Efficient Inference for Multilingual Neural Machine Translation

Alexandre Bérard∗
NAVER LABS Europe

Dain Lee†
NAVER Corp.

Stéphane Clinchant∗
NAVER LABS Europe

Kweonwoo Jung‡
NAVER Corp.

Vassilina Nikoulina∗
NAVER LABS Europe

Abstract

Multilingual NMT has become an attractive solution for MT deployment in production. But to match bilingual quality, it comes at the cost of larger and slower models. In this work, we consider several ways to make multilingual NMT faster at inference without degrading its quality. We experiment with several “light decoder” architectures in two 20-language multi-parallel settings: small-scale on TED Talks and large-scale on ParaCrawl. Our experiments demonstrate that combining a shallow decoder with vocabulary filtering leads to more than $\times 2$ faster inference with no loss in translation quality. We validate our findings with BLEU and chrF (on 380 language pairs), robustness evaluation and human evaluation.

1 Introduction

Multilingual machine translation (Johnson et al., 2017; Bapna and Firat, 2019; Aharoni et al., 2019; Zhang et al., 2020; Fan et al., 2020; Lyu et al., 2020) has made a lot of progress in the last years. It is attractive because it allows handling multiple language directions within a single model, thus significantly reducing training and maintenance costs. However, to preserve good performance across all the language pairs, both the vocabulary size and model size have to be increased compared to bilingual NMT, which hurts inference speed. For example, the recently released M2M-100 (Fan et al., 2020) has 15B parameters and needs multiple GPUs for inference. The problem of inference speed has been well studied in bilingual settings (Kasai et al., 2021a,b; Chen et al., 2018; Hsu et al., 2020; Kim et al., 2019; Li et al., 2020; Shi and Knight, 2017). One line of work consists in using lighter decoder architectures (e.g., shallow decoder: Kasai et al., 2021a, RNN decoder: Chen et al., 2018; Kim et al., 2019; Lei, 2021). These works demonstrate that it is possible to significantly speed up inference with almost no loss in translation quality as measured by BLEU.

Figure 1 compares the inference time spent in each NMT component by bilingual (vocab size 16k) vs multilingual (vocab size 69k) Transformer Big 6-6 models on TED2020 valid EN-DE (average over 10 runs).

Figure 1: Decoding time spent in different components of bilingual (vocab size 16k) vs multilingual (vocab size 69k) Transformer Big 6-6 models on TED2020 valid EN-DE (average over 10 runs).

However, it is not so obvious that lighter decoder architectures would preserve translation quality in multilingual settings, where the decoder may need more capacity to deal with multiple languages. Therefore, the goal of this work is to benchmark different architectures in terms of inference speed/translation quality trade-off and to identify the best combination for multilingual NMT. The contributions of this paper are:

- A benchmark of two popular “light decoder” NMT architectures (deep encoder / shallow decoder, Kasai et al., 2021a; and RNN decoder, Chen et al., 2018) on two multilingual datasets

∗first.last@naverlabs.com  †dain.l@navercorp.com  ‡kweonwoo.jung@navercorp.com
(TED Talks and ParaCrawl) in both English-centric and multi-parallel settings. It demonstrates that the previous findings transfer to multilingual models.

- A combination of shallow decoder with per-language vocabulary filtering for further speed gains (achieving a global 2 to 3× speed-up over the baseline) with no loss in translation quality.
- Experiments with separate language-specific shallow decoders, which trade memory for higher BLEU performance, with comparable speed as the single-decoder approach.
- A validation of these findings through extensive analysis, including robustness evaluation and human evaluation.

2 Related work

Lightweight decoder. As shown in Figure 1, more than half of the inference time is devoted to the decoder and 30× more time is spent in the decoder than in the encoder (due to the autoregressive nature of the models). This explains why many efficient NMT works focus on lightweight alternatives to the Transformer decoder. Kim et al. (2019) perform an extensive study of various lightweight RNN architectures and obtain a 4× gain in inference speed. Kasai et al. (2021a) show that, in bilingual settings, Transformer models with a deep encoder and shallow decoder (e.g., 10-2) can achieve similar BLEU performance as baseline 6-6 Transformers, while being much faster at inference time (on a par with current non-autoregressive MT approaches). Behnke and Heafield (2020) show that it is possible to prune up to 75% of the attention heads in a Transformer, thus increasing inference speed by 50%. Similarly, Hsu et al. (2020) reduce the cost of cross-attention and self-attention by replacing it with an RNN or by pruning attention heads, obtaining up to 35% higher speed.

Although most of the above works report speed improvements with similar BLEU scores as Transformer baselines, it is uncertain that the same will hold in multilingual many-to-many settings, where the decoder may need more capacity to deal with multiple languages. In particular, Kong et al. (2021) observe that single shallow decoders degrade one-to-many MT quality and propose to train shallow language-specific decoders or decoders that are specific to a language family or group of languages.

Modular multilingual NMT. Lyu et al. (2020); Escolano et al. (2021) propose modular MT models with jointly trained language-specific encoders and decoders. Such models have higher per-language capacity, increasing their performance without hurting inference speed (contrary to the common approach of training bigger multilingual models). They are also more flexible for adding new languages. Zhang et al. (2021) study how language-specific and language-independent parameters naturally emerge in multilingual NMT. Their findings indicate that language-independent parameters can be distributed within the encoder and decoder and benefit final NMT performance.

3 Inference speed-up methods

3.1 Deep encoder, shallow decoder

First, we analyze how deep encoder / shallow decoder models Kasai et al. (2021a) behave in multilingual settings (many-to-many English-centric and multi-parallel).

Our initial experiments in bilingual settings showed that a 12-2 architecture gives the best BLEU/speed trade-off (also reported by Li et al., 2021). We thus focus on this architecture and compare it with 6-6 and 6-2 architectures.

We find that in some cases (with Transformer Base on TED Talks), post-norm 12-2 models fail to converge when trained from scratch. When this happens, we initialize the 12-2 model with a pre-trained 6-6 model’s parameters, by duplicating its encoder layers and taking its bottom 2 decoder layers. See Table 8 in Appendix for a comparison between this approach, training from scratch, and pre-norm Transformers.

