Towards Reasonably-Sized Character-Level Transformer NMT by Finetuning Subword Systems

Jindřich Libovický and **Alexander Fraser**
Center for Information and Speech Processing
Ludwig Maximilian University of Munich
Munich, Germany
{libovicky, fraser}@cis.lmu.de

Abstract

Applying the Transformer architecture on the character level usually requires very deep architectures that are difficult and slow to train. These problems can be partially overcome by incorporating a segmentation into tokens in the model. We show that by initially training a subword model and then finetuning it on characters, we can obtain a neural machine translation model that works at the character level without requiring token segmentation. We use only the vanilla 6-layer Transformer Base architecture. Our character-level models better capture morphological phenomena and show more robustness to noise at the expense of somewhat worse overall translation quality. Our study is a significant step towards high-performance and easy to train character-based models that are not extremely large.

1 Introduction

State-of-the-art neural machine translation (NMT) models operate almost end-to-end except for input and output text segmentation. The segmentation is done by first employing rule-based tokenization and then splitting into subword units using statistical heuristics such as byte-pair encoding (BPE; Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018).

Recurrent sequence-to-sequence (S2S) models can learn translation end-to-end (at the character level) without changes in the architecture (Cherry et al., 2018), given sufficient model depth. Training character-level Transformer S2S models (Vaswani et al., 2017) is more complicated because the self-attention size is quadratic in the sequence length.

In this paper, we empirically evaluate Transformer S2S models. We observe that training a character-level model directly from random initialization suffers from instabilities, often preventing it from converging. Instead, we propose finetuning subword-based models to get a model without explicit segmentation. Our character-level models show slightly worse translation quality, but have better robustness towards input noise and better capture morphological phenomena. Our approach is important because previous approaches have relied on very large transformers, which are out of reach for much of the research community.

2 Related Work

Character-level decoding seemed to be relatively easy with recurrent S2S models (Chung et al., 2016). But early attempts at achieving segmentation-free NMT with recurrent networks used input hidden states covering a constant character span (Lee et al., 2017). Cherry et al. (2018) showed that with a sufficiently deep recurrent model, no changes in the model are necessary, and they can still reach translation quality that is on par with subword models. Luong and Manning (2016) and Ataman et al. (2019) can leverage character-level information but they require tokenized text as an input and only have access to the character-level embeddings of predefined tokens.

Training character-level transformers is more challenging. Choe et al. (2019) successfully trained a character-level left-to-right Transformer language model that performs on par with a subword-level model. However, they needed a large model with 40 layers trained on a billion-word corpus, with prohibitive computational costs.

In the most related work to ours, Gupta et al. (2019) managed to train a character-level NMT with the Transformer model using Transparent Attention (Bapna et al., 2018). Transparent attention attends to all encoder layers simultaneously, making the model more densely connected but also more computationally expensive. During training, this improves the gradient flow from the decoder to the encoder. Gupta et al. (2019) claim that Trans-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2572–2579,
November 16–20, 2020. ©2020 Association for Computational Linguistics

tokeni- zation	The␣cat␣sleeps␣on␣a␣mat. _The␣_cat␣_sleeps␣_on␣_a␣_mat␣.
32k	_The␣_cat␣_sle␣eps␣_on␣_a␣_mat␣.
8k	_The␣_c␣at␣_s␣le␣eps␣_on␣_a␣_m␣at␣.
500	_The␣_c␣at␣_s␣le␣ep␣s␣_on␣_a␣_m␣at␣.
0	_␣T␣h␣e␣_␣c␣a␣t␣_␣s␣l␣e␣e␣p␣s␣_␣o␣n␣_␣ a␣_␣m␣a␣t␣.

Table 1: Examples of text tokenization and subword segmentation with different numbers of BPE merges.

parent Attention is crucial for training character-level models, and show results on very deep networks, with similar results in terms of translation quality and model robustness to ours. In contrast, our model, which is not very deep, trains quickly. It also supports fast inference and uses less RAM, both of which are important for deployment.

Gao et al. (2020) recently proposed adding a convolutional sub-layer in the Transformer layers. At the cost of a 30% increase of model parameter count, they managed to narrow the gap between subword- and character-based models by half. Similar results were also reported by Banar et al. (2020), who reused the convolutional preprocessing layer with constant step segments Lee et al. (2017) in a Transformer model.

