	39.0
+ ensemble	29.4	39.7
+ post-process	31.3	-
+ doc-level re-rank	31.2	-
	newstest2020	
WMT’20 SUBMISSION	**36.5**	**28.5**

Table 7: Successive improvements in the BLEU scores on the English → Czech and Czech → English directions, computed with SacreBLEU.

4.5 English ↔ Inuktitut

In contrast to all the other directions, for Inuktitut we had much less monolingual data (10k after cleaning) than bitext (1.1M). In the first step, we back-translated the monolingual data with beam 10, and kept all the possible variants (0.1M sentences). We also back-translated the English Europarl v10 corpus (2M), because we believed it to help with the Hansard (Joanis et al., 2020) part of the dev- and test-sets. We merged it with the two-directional parallel data (2.2M) and trained a bi-directional model, from scratch. We used it for the general first iteration of the MADL corpus (6.6M), and used all of the data to once more train a model from scratch. Here we examined the effect of pre-training. We used the sample (10M)

	newsdev2020	
System	**En → Iu**	**En → Iu**
Baseline	15.3	28.3
+ BT	15.5	29.9
+ MADL	15.5	30.4
+ masked mono	15.8	30.5
+ transformer-big	15.8	32.2
+ fine-tune	15.8	32.5
+ ensemble	16.2	32.7
+ MADL2	15.8	32.7
+ ensemble	16.3	32.9
+ noisy channel	16.4	33.2
	newstest2020	
WMT'20 SUBMISSION	**11.0**	**25.6**

Table 8: Successive improvements in the BLEU scores on the English → Inuktitut and Inuktitut → English directions, computed with SacreBLEU.

from the English Newscrawl 2019 and the Inuktitut part of the parallel data, up-sampled 5 times (5.5M). With the pipeline approach, fine-tuning on bitext was giving us similar results as training on bitext from scratch. Nevertheless, we were however able to achieve some improvement, when training a fresh model on the merged parallel and noised monolingual data. We were able to achieve further improvement with increased model size – 1024 embedding dimension, 4096 forward dimension and 16 heads.

Next, we started fine-tuning on each direction independently, using the parallel data for En → Iu, and 20-times up-sampled the parallel data (22M) together with the back-translated Newscrawl 2018 and 2019 (48M) for Iu → En. Then, we used an ensemble of models to once again generate the MADL corpus, use it to fine-tune the unidirectional models and the ensemble once again. We used the Noisy Channel Reranking method and saw some improvement on both the dev-set and the test-set.

5 Conclusions

In this paper, we have described the submission to the WMT20 shared news translation task by Samsung R&D Institute Poland. All submitted systems were constrained and utilized only the permitted data. With our approach, we were able to leverage two important techniques that improve the translation quality. One method was deepening the model, while still being able to train it effectively. The other one was filtering and improving the quality of the training data and producing high quality synthetic data. Our iterative approach of improving the training data and improving the translation model proved to be successful, showing gradual increase in the BLEU scores.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Denis Emelin, Ivan Titov, and Rico Sennrich. 2019. Widening the Representation Bottleneck in Neural Machine Translation with Lexical Shortcuts. In *Proceedings of the Fourth Conference on Machine Translation*, pages 102–115, Florence, Italy. Association for Computational Linguistics.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On Using Monolingual Corpora in Neural Machine Translation. *arXiv e-prints*, page arXiv:1503.03535.

Tianyu He, Xu Tan, and Tao Qin. 2019. Hard but Robust, Easy but Sensitive: How Encoder and Decoder Perform in Neural Machine Translation. *arXiv e-prints*, page arXiv:1908.06259.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, page arXiv:1503.02531.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations*, San Diego, California, USA.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv e-prints*, page arXiv:1607.06450.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *arXiv e-prints*, page arXiv:2001.08210.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019a. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019b. Multi-agent dual learning. In *International Conference on Learning Representations*.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Depth growing for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5558–5563, Florence, Italy. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

A Appendix

source	”To tys vymyslel mihotavé reflektory?” <SEP> ”...Ne. <SEP> V nanouffu nejsem moc dobrá. <SEP> Přišel na ně můj přítel z Londýna.
target	”A vymyslel jste taky ty reflektorky?” <SEP> ”Ne. <SEP> V nanotechnologii tak dobrý nejsem. <SEP> S tím přišel jeden můj známý z Londýna.
source	A na čest svého domu, prohlašuji, že můj milovaný Robert.. . . -Určitě? <SEP> Radši se podepiš s Jaimem Lannisterem, Králokatem. <SEP> To město je nudný. <SEP> Prosím, Andrewe.
target	”A já prohlašuji na čest svého rodu, že můj milovaný bratr Robert...” <SEP> Dej tam serem Jaimem Lannisterem, Králokatem. <SEP> Tohle město páchne... <SEP> Prosím!
source	Řekla jsem: ”Čí byl nápad?” <SEP> Jejich modré oči byly jasné jako plavecký bazén. <SEP> ”To přišel točení Ernesto.” vzdálený Dezertér nebo lhář. <SEP> Byli jste dost dobří přátelé?”
target	”Čí to byl nápad?” zeptala jsem se. <SEP> Podíval se na mě zpříma a jeho modré oči byly průzračné jak studánky. <SEP> ”Earnesto s tím přišel.” <SEP> ”Byli jste dobří kamarádi?”

Table 9: Example of the training data used to train the document-level re-rank model. Target is a quadruple of consecutive sentences extracted from the CzEng 2.0 parallel corpus. Source is a translation of the matching English sequence, produced on the sentence level.

		En → Cs	**Cs → En**
newstest2019	WMT‘19 best	29.9	-
	SRPOL‘20	31.3 (+1.4)	-
newstest2018	WMT‘18 best	26.0	33.9
	SRPOL‘20	27. 4 (+1.4)	35.3 (+1.4)
newstest2017	WMT‘17 best	26.1	30.9
	SRPOL‘20	27.7 (+1.6)	35.1 (+4.2)

Table 10: SacreBLEU scores of the final systems for the En ↔ Cs directions, without document level re-ranking, on test-sets from previous years.