3.2 RNN decoder

Chen et al. (2018) first introduced a hybrid model combining a Transformer encoder with an RNN decoder. Hybrid Transformer/RNN models are considered a good practical choice in production settings due to their ideal performance-speed trade-off (Caswell and Bowen Liang, 2020). However, Chen et al. (2018) do not experiment with hybrid models in a multilingual setting, nor do they try shallower RNN decoders. We experiment with 12-layer Transformer encoders combined with either 2-layer or 3-layer LSTM decoders (noted Hybrid 12-2 / 12-3). Because LSTMs are slower to train,

1 We prefer post-norm Transformers, as Liu et al. (2020) show that when they do converge, they often lead to better performance than pre-norm.

2 Hybrid 12-3 has a similar amount of parameters as Transformer 12-2.
we first train 12-2 Transformers which we fine-tune into Hybrid models (by initializing the LSTM decoder at random). Precise architecture details are given in Appendix A.2.

3.3 Target vocabulary filtering

As illustrated by Figure 1, decoding speed can also be impacted by the size of the target vocabulary, because the softmax layer’s complexity is linear with respect to the vocabulary size. Some solutions have been proposed to compress vocabulary in bilingual settings: vocabulary hashing or vocabulary shortlists (Shu and Nakayama, 2017; Shi and Knight, 2017; Senellart et al., 2018; Kim et al., 2019). Ding et al. (2019) also showed that the BPE size can be reduced drastically without hurting BLEU. However, reducing the BPE size too aggressively will result in longer sequences and hurt decoding speed. Lyu et al. (2020) train a separate smaller BPE model per language. However, we think that this may hurt transfer learning between languages that share words (one of the reasons why multilingual NMT uses shared vocabularies in the first place). Therefore, we propose a solution that combines the best of both worlds: have a large shared BPE vocabulary at train time, which we decompose into smaller language-specific vocabularies at test time, based on per-language token frequencies. More precisely, we train a shared BPE model of size 64k, then for each language:

1. We tokenize its training data and count the wordpiece and character occurrences.
2. We build a vocabulary containing only tokens whose frequency is above threshold $K$ and only the $N$ most frequent wordpieces.
3. At test time, we can filter the model’s target vocabulary and embeddings to only contain these tokens, resulting in a model with a single shared source embedding matrix and several smaller per-language target embedding matrices. We call this approach “test-time BPE filtering” (with parameters $N_{test}$ and $K_{test}$). Appendix Tables 15 & 16 give the incurring parameter cost.
4. We also try combining this approach with “train-time BPE filtering” (with parameters $N_{train}$ and $K_{train}$). For target-side training sentences in this language, we force the shared BPE model to only generate wordpieces that belong to this language’s filtered vocabulary.\footnote{By using \texttt{subword-nmt}'s \texttt{--vocabulary-}}

3.4 Shared encoder, language-specific decoders

Lyu et al. (2020) show that one can significantly increase the capacity (and thus performance) of a multilingual model without hurting decoding speed by training language-specific encoders and decoders (i.e., trading away memory for speed). We take the approach of a deeper shared encoder and multiple language-specific shallow decoders (similar to Kong et al., 2021). This approach keeps the memory usage to a reasonable value,\footnote{As shown in Appendix (Table 16) a 20-language Big 12-2 multi-decoder model has 823M parameters in total, while a Big 6-6 or Big 12-2 multi-encoder + multi-decoder model would have $\approx 20 \times 180M = 3.6B$ parameters.} and can maximize transfer learning on the encoder side.

Contrary to Lyu et al. (2020) and Kong et al. (2021), to save computation time, we first train shared multilingual MT models, which we use as initialization to our multi-decoder models (i.e., the same 2-layer decoder is copied). We use language-specific target embeddings that are initialized with the shared embeddings obtained with the “train-time BPE filtering” technique described in the previous section. We refer to the models with shallow language-specific decoders as “multi-decoder models.”

3.5 Summary of notations

- **Base/Big 6-6/12-2** correspond respectively to Transformer Base/Big with 6 encoder layers and 6 decoder layers (resp. 12 and 2 layers).
- By default, models are trained on **English-centric** data (i.e., data in all languages paired with English, in both directions).
- **Multi-parallel** models are fine-tuned on data in all language directions (not just paired with English).
- Some models use test-time BPE filtering ($N_{test}$ or $K_{test}$) while others use both train-time and test-time filtering ($N_{train}$ or $K_{train}$).
- **Hybrid** models have a Transformer encoder and LSTM decoder and are fine-tuned from English-centric Transformers with multi-parallel data.
- **Multi-decoder** models have language-specific shallow decoders and are fine-tuned from English-centric models with multi-parallel data.

threshold option.
4 TED Talks Experiments

4.1 Data and hyper-parameters

We experiment with the TED Talks corpus (Qi et al., 2018) with the same set of 20 languages as Philip et al. (2020). This corpus is multi-parallel, i.e., it has training data for all 380 (20 × 19) language pairs (see Table 7 in Appendix for detailed statistics). It also includes official valid and test splits for all these language pairs.

We train the English-centric models for 120 epochs (≈1.8M updates). The Base 12-2 English-centric model is initialized from Base 6-6 at epoch 60 and trained for another 60 epochs, using the procedure described in Section 3.1. These models are then fine-tuned with multi-parallel data for another 10 epochs (≈1.4M updates) into single-decoder Transformers or Hybrid models or multi-decoder Transformers. We create a shared BPE model with 64k merge operations (vocabulary size 70k) and with inline casing (Berard et al., 2019). More hyper-parameters are given in Appendix A.2.

4.2 Evaluation settings

The TED Talks models are evaluated on the provided multi-parallel validation and test sets. Since those are already word-tokenized, we run SacreBLEU with the --tok none option.5

We report average test BLEU scores into English (→EN, 19 directions), from English (←EN, 19 directions) and outside of English (/EN, 342 directions). We also compute the decoding speed in Words Per Second (WPS)6 when translating the concatenated →EN valid sets on a V100 with batch size 64 and beam size 5 (averages over 3 runs). Additional speed benchmarks with other decoding settings and time spent in each component are given in Appendix Table 17. We also report chrF scores and results on more models in Appendix Table 20.

4.3 Position of the language code

The prevalent approach in multilingual NMT for choosing the target language is to prefix the source sequence with a language code (Johnson et al., 2017). However, it is also possible, like Tang et al. (2020), to put this code on the target side. Table 6 in Appendix analyzes the impact of the position of this language code on BLEU performance. Like observed by Wu et al. (2021), decoder-side language codes result in very low zero-shot performance in the English-centric setting. They also degrade the performance of the Base 12-2 models in all translation directions. For this reason, all our experiments use source-side language codes.