3 Our Method

We train our character-level models by finetuning subword models, which does not increase the number of model parameters. Similar to the transfer learning experiments of Kocmi and Bojar (2018), we start with a fully trained subword model and continue training with the same data segmented using only a subset of the original vocabulary.

To stop the initial subword models from relying on sophisticated tokenization rules, we opt for the loss-less tokenization algorithm from SentencePiece (Kudo and Richardson, 2018). First, we replace all spaces with the _ sign and do splits before all non-alphanumerical characters (first line of Table 1). In further segmentation, the special space sign _ is treated identically to other characters.

We use BPE (Sennrich et al., 2016) for subword segmentation because it generates the merge operations in a deterministic order. Therefore, a vocabulary based on a smaller number of merges is a subset of a vocabulary based on more merges estimated from the same training data. Examples of the segmentation are provided in Table 1. Quantitative effects of different segmentation on the data are presented in Table 2, showing that character sequences are on average more than 4 times longer than subword sequences with 32k vocabulary.

# merges	segm. / sent.	segm. / token	avg. unit size	
			en	de
32k	28.4	1.3	4.37	4.51
16k	31.8	1.4	3.95	3.98
8k	36.2	1.6	3.46	3.50
4k	41.5	1.9	3.03	3.04
2k	47.4	2.1	2.66	2.67
1k	54.0	2.4	2.32	2.36
500	61.4	2.7	2.03	2.08
0	126.1	5.6	1.00	1.00

Table 2: Statistics of English-German parallel data under different segmentations.

We experiment both with deterministic segmentation and stochastic segmentation using BPE Dropout (Provilkov et al., 2020). At training time, BPE Dropout randomly discards BPE merges with probability p, a hyperparameter of the method. As a result of this, the text gets stochastically segmented into smaller units. BPE Dropout increases translation robustness on the source side but typically has a negative effect when used on the target side. In our experiments, we use BPE Dropout both on the source and target side. In this way, the character-segmented inputs will appear already at training time, making the transfer learning easier.

We test two methods for finetuning subword models to reach character-level models: first, direct finetuning of subword models, and second, iteratively removing BPE merges in several steps in a curriculum learning setup (Bengio et al., 2009). In both cases we always finetune models until they are fully converged, using early stopping.

4 Experiments

To cover target languages of various morphological complexity, we conduct our main experiments on two resource-rich language pairs, English-German and English-Czech; and on a low-resource pair, English-Turkish. Rich inflection in Czech, compounding in German, and agglutination in Turkish are examples of interesting phenomena for character models. We train and evaluate the English-German translation using the 4.5M parallel sen-