4.4 BLEU results

Table 1 evaluates the techniques we proposed in Section 3 on TED Talks. First, we see that the Base 12-2 models (3, 6) perform as well or better than the Base 6-6 models (2, 5) in all language directions, with a 1.7× speed boost. Multi-parallel fine-tuning (5, 6) significantly increases translation quality between non-English languages and incurs no drop in performance in the ↔EN directions. Test-time filtering of the vocabulary with \( K_{test} = 10 \) (see Section 3.3) does not degrade BLEU but increases decoding speed by 30% (7). More aggressive filtering with \( N_{test} = 4k \) results in a drop in BLEU (8). The latter leads to slightly longer outputs (in terms of BPE units), which explains why it is not faster. When training with \( N_{train} = 4k \), we can get the same speed boost (9, 10), without any drop in BLEU compared to models without BPE filtering (5, 6).

Then, we observe that Hybrid models (11, 12) are slightly worse than Transformers in terms of BLEU, but are also much faster at decoding. Hybrid 12-2 (11) is 3× faster than Base 6-6 (-0.2 BLEU on average) and 1.7× faster than Base 12-2 (-0.3 BLEU). Hybrid 12-3 (12) is slower than Hybrid 12-2 and not clearly better in terms of BLEU (+0.1 BLEU).

Finally, we see that fine-tuning the English-centric 12-2 model into 20 language-specific shallow decoders with the multi-parallel data (14) results in the highest BLEU scores overall, with the same speed benefits as with a single shallow decoder (10). A Base-6 model can also be finetuned into multiple 2-layer language-specific decoders (13) and get the same performance as the single Base 6-6 or Base 12-2 models (9, 10). This is convenient if one wants to quickly improve the decoding speed of existing 6-6 models.

5Note that an “epoch” when using multi-parallel data corresponds to approximately 9 English-centric epochs in terms of updates.

6SacreBLEU signature:

\[
\text{BLEU+c,mixed+#.1+es.exp+tok.nontextv.1.5.1}
\]

7Not BPE tokens per second, as we do not want the speed measurement to depend on the BPE tokenization used.
| Model       | →EN | ←EN | / EN | WPS |
|-------------|-----|-----|------|-----|
| English-centric |
| Bilingual   | 32.4| 24.4| 15.0 | –   |
| Best multi  | 32.3| 24.1| 15.8 | –   |

Table 1: Test BLEU scores and decoding speed of TED Talks models of various depths. SOTA’s “best multi” is a Transformer Small 6-6 multi-parallel model with adapter layers. WPS: speed in words per second for →EN translation. Table 20 in Appendix reports scores by more models and with additional metrics.

We train the English-centric models for 1M steps and fine-tune them with multi-parallel data for 200k more steps. Hybrid and Multi-decoder models are also fine-tuned for 200k steps from the English-centric models with multi-parallel data. Big 6-6 bilingual baselines are trained with the same hyper-parameters for 120k steps, with joint BPE vocabularies of size 16k. More hyper-parameters are given in Appendix A.2.

The Big 6-6 and Big 12-2 English-centric models took each around 17 days to train on 4 A100s. The multi-parallel fine-tuning stages (single/multi-decoder and hybrid) took ≈4.5 days on 2 A100s each.

5.2 Evaluation settings

The ParaCrawl models are evaluated on our own valid and test splits from TED2020 (Reimers and Gurevych, 2020). We shuffle the parallel corpus for each translation direction and take 3000 line pairs for the validation set and 3000 for the test set. To compare against the state of the art, we also provide scores on standard test sets from WMT for some language pairs. In both cases, we use SacreBLEU with its default options.

Like in Section 4, we compute average →EN, ←EN and / EN test BLEU scores and WPS on →EN TED2020 valid. Table 21 in Appendix reports TED2020 test chrF (Popović, 2015), as well as spBLEU scores on FLORES devtest (Goyal et al., 2021).

5.3 BLEU results

Table 2 shows that, like in the TED Talks experiments, the 12-2 architecture (17, 20) gets as good or better BLEU scores than the standard 6-6 Transformer (16, 19) and is 70% faster. It outperforms the Big 6-6 baseline in all 38 English-centric directions, both according to BLEU and chrF. Test-time BPE filtering with \( N_{train} = 16k \) (see Section 3.3) does not degrade BLEU (21) and improves decoding speed by 30%. However, \( N_{test} = 8k \) leads to a large drop in BLEU (22) without any additional speed benefit. Indeed, in this setting 1.5% of the tokens that would have been generated by the non-

\[ \text{SacreBLEU signature:} \quad \text{BLEU} + c.\text{mixed} + s.\text{exp} + \text{tok.13a} + v.1.5.1 \]

\( ^{10} \) [fr, de, es, it, pt, nl, nb, cs, pl, sv, da, el, fi, hr, hu, bg, ro, sk, lt]

\( ^{11} \) TED2020 is a different crawl from TED than that of the “TED Talks” corpus. It is more recent, it has more data and languages and it is not word-tokenized.

\( ^{12} \) SacreBLEU signature: BLEU+c.mixed+s.exp+tok.13a+v.1.5.1

\( ^{13} \) Note that when \( N_{train} \) or \( N_{test} \) is set, we additionally apply a frequency threshold of \( K = 100 \) on BPE tokens and characters.
Table 2: TED2020 test BLEU scores and decoding speed of ParaCrawl models of various depths. WPS: speed in words per second for $\rightarrow$EN translation. Table 21 in Appendix reports scores by more models and with additional metrics.

| Model                  | $\rightarrow$EN | $\leftarrow$EN | /EN | WPS  |
|------------------------|------------------|----------------|-----|------|
| English-centric        |                  |                |     |      |
| 15 Base 6-6            | 31.4             | 26.0           | 13.2| 656  |
| 16 Big 6-6             | 35.0             | 29.0           | 14.4| 623  |
| 17 Big 12-2            | **35.5**         | **29.6**       | 16.7| 1033 |
| 18 + EN pivot          | –                | –              | 21.5| –    |
| **+ Multi-parallel**   |                  |                |     |      |
| 19 Big 6-6             | 34.2             | 28.3           | 20.7| 595  |
| 20 Big 12-2            | 34.9             | 29.2           | 21.3| 1030 |
| 21 + $N_{test} = 16k$ | 34.8             | 29.0           | 21.2| 1328 |
| 22 + $N_{test} = 8k$  | 32.9             | 27.8           | 20.4| 1283 |
| **+ BPE filtering ($N_{train} = 8k$)** |                  |                |     |      |
| 23 Big 6-6             | 34.0             | 28.4           | 20.7| 679  |
| 24 Big 12-2            | 34.8             | 29.1           | 21.2| 1261 |
| 25 Hybrid 12-2         | 33.9             | 28.3           | 20.7| **1796** |
| 26 Hybrid 12-3         | 34.0             | 28.3           | 20.8| 1659 |
| **+ Multi-decoder**    |                  |                |     |      |
| 27 6-6 $\rightarrow$ 6-2 | 34.0             | 28.8           | 21.0| 1333 |
| 28 12-2 $\rightarrow$ 12-2 | 35.3             | 29.4           | **21.8** | 1270 |

Table 3: Comparison of our multi-parallel ($N_{train} = 8k$) ParaCrawl models with bilingual baselines and with the state of the art on newstest2019. Shi et al. (2020) is the top-ranking submission at WMT20 in those languages. We report their baseline scores (i.e., without back-translation, ensembling, etc.) Pivot translation with bilingual models (resp. with Big 12-2 English-centric) gives 34.2 BLEU (resp. 32.0 BLEU) on DE-FR.

Table 3 compares our multi-parallel models with bilingual Big 6-6 baselines. The languages are sorted from highest-resource to lowest-resource.

Figure 2: BLEU deltas on $\leftarrow$EN TED2020 test (in a subset of 8 languages) by our multilingual ParaCrawl models (17, 28, 16, 24, 25, 15) compared to bilingual Big 6-6 baselines. The languages are sorted from highest-resource to lowest-resource.

Figure 2 shows the $\leftarrow$EN BLEU difference between our multilingual models and the ParaCrawl bilingual baselines on a subset of 8 languages. We see the same trend as in the literature: multilingual training hurts performance on high-resource languages and helps on lower-resource languages. We also see that Transformer Big 12-2 consistently outperforms Big 6-6 and that multi-parallel training consistently hurts $\leftarrow$EN performance. Figure 6 in Appendix shows similar scores for the $\rightarrow$EN and $\leftarrow$EN performance.

†Average number on the $\leftarrow$EN TED2020 valid outputs of the Big 12-2 multi-parallel model.

†Average number on the training data.
When reducing the depth of the decoder, one could expect that it would have trouble generating long sequences. Figure 3 reports BLEU scores for different length buckets. We observe no abnormal patterns in any of the proposed architectures. We first note that Big 12-2 (24) performs consistently better than Big 6-6 (23) across all sentence lengths. The performance of the Hybrid 12-2 model (25) is also consistent (slightly lower than Transformers). Figure 7 in Appendix shows scores by length in the →EN direction and with greedy decoding.

### 5.6 Robustness analysis

Even if different decoder architectures reach similar BLEU performance, some architectures might be more brittle to noise than others. To test each model’s robustness, we introduce synthetic noise by either adding an unknown character (unk) randomly at the beginning, middle, or end of the sentence; or by applying 3 random char-level operations (del, ins, swap, or sub) (char). Table 4 reports the BLEU consistency (“Cy BLEU”) as introduced by Niu et al. (2020) on →EN translation. As previously, deep-encoder / shallow decoder models (Big 12-2, Big 12-2 Multi-decoder) outperform the other architectures. BPE filtering slightly hurts robustness, despite showing close BLEU scores on the clean test sets. Additional results are given in the Appendix (Table 10).

### 5.7 Human evaluation

We conduct a human evaluation to compare the English-centric Big 6-6 and Big 12-2 models. It is done by certified professionals who are proficient in both the source and target language. We use bilingual direct assessment (DA), where raters have to evaluate the adequacy and fluency of each translation on a 0-5 scale given the source sentence. We select a random subset of 200 sentences from newstest2019 for DE-EN / EN-DE and from newstest2014 for FR-EN / EN-FR. For each translation direction, 3 raters are shown all the source sentences and their translations by both systems in random order. Table 5 reports relative results averaged across the 3 raters. Big 12-2 outperforms Big 6-6 in 3 out of 4 language directions. Contrary to Kong et al. (2021) and according to both human evaluation and automatic metrics, our single-shallow-decoder model performs at least as well as the baseline model.\(^\text{18}\)

---

\(^{16}\)Bleu consistency measures the similarity between the translations by the same model of the clean sentence and its noised version.

\(^{17}\)We ensure that the source sentences are original text in the corresponding language to avoid biased evaluation results due to Translationese.

\(^{18}\)Note that our human evaluation results are only on high-resource languages, but Kong et al. (2021) observed the largest BLEU drop on high-resource languages.
| Model     | 12-2 > 6-6 | 12-2 = 6-6 | 12-2 < 6-6 |
|-----------|------------|------------|------------|
| EN→FR     | 26%        | 51%        | 23%        |
| FR→EN     | 25%        | 51%        | 24%        |
| EN→DE     | 31%        | 44%        | 25%        |
| DE→EN     | 24%        | 50%        | 26%        |

Table 5: Human evaluation on English-centric ParaCrawl models (16 and 17).

6 Conclusion

On one hand, multilingual NMT saves training and deployment costs. On the other hand, larger architectures (required to keep performance on a par with bilingual MT) and large shared vocabularies penalize inference speed and user latency. In this work, we study various approaches to improve the speed of multilingual models without degrading translation quality. We find that Transformers with a deep encoder and a shallow decoder can outperform a baseline Transformer at a much faster decoding speed. This can be combined with per-language vocabulary filtering to reach a global 2× speed-up with no loss in BLEU. A careful analysis of the results on different aspects such as sequence length, robustness to noise, and human evaluation validates this finding. Additionally, language-specific shallow decoders can be trained to get even better performance at the same speed. And finally, hybrid models with a shallow RNN decoder offer an excellent BLEU-speed trade-off (3× faster than baseline with a minor drop in BLEU). We also provide supplementary material (code, checkpoints, test splits, translation outputs) to facilitate reproducibility.19

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, GemaRamirez-Sánchez, Elsa Sarrías, Marek Střelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4555–4567, Online. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation 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 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2664–2674, Online. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver Labs Europe’s systems for the WMT19 machine translation robustness task. In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), pages 526–532, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell and Google Research Bowen Liang. Software Engineers. 2020. Recent advances in Google Translate.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Lilian Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Ziheng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Shuo yang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In Proceedings of Machine Translation Summit XVII Volume 1: Research Track, pages 204–213, Dublin, Ireland. European Association for Machine Translation.
Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4465–4470, Online. 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.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4512–4525, Online. Association for Computational Linguistics.