		From random initialization								Direct finetuning from			In steps
		32k	16k	8k	4k	2k	1k	500	0	500	1k	2k	
en-de	BLEU	26.9	26.9	26.7	26.4	26.4	26.1	25.8	22.6	25.2	25.0	25.0	24.6
		-0.03	*	-0.20	-0.47	-0.50	-0.78	-1.07	-4.29	-1.65 / -0.58	-1.88 / -1.10	-1.85 / -0.78	-2.23 / -1.16
	chrF	.569	.568	.568	.568	.564	.564	.561	.526	.559	.559	.559	.556
	METEOR	47.7	48.0	47.9	47.8	47.9	47.7	47.6	45.0	46.5	46.4	46.4	46.3
	Noise sens.	-1.07	-1.06	-1.05	-1.03	-1.01	-1.02	-1.00	-0.85	-0.99	-0.99	-0.99	-0.88
	MorphEval	90.0	89.5	89.4	89.6	89.8	90.0	89.2	89.2	89.9	90.3	89.3	90.1
de-en	BLEU	29.8	30.1	29.6	29.3	28.6	28.5	28.1	26.6	28.2	28.4	27.7	28.2
		-0.34	*	-0.53	-0.83	-1.62	-1.67	-1.99	-3.51	-1.94 / +0.05	-1.76 / -0.10	-2.52 / -0.90	-1.89 / +0.10
	chrF	.570	.573	.568	.567	.562	.558	.558	.543	.562	.564	.559	.563
	METEOR	37.1	37.4	37.2	37.2	36.9	37.2	36.9	35.1	36.4	36.4	36.0	36.4
	Noise sens.	-0.45	-0.43	-0.41	-0.42	-0.43	-0.42	-0.41	-0.30	-0.37	-0.37	-0.37	-0.36
en-cs	BLEU	21.1	20.8	20.9	20.6	20.1	20.0	19.5	18.2	19.2	19.3	19.4	19.3
		*	-0.25	-0.13	-0.46	-0.96	-1.05	-1.54	-2.82	-1.81 / -0.27	-1.73 / -0.68	-1.64 / -0.68	-1.81 / -0.27
	chrF	.489	.488	.490	.487	.483	.482	.478	.465	.477	.476	.478	.477
	METEOR	26.0	25.8	26.0	25.8	25.7	25.7	25.4	24.6	25.2	25.2	25.2	25.1
	Noise sens.	-1.03	-1.01	-1.01	-1.01	-0.94	-0.93	-0.91	-0.79	-0.82	-0.84	-0.87	-0.82
	MorphEval	83.9	84.6	83.7	83.9	84.3	84.4	84.7	82.1	84.7	84.1	81.9	81.3
en-tr	BLEU	12.6	13.1	12.7	12.8	12.5	12.3	12.2	12.4	12.0	12.6	12.3	11.6
		-0.48	*	-0.36	-0.29	-0.58	-0.77	-0.86	-.0.73	-1.08 / -0.22	-0.85 / -0.08	-0.82 / -0.53	-1.54 / -0.68
	chrF	.455	.462	.459	.456	.457	.457	.455	.461	.456	.460	.459	.450
	Noise sens.	-0.99	-0.91	-0.90	-0.87	-0.85	-0.83	-0.79	-0.62	-0.66	-0.66	-0.66	-0.68

Table 3: Quantitative results of the experiments with deterministic segmentation. The left part of the table shows subword-based models trained from random initialization, the right part shows character-level models trained by finetuning. The yellower the background color, the better the value. Small numbers denote the difference from the best model, $*$ is the best model. For finetuning experiments (on the right) we report both difference from the best model and from the parent model. Validation BLEU score are in in the Appendix.

tences of the WMT14 data (Bojar et al., 2014). Czech-English is trained on 15.8M sentence pairs of the CzEng 1.7 corpus (Bojar et al., 2016) and tested on WMT18 data (Bojar et al., 2018). English-to-Turkish translation is trained on 207k sentences of the SETIMES2 corpus (Tiedemann, 2012) and evaluated on the WMT18 test set.

We follow the original hyperparameters for the Transformer Base model (Vaswani et al., 2017), including the learning rate schedule. For finetuning, we use Adam (Kingma and Ba, 2015) with a constant learning rate 10^{-5}. All models are trained using Marian (Junczys-Dowmunt et al., 2018). We also present results for character-level English-German models having about the same number of parameters as the best-performing subword models. In experiments with BPE Dropout, we set dropout probability $p = 0.1$.

We evaluate the translation quality using BLEU (Papineni et al., 2002), chrF (Popović, 2015), and METEOR 1.5 (Denkowski and Lavie, 2014). Following Gupta et al. (2019), we also conduct a noise-sensitivity evaluation to natural noise as introduced by Belinkov and Bisk (2018). With probability p words are replaced with their variants from a misspelling corpus. Following Gupta et al. (2019), we assume the BLEU scores measured with input can be explained by a linear approximation with intercept α and slope β using the noise probability p: BLEU $\approx \beta p + \alpha$. However, unlike them, we report the relative translation quality degradation β/α instead of only β. Parameter β corresponds to absolute BLEU score degradation and is thus higher given lower-quality systems, making them seemingly more robust.

To look at morphological generalization, we evaluate translation into Czech and German using MorphEval (Burlot and Yvon, 2017). MorphEval consists of 13k sentence pairs that differ in exactly one morphological category. The score is the percentage of pairs where the correct sentence is preferred.