Jean Senellart, Dakun Zhang, Bo Wang, Guillaume Klein, Jean-Pierre Ramat Chandirin, Josep Crego, and Alexander Rush. 2018. OpenNMT system description for WNMT 2018: 800 words/sec on a single-core CPU. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 122–128, Melbourne, Australia. Association for Computational Linguistics.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao. 2020. OPPO’s machine translation systems for WMT20. In Proceedings of the Fifth Conference on Machine Translation, pages 282–292, Online. Association for Computational Linguistics.

Xing Shi and Kevin Knight. 2017. Speeding up neural machine translation decoding by shrinking runtime vocabulary. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 574–579, Vancouver, Canada. Association for Computational Linguistics.

Raphael Shu and Hideki Nakayama. 2017. Compressing word embeddings via deep compositional code learning. arXiv preprint arXiv:1711.01068.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. arXiv preprint arXiv:2008.00401.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3001–3007, Online. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In International Conference on Learning Representations.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1628–1639, Online. Association for Computational Linguistics.
A Appendix

A.1 Position of the language code

Table 6 analyzes the impact of the language code position on BLEU performance. With the Base 6-6 architecture, decoder-side codes perform approximately as well as encoder-side codes (except for zero-shot translation). However, with the Base 12-2 architecture, decoder-side codes result in a noticeable drop in performance in most directions. Indeed, when the lang code is on the source side, the deep encoder knows the target language and can start “translating.” When it is on the target side, the encoder has no way of knowing which language to start translating into. So it outputs a universal representation that is harder to transform into a target-language sentence by the limited-capacity shallow decoder. Note that $\rightarrow$EN performance in the English-centric setting is not affected. We believe this is because the encoder can easily guess that the target language is English by detecting the language of the input. We believe this is also the reason for the low zero-shot performance: the encoder starts translating all non-English inputs into English, and the decoder receives a representation that it cannot translate into other languages than English.

| Lang code position | Model     | $\rightarrow$EN | $\leftarrow$EN | / EN |
|--------------------|-----------|-----------------|---------------|-----|
| 2                  | Encoder   | Base 6-6        | 31.8          | 24.2| 13.5 |
| 29                 | Decoder   | Base 6-6        | 32.0          | 23.9| 5.43 |
| 3                  | Encoder   | Base 12-2       | 33.6          | 24.3| 14.1 |
| 30                 | Decoder   | Base 12-2       | 33.5          | 22.8| 0.74 |
| Multi-parallel + BPE filtering (N$_{train}$ = 4k) | | | |
| 9                  | Encoder   | Base 6-6        | 32.9          | 24.2| 16.3 |
| 31                 | Decoder   | Base 6-6        | 32.7          | 23.9| 16.2 |
| 10                 | Encoder   | Base 12-2       | 33.3          | 24.3| 16.3 |
| 32                 | Decoder   | Base 12-2       | 32.5          | 23.0| 15.7 |

Table 6: Test BLEU scores of TED Talks models with encoder-side or decoder-side language codes. Like Tang et al. (2020), we implement decoder-side lang codes by replacing BOS (i.e., the first embedding input to the decoder) with the lang code.

A.2 Framework and hyper-parameters

We do our experiments in the fairseq v0.10.2 framework (Ott et al., 2019), which we modify to implement on-the-fly pre-processing and sampling from multilingual corpora.

We randomly sample language pairs with $p_k = \frac{D_k}{\sum D_i}$ where $D_k$ is the number of sentence pairs for language pair $k$ and $T$ is the temperature (Arivazhagan et al., 2019). Tables 7 and 9 give the resulting sampling probabilities by target language. We build heterogeneous batches using this sampling strategy (i.e., containing any mixture of languages), by sampling 100k sentence pairs at a time and sorting them by length into batches. Language-specific decoders are trained with homogeneous batches with respect to the target language (we increase the “buffer size” to 1M and group sentence pairs by target language before batching).

Tables 11 and 12 give the fairseq hyperparameters of our TED Talks and ParaCrawl Transformer models. Tables 13 and 14 give the training details of the fine-tuned models.

Our Hybrid models use a variant of the hybrid RNMT+ architecture proposed by Chen et al. (2018). Contrary to them, we use single-head additive attention (Bahdanau et al., 2015); sum the attention and LSTM output before the vocabulary projection; and apply layer normalization on the input of the LSTMs (rather than on the gates). We apply the same amounts of dropout as in the Transformer but on both the LSTM outputs (except for the first LSTM) and the target embeddings.
| Language          | Code | English-centric lines | Multi-parallel lines |
|------------------|------|-----------------------|----------------------|
| English          | en   | 3,556,962             | 3,556,962            |
| Arabic           | ar   | 214,111               | 3,430,385            |
| Hebrew           | he   | 211,819               | 3,399,679            |
| Russian          | ru   | 208,458               | 3,379,440            |
| Korean           | ko   | 205,640               | 3,350,599            |
| Italian          | it   | 204,503               | 3,350,483            |
| Japanese         | ja   | 204,090               | 3,312,997            |
| Mandarin Chinese | zh_cn| 199,855               | 3,297,628            |
| Spanish          | es   | 196,026               | 3,234,798            |
| French           | fr   | 192,304               | 3,192,551            |
| Brazilian Portuguese | pt_br| 184,755               | 3,110,048            |
| Dutch            | nl   | 183,767               | 3,053,593            |
| Turkish          | tr   | 182,470               | 3,017,706            |
| Romanian         | ro   | 180,484               | 3,055,943            |
| Polish           | pl   | 176,169               | 3,002,206            |
| Bulgarian        | bg   | 174,444               | 2,946,693            |
| Vietnamese       | vi   | 171,995               | 2,807,695            |
| German           | de   | 167,888               | 2,900,115            |
| Persian          | fa   | 150,965               | 2,405,646            |
| Hungarian        | hu   | 147,219               | 2,465,081            |
| **Total**        | all  | 7,113,924             | 62,270,248           |

Table 7: Size of the Top 20 TED Talks corpus. English has 253,292 unique lines. The average English sentence length is 21.1 words and 26.7 wordpieces.

| Model              | →EN | ←EN | / EN |
|--------------------|-----|-----|------|
| **English-centric**|     |     |      |
| 2 Base 6-6 post-norm | 32.1 | **24.5** | 13.6 |
| 3 Base 12-2 post-norm pre-trained | **33.7** | **24.5** | 14.1 |
| 33 Base 12-2 pre-norm | 33.5 | 24.3 | 12.6 |
| 34 Base 12-2 enc pre-norm | 33.3 | 24.1 | 13.1 |
| 35 Base 18-1 enc pre-norm* | 31.6 | 22.8 | 12.1 |

Table 8: Valid BLEU scores of English-centric TED Talks models with deep encoders, depending on the training strategy used. *: this model was stopped before the end (after 60 epochs).

![Figure 4](image_url) Figure 4: Training progress of English-centric models trained on TED Talks. The names in the legend are sorted from highest to lowest BLEU score. The pre-trained 12-2 model was initialized with the 6-6 model’s checkpoint at epoch 60 and fine-tuned for 60 more epochs. The 18-1 model was stopped in the middle of training.
Figure 5: Training progress of English-centric and multi-parallel models trained on ParaCrawl (without BPE filtering). The names in the legend are sorted from highest to lowest BLEU score. The pre-trained 12-2 model was initialized with the 6-6 model’s checkpoint at step 1M and fine-tuned for 1M more steps.

Figure 6: BLEU deltas on TED2020 test (in a subset of 8 languages) by our multilingual ParaCrawl models (17, 28, 16, 24, 25, 15) compared to pivot translation through English with bilingual Big 6-6 baselines. The “non-English” score for language X is the average of test scores from all the other languages (except English) into X. The languages are sorted from highest-resource to lowest-resource (in terms of English-centric data amounts).
| Language | Code | Family | English-centric | Multi-parallel |
|----------|------|--------|----------------|----------------|
|          |      |        | Lines          | Trg. prob (T=5) | Lines          | Trg. prob (T=2) |
| English  | en   | Germanic | 450,298,290   | 0.500          | 450,298,290   | 0.109          |
| French   | fr   | Romance  | 95,432,158    | 0.038          | 215,631,123   | 0.070          |
| German   | de   | Romance  | 76,490,492    | 0.036          | 192,673,679   | 0.068          |
| Spanish  | es   | Romance  | 72,973,508    | 0.036          | 191,713,598   | 0.068          |
| Italian  | it   | Romance  | 38,054,969    | 0.031          | 136,107,553   | 0.061          |
| Portuguese | pt | Romance | 29,181,190    | 0.030          | 117,682,221   | 0.058          |
| Dutch    | nl   | Germanic | 27,361,570    | 0.029          | 104,353,586   | 0.055          |
| Norwegian | nb | Germanic | 15,384,700    | 0.026          | 65,367,888    | 0.045          |
| Czech    | cs   | Slavic   | 12,922,615    | 0.025          | 65,547,562    | 0.046          |
| Polish   | pl   | Slavic   | 12,877,872    | 0.025          | 69,265,211    | 0.047          |
| Swedish  | sv   | Germanic | 10,969,372    | 0.025          | 60,160,949    | 0.044          |
| Danish   | da   | Germanic | 9,792,687     | 0.024          | 61,276,295    | 0.044          |
| Greek*   | el   | Hellenic | 8,915,258     | 0.024          | 48,294,800    | 0.039          |
| Finnish  | fi   | Uralic   | 6,833,568     | 0.022          | 47,624,701    | 0.040          |
| Croatian | hr   | Slavic   | 6,338,125     | 0.022          | 30,467,579    | 0.031          |
| Hungarian| hu   | Uralic   | 6,294,289     | 0.022          | 42,527,237    | 0.037          |
| Bulgarian* | bg | Slavic   | 6,098,653     | 0.022          | 36,835,445    | 0.035          |
| Romanian | ro   | Romance  | 5,786,263     | 0.022          | 40,521,359    | 0.037          |
| Slovak   | sk   | Slavic   | 4,557,803     | 0.021          | 36,387,740    | 0.035          |
| Lithuanian | lt | Baltic   | 4,033,198     | 0.020          | 30,205,598    | 0.032          |
| Total    | all  | –        | 900,596,580   | 1.0            | 2,042,942,414 | 1.0            |
| Russian* | ru   | Slavic   | 5,120,207     | –              | –              | –              |
| Latvian  | lv   | Baltic   | 3,607,272     | –              | –              | –              |

Table 9: Size of the Top 20 ParaCrawl corpus and target language sampling probabilities in its English-centric setting and its multi-parallel setting. T = sampling temperature. English has 271,851,754 unique lines. ⋆: all languages use the latin script, except for Greek (Greek alphabet) and Bulgarian/Russian (Cyrillic). The average English sentence length is 19.0 words and 30.4 wordpieces.

| Model   | BLEU unk | BLEU Consistency unk | BLEU char | BLEU Consistency char |
|---------|----------|----------------------|-----------|-----------------------|
| 19 Big 6-6          | 24.2     | 73.3                 | 19.5      | 54.2                  |
| 20 Big 12-2         | **26.0** | **76.4**             | **21.1**  | **56.1**              |
| 21 Big 12-2 (N_{test} = 16k) | 26.0     | 75.9                 | 21.0      | 56.0                  |
| 23 Big 6-6 (N_{train} = 8k) | 24.4     | 74.4                 | 19.6      | 54.5                  |
| 24 Big 12-2 (N_{train} = 8k) | 25.6     | 73.7                 | 20.9      | 55.5                  |
| 25 Hybrid 12-2 (N_{train} = 8k) | 24.8     | 75.0                 | 20.4      | 55.3                  |
| 26 Hybrid 12-3 (N_{train} = 8k) | 25.1     | 76.5                 | 20.2      | 55.0                  |

Table 10: Robustness evaluation on ←EN test sets of the multi-parallel ParaCrawl models. “Unk” adds one unknown symbol at a random position in each test sentence and “char” does 3 random character-level operations per sentence. “BLEU consistency” by Niu et al. (2020) is a BLEU score between the translations of the clean and noisy versions of the same test set by a given model.
| Parameter name | Parameter value |
|----------------|----------------|
| share_all_embeddings | True |
| arch | transformer |
| lr_scheduler | inverse_sqrt |
| optimizer | adam |
| adam_betas | 0.9, 0.999 |
| fp16 | True |
| clip_norm | 0.0 |
| lr | 0.0005* |
| warmup_updates | 4000 |
| warmup_init_lr | 1e-07 |
| criterion | label_smoothed_cross_entropy |
| label_smoothing | 0.1 |
| dropout | 0.3* |
| max_tokens | 4000 |
| max_epoch | 120* |
| save_interval | 1 |
| validate_interval | 1 |
| keep_last_epochs | 1 |
| update_freq | 4† |
| lang_temperature | 5 |
| decoder_dropout | 0.3* |