5 Results

The results of the experiments are presented in Table 3. The translation quality only slightly decreases when drastically decreasing the vocabulary. However, there is a gap between the character-

Direction	Determ. BPE		BPE Dropout	
	BLEU	chrF	BLEU	chrF
en-de	25.2	.559	24.9	.560
de-en	28.2	.562	28.5	.564
en-cs	19.3	.447	19.5	.480
en-tr	12.0	.456	12.3	.460

Table 4: BLEU scores of character-level models trained by finetuning of the systems with 500 token vocabularies using deterministic BPE segmetnation and BPE dropout.

vocab.	architecture	# param.	BLEU
BPE 16k	Base	42.6M	26.86
char.	Base	35.2M	25.21
char.	Base + FF dim. 2650	42.6M	25.37

Table 5: Effect of model size on translation quality for Engslih-to-German translation.

level and subword-level model of 1–2 BLEU points. With the exception of Turkish, models trained by finetuning reach by a large margin better translation quality than character-level models trained from scratch.

In accordance with Provilkov et al. (2020), we found that BPE Dropout applied both on the source and target side leads to slightly worse translation quality, presumably because the stochastic segmentation leads to multimodal target distributions. The detailed results are presented in Appendix A. However, for most language pairs, we found a small positive effect of BPE dropout on the finetuned systems (see Table 4).

For English-to-Czech translation, we observe a large drop in BLEU score with the decreasing vocabulary size, but almost no drop in terms of METEOR score, whereas for other language pairs, all metrics are in agreement. The differences between the subword and character-level models are less pronounced in the low-resourced English-to-Turkish translation.

Whereas the number of parameters in transformer layers in all models is constant at 35 million, the number of parameters in the embeddings decreases 30× from over 15M to only slightly over 0.5M, with overall a 30% parameter count reduction. However, matching the number of parameters by increasing the model capacity narrows close the performance gap, as shown in Table 5.

In our first set of experiments, we finetuned the model using the character-level input directly. Experiments with parent models of various vocabulary sizes (column "Direct finetuning" in Table 3) suggest the larger the parent vocabulary, the worse the character-level translation quality. This result led us to hypothesize that gradually decreasing the vocabulary size in several steps might lead to better translation quality. In the follow-up experiment, we gradually reduced the vocabulary size by 500 and always finetuned until convergence. But we observed a small drop in translation quality in every step, and the overall translation quality was slightly worse than with direct finetuning (column "In steps" in Table 3).

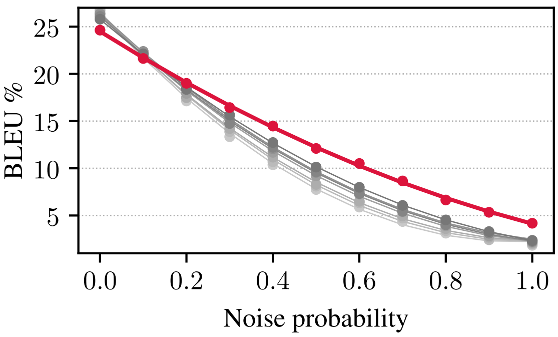

Figure 1: Degradation of the translation quality of the subword (gray, the darker the color, the smaller the vocabulary) and character-based systems (red) for English-German translation with increasing noise.

With our character-level models, we achieved higher robustness towards source-side noise (Figure 1). Models trained with a smaller vocabulary tend to be more robust towards source-side noise.

Character-level models tend to perform slightly better in the MorphEval benchmark. Detailed results are shown in Table 6. In German, this is due to better capturing of agreement in coordination and future tense. This result is unexpected because these phenomena involve long-distance dependencies. On the other hand, the character-level models perform worse on compounds, which are a local phenomenon. Ataman et al. (2019) observed similar results on compounds in their hybrid character-word-level method. We suspect this might be caused by poor memorization of some compounds in the character models.