Table 11: fairseq hyper-parameters of the Base 6-6 TED Talks models. †: we normalize this value by the number of GPUs to have a constant batch size. For instance, models trained on 4 GPUs use update_freq=1. *: as shown in Table 13, the fine-tuned models use different values for these parameters. The Small 6-6 models use the transformer_iwslt_de_en architecture.

| Parameter name | Parameter value |
|----------------|----------------|
| max_source_positions | 256 |
| max_target_positions | 256 |
| share_all_embeddings | True |
| arch | transformer_vaswani_wmt_en_de_big |
| lr_scheduler | inverse_sqrt |
| optimizer | adam |
| adam_betas | 0.9, 0.98 |
| fp16 | True |
| clip_norm | 1.0 |
| lr | 0.0005* |
| warmup_updates | 4000 |
| warmup_init_lr | 1e-07 |
| criterion | label_smoothed_cross_entropy |
| label_smoothing | 0.1 |
| dropout | 0.1 |
| max_tokens | 8000 |
| max_update | 1000000* |
| save_interval_updates | 20000 |
| validate_interval_updates | 20000 |
| update_freq | 32† |
| lang_temperature | 5† |

Table 12: fairseq hyper-parameters of the Big 6-6 ParaCrawl models. †: we normalize this value by the number of GPUs to have a constant batch size. For instance, models trained on 4 GPUs use update_freq=8. †: we use a temperature of 2 in the multi-parallel (or multi-decoder) finetuning stage. *: as shown in Table 14, the fine-tuned models use different values for these parameters.
| Models                          | Initialized with | LR reset | Dropout | Data    | Epochs |
|--------------------------------|------------------|----------|---------|---------|--------|
| Base 6-6 (2)                   | –                | –        | 0.3     | TED-EN  | 120    |
| Base 6-2 (36)                  | –                | –        | 0.3     | TED-EN  | 120    |
| Base 12-2 (3)                  | (2) @60          | yes (0.0005) | 0.3   | TED-EN  | 60     |
| Base 6-6 Multi-parallel (5, 9) | (2) @120         | no       | 0.1     | TED-ALL | 10     |
| Base 12-2 Multi-parallel (6, 10)| (3) @60        | no       | 0.1     | TED-ALL | 10     |
| Hybrid Multi-parallel (11, 12) | (3) @60          | yes (0.0003) | 0.1    | TED-ALL | 10     |
| Base 12-2 Multi-decoder (14)   | (3) @60          | yes (0.0001) | 0.1 / 0.3 | TED-ALL | 10     |

Table 13: Details about multi-stage training of TED Talks models. *: different dropout values in the encoder and decoders.

| Models                          | Initialized with | LR reset | Data    | Updates |
|--------------------------------|------------------|----------|---------|---------|
| Big 6-6 (16)                    | –                | –        | Para-EN | 1M      |
| Big 6-2 (37)                    | –                | –        | Para-EN | 1M      |
| Big 12-2 (17)                   | –                | –        | Para-EN | 1M      |
| Big 6-6 Multi-parallel (19, 23) | (16) @1M         | no       | Para-ALL| 200k    |
| Big 12-2 Multi-parallel (20, 24)| (17) @1M         | no       | Para-ALL| 200k    |
| Hybrid Multi-parallel (25, 26)  | (17) @1M         | yes (0.0005) | Para-ALL| 200k    |
| Big 12-2 Multi-decoder (28)     | (17) @1M         | yes (0.0005) | Para-ALL| 200k    |

Table 14: Details about multi-stage training of ParaCrawl models.

| Models                          | Params without embeddings (M) | Embeddings (M) |
|--------------------------------|-------------------------------|----------------|
| Base 6-6 (2, 5)                 | 44.1 (18.9 + 25.2)            | 36.0           |
| Base 6-2 (36)                   | 27.3 (18.9 + 8.4)             | 36.0           |
| Base 12-2 (3, 6)                | 46.2 (37.8 + 8.4)             | 36.0           |
| Base 12-2 (K = 10) (7)          | 46.2                          | 36.0 + 76.2    |
| Base 12-2 (N = 4k) (8, 10)      | 46.2                          | 36.0 + 45.4    |
| Hybrid 12-2 (N = 4k) (11)       | 43.6 (37.8 + 5.8)             | 36.0 + 45.4    |
| Hybrid 12-3 (N = 4k) (12)       | 46.8 (37.8 + 8.9)             | 36.0 + 45.4    |
| Multi-decoder Base 12-2 (N = 4k)| 206.0 (37.8 + 20×33.6)        | 36.0 + 45.4    |

Table 15: Number of parameters in the TED Talks models.

| Models                          | Params without embeddings (M) | Embeddings (M) |
|--------------------------------|-------------------------------|----------------|
| Big 6-6 (16, 19)                | 176.4 (75.6 + 100.8)          | 70.7           |
| Big 6-2 (37)                    | 109.2 (75.6 + 33.6)           | 70.7           |
| Big 12-2 (17, 20)               | 184.7 (151.1 + 33.6)          | 70.7           |
| Big 12-2 (N = 16k) (21)         | 184.7                         | 70.7 + 332.3   |
| Big 12-2 (N = 8k) (22, 24)      | 184.7                         | 70.7 + 172.2   |
| Hybrid 12-2 (N = 8k) (25)       | 174.2 (151.1 + 23.1)          | 70.7 + 172.2   |
| Hybrid 12-3 (N = 8k) (26)       | 186.8 (151.1 + 35.7)          | 70.7 + 172.2   |
| Multi-decoder Big 12-2 (N = 8k) | 823.0 (151.2 + 20×33.6)       | 70.7 + 172.2   |

Table 16: Number of parameters in the ParaCrawl models.
### Table 17: Time benchmark across different TED Talks model sizes and decoding settings.