In Czech, models with a smaller vocabulary better cover agreement in gender and number in pronouns, probably due to direct access to inflective endings. Unlike German, character-level models capture worse agreement in coordinations, presum-

	en-de		en-cs	
	BPE16k	char	BPE16k	char
Adj. strong	95.5	97.2	—	—
Comparative	93.4	91.5	78.0	78.2
Compounds	63.6	60.4	—	—
Conditional	92.7	92.3	45.8	47.6
Coordverb-number	96.2	98.1	83.0	78.8
Coordverb-person	96.4	98.1	83.2	78.6
Coordverb-tense	96.6	97.8	79.2	74.8
Coref. gender	94.8	92.8	74.0	75.8
Future	82.1	89.0	84.4	83.8
Negation	98.8	98.4	96.2	98.0
Noun Number	65.5	66.6	78.6	79.2
Past	89.9	90.1	88.8	87.4
Prepositions	—	—	91.7	94.1
Pronoun gender	—	—	92.6	92.2
Pronoun plural	98.4	98.8	90.4	92.8
Rel. pron. gender	71.3	71.3	74.8	80.1
Rel. pron. number	71.3	71.3	76.6	80.9
Superlative	98.9	99.8	92.0	92.0
Verb position	95.4	94.2	—	—

Table 6: MorphEval Results for English to German and English to Czech.

	32k	16k	8k	4k	2k	1k	500	0
T	1297	1378	1331	1151	1048	903	776	242
I	21.8	18.3	17.2	12.3	12.3	8.8	7.3	3.9
B	26.9	26.9	26.7	26.4	26.4	26.1	25.8	25.2

Table 7: Training (T) and inference (I) speed in sentences processed per second on a single GPU (GeForce GTX 1080 Ti) compared to BLEU scores (B) for English-German translation.

ably due to there being a longer distance in characters.

Training and inference times are shown in Table 7. Significantly longer sequences also manifest in slower training and inference. Table 7 shows that our character-level models are 5–6× slower than subword models with 32k units. Doubling the number of layers, which had a similar effect on translation quality as the proposed finetuning (Gupta et al., 2019), increases the inference time approximately 2–3× in our setup.

6 Conclusions

We presented a simple approach for training character-level models by finetuning subword models. Our approach does not require computationally expensive architecture changes and does not require dramatically increased model depth. Subword-based models can be finetuned to work on the character level without explicit segmentation with somewhat of a drop in translation quality. The models are robust to input noise and better capture some morphological phenomena. This is important for research groups that need to train and deploy character Transformer models without access to very large computational resources.

Acknowledgments

The work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 640550) and by German Research Foundation (DFG; grant FR 2829/4-1).

References

Duygu Ataman, Orhan Firat, Mattia A. Di Gangi, Marcello Federico, and Alexandra Birch. 2019. On the importance of word boundaries in character-level neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 187–193, Hong Kong. Association for Computational Linguistics.

Nikolay Banar, Walter Daelemans, and Mike Kestemont. 2020. Character-level transformer-based neural machine translation. *CoRR*, abs/2005.11239.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033, Brussels, Belgium. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, pages 41–48, Montreal, Quebec, Canada.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and

Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Ondřej Bojar, 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 *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London. Springer International Publishing.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Dokook Choe, Rami Al-Rfou, Mandy Guo, Heeyoung Lee, and Noah Constant. 2019. Bridging the gap for tokenizer-free language models. *CoRR*, abs/1908.10322.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Yingqiang Gao, Nikola I. Nikolov, Yuhuang Hu, and Richard H.R. Hahnloser. 2020. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1591–1604, Online. Association for Computational Linguistics.

Rohit Gupta, Laurent Besacier, Marc Dymetman, and Matthias Gallé. 2019. Character-based NMT with transformer. *CoRR*, abs/1911.04997.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, CA, USA.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Belgium, Brussels. 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.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 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.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

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

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

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

A Effect of BPE Dropout

We discussed the effect of BPE dropout in Section 3. Table 8 shows the comparison of the main quantitative results with and without BPE dropout.

B Notes on Reproducibility

The training times were measured on machines with GeForce GTX 1080 Ti GPUs and with Intel Xeon E5–2630v4 CPUs (2.20GHz). The parent models were trained on 4 GPUs simultaneously, the finetuning experiments were done on a single GPU.

We used model hyperparameters used by previous work and did not experiment with the hyperparameters of the architecture and training of the initial models. The only hyperparameter that we tuned was the learning rate of the finetuning. We set the value to 10^{-5} after several experiments with English-to-German translation with values between 10^{-7} and 10^{-3} based on the BLEU score on validation data.

We downloaded the training data from the official WMT web (http://www.statmt.org/wmt18/).The test and validation sets were downloaded using SacreBleu (https://github.com/mjpost/sacreBLEU). The BPE segmentation is done using FastBPE (https://github.com/glample/fastBPE). For BPE Dropout, we used YouTokenToMe (https://github.com/VKCOM/YouTokenToMe). A script that downloads and pre-processes the data is attached to the source code. It also includes generating the noisy synthetic data (using https://github.com/ybisk/charNMT-noise) and preparing data and tools required by MorphEval (https://github.com/franckbrl/morpheval).

The models were trained and evaluated with Marian v1.7.0 (https://github.com/marian-nmt/marian/releases/tag/1.7.0).

Validation BLEU scores are tabulated in Table 9.

		From random initialization								Direct finetuning from		
		32k	16k	8k	4k	2k	1k	500	0	500	1k	2k
en-de	BLEU	26.9	26.9	26.7	26.4	26.4	26.1	25.8	22.6	25.2	25.0	25.0
		25.7	26.3	25.9	26.2	25.6	25.7	25.3		24.9	24.3	24.7
	chrF	.569	.568	.568	.568	.564	.564	.561	.526	.559	.559	.559
		.563	.565	.565	.568	.561	.561	.559		.560	.553	.557
	METEOR	47.7	48.0	47.9	47.8	47.9	47.7	47.6	45.0	46.5	46.4	46.4
		47.0	47.8	47.4	48.0	47.5	47.8	47.7		46.5	46.1	46.3
de-en	BLEU	29.8	30.1	29.6	29.3	28.6	28.5	28.1	26.6	28.2	28.4	27.7
		29.8	29.3	28.8	29.5	28.7	28.8	28.6		28.5	27.9	28.5
	chrF	.570	.573	.568	.567	.562	.558	.558	.543	.562	.564	.559
		.573	.570	.569	.571	.565	.566	.566		.564	.561	.565
	METEOR	37.1	37.4	37.2	37.2	36.9	37.2	36.9	35.1	36.4	36.4	36.0
		37.0	37.1	36.9	37.2	37.0	37.0	37.0		36.5	36.3	36.5
en-cs	BLEU	21.1	20.8	20.9	20.6	20.1	20.0	19.5	18.2	19.2	19.3	19.4
		20.7	20.7	20.7	20.3	20.0	20.0	19.7		19.5	19.0	19.7
	chrF	.489	.488	.490	.487	.483	.482	.478	.465	.477	.476	.478
		.488	.489	.488	.486	.484	.482	.480		.480	.475	.482
	METEOR	26.0	25.8	26.0	25.8	25.7	25.7	25.4	24.6	25.2	25.2	25.2
		25.7	25.8	25.9	25.7	25.6	25.7	25.7		25.1	24.8	25.1
en-tr	BLEU	12.6	13.1	12.7	12.8	12.5	12.3	12.2	12.4	12.0	12.6	12.3
		10.7	11.6	12.2	12.7	12.6	12.5	12.5		12.3	12.2	12.6
	chrF	.455	.462	.459	.456	.457	.457	.455	.461	.456	.460	.459
		.436	.446	.457	.461	.464	.461	.459		.460	.461	.464

Table 8: Comparison of the trasnaltion quality without (gray numbers) and with BPE Dropout (with the same color coding as in Table 3).

	From random initialization								Direct finetuning from			In steps
	32k	16k	8k	4k	2k	1k	500	0	500	1k	2k	
en-de	29.07	29.76	28.6	28.7	28.11	27.61	27.66	26.09	28.04	27.89	27.87	27.75
de-en	35.05	35.26	34.34	35.34	34.37	34.84	33.83	27.96	32.61	33.47	33.68	32.44
en-cs	22.47	22.45	22.53	22.29	21.94	21.78	21.49	20.26	22.03	21.31	21.4	21.14
en-tr	13.40	14.18	14.25	14.11	14.05	13.72	13.94	14.55	12.02	12.25	12.28	11.56

Table 9: BLEU scores on the validation data: WMT13 test set for English-German in both directions, WMT17 test set for English-Czech and English-Turkish directions.