Time in seconds spent decoding the concatenation of all X→EN TED valid sets (averages over 3 runs). Note that to mimic a true online setting, no sorting by length is applied (i.e., buffer_size=batch_size). We modify fairseq’s code to avoid the slow beam search code when beam=1 (which unnecessarily computes and stores log probabilities). We see that in this setting, vocabulary size has a minor impact on speed.

| Time (s) | Parameters | English-centric | Multi-parallel (N_{train} = 4k) |
|----------|------------|-----------------|-----------------------------|
|          |            | Base 6-6        | Base 6-2 | Base 12-2 | Base 12-2 | Hybrid 12-2 |
|          |            | (2)              | (3)        |            |            |            |
| Total    | beam=1 bs=1| 18259           | 8455      | 9177       | 8483       | 5227       |
|          | beam=5 bs=1| 34216           | 18215     | 18861      | 15351      | 10617      |
|          | beam=1 bs=64| 1364       | 555       | 551        | 622        | 273        |
|          |            | 2330           | 1334      | 1329       | 1050       | 691        |
| Encoder  | beam=5 bs=64| 1435             | 554       | 545        | 548        | 234        |
| Decoder  |            | 477             | 162       | 159        | 168        | 71         |
| Self-attn / RNN | beam=5 bs=64| 399             | 133       | 132        | 139        | 60         |
| Cross-attn |            | 49              | 48        | 48         | 21         | 21         |
| Softmax  |            | 350             | 341       | 334        | 38         | 38         |

| Time (s) | Parameters | English-centric | Multi-parallel (N_{train} = 8k) |
|----------|------------|-----------------|-----------------------------|
|          |            | Big 6-6 | Big 6-2 | Big 12-2 | Big 12-2 | Hybrid 12-2 |
|          |            | (21)    | (37)    | (16)     | (24)     | (25)       |
| Total    | beam=1 bs=1| 14464   | 6131    | 6535     | 6544     | 3519       |
|          | beam=5 bs=1| 23990   | 12646   | 13264    | 10873    | 7034       |
|          | beam=1 bs=64| 912    | 375     | 402      | 397      | 195        |
|          |            | 1492    | 854     | 902      | 708      | 495        |
| Encoder  | beam=5 bs=64| 899     | 350     | 364      | 343      | 155        |
| Decoder  |            | 296    | 101     | 105      | 105      | 44         |
| Self-attn / RNN | beam=5 bs=64| 257     | 85      | 89       | 88       | 42         |
| Cross-attn |            | 39     | 38      | 39       | 19       | 20         |
| Softmax  |            | 207    | 204     | 209      | 40       | 40         |

Table 18: Time benchmark across different ParaCrawl model sizes and decoding settings. Time in seconds spent decoding the concatenation of all →EN TED2020-valid sets (averages over 3 runs).

| Model | →EN     | ←EN     | / EN     | WPS     |
|-------|---------|---------|---------|---------|
| 2     | Base 6-6 | 31.8 / 30.5 | 24.2 / 23.6 | 13.5 / 13.2 | 724 / 385 |
| 3     | Base 12-2 | 33.6 / 32.9 | 24.3 / 24.2 | 14.1 / 14.1 | 1321 / 765 |
|       | + Multi-parallel |
| 5     | Base 6-6 | 32.8 / 31.5 | 24.3 / 23.4 | 16.3 / 15.6 | 760 / 390 |
| 6     | Base 12-2 | 33.5 / 32.7 | 24.5 / 24.2 | 16.3 / 16.0 | 1258 / 723 |
| 11    | Hybrid 12-2 | 32.8 / 31.7 | 23.5 / 23.6 | 16.1 / 15.6 | 2546 / 1403 |
| 12    | Hybrid 12-3 | 32.9 / 31.6 | 23.7 / 23.6 | 16.1 / 15.6 | 2279 / 1338 |

Table 19: TED Talks experiments on another MT framework show that our results are reproducible. The first number in each cell is the value obtained with fairseq and the second number is obtained with our internal TensorFlow implementation. Both implementations share the same hyper-parameters, with one notable exception: in the TensorFlow implementation, source/target embeddings are not shared. Additionally, the Hybrid models trained with fairseq use train-time BPE filtering, while the TensorFlow models do not.
Figure 7: BLEU scores on ↔EN TED2020 test by the ParaCrawl models with beam search or greedy search (23, 24, 25, 28) according to sentence length. Top: beam search. Bottom: greedy search. Left: →EN translation. Right: ←EN translation.
| Model                               | Test BLEU | Test chrF |
|-------------------------------------|-----------|-----------|
|                                     | →EN      | ←EN      | / EN  | →EN      | ←EN      | / EN  | WPS |
| SOTA (Philip et al., 2020)          |           |           |       |           |           |       |     |
| Bilingual                           | 32.4      | 24.1      | 15.8  |           |           |       |     |
| Best multi                          | 32.3      | 24.1      | 15.8  |           |           |       |     |
|                                     |           |           |       |           |           |       |     |
| English-centric                     |           |           |       |           |           |       |     |
| 1 Small 6-6                         | 31.6      | 23.1      | 11.6 (14.4) | .527 | .474 | .342 (.373) | 703 |
| 2 Base 6-6                          | 31.8      | 24.2      | 13.5 (15.0) | .529 | .486 | .370 (.381) | 753 |
| 29 (2) + dec. lang code             | 32.0      | 23.9      | 5.4 (14.9)  | .532 | .484 | .186 (.381) | 766 |
| 36 Base 6-2                         | 32.5      | 23.3      | 12.7 (14.6) | .536 | .475 | .358 (.377) | 1271 |
| 3 Base 12-2, init with (2)          | 33.6      | 24.3      | 14.1 (15.4) | .548 | .487 | .378 (.389) | 1287 |
| 33 Base 12-2, pre-norm              | 33.1      | 24.2      | 13.2 (15.3) | .544 | .487 | .368 (.388) | 1312 |
| 34 Base 12-2, enc. pre-norm         | 33.3      | 24.3      | 12.7 (15.3) | .546 | .487 | .359 (.389) | 1240 |
| 30 (3) + dec. lang code             | 33.4      | 22.8      | 0.7 (14.7)  | .547 | .472 | .102 (.380) | 1307 |
| 38 (2) + TED-59                     | 25.5      | 19.9      | 9.0 (11.5)  | .455 | .423 | .282 (.324) | 715 |
|                                     |           |           |       |           |           |       |     |
| + Multi-parallel                    |           |           |       |           |           |       |     |
| 5 Base 6-6                          | 32.8      | 24.3      | 16.3  | .540 | .482 | .395      | 732 |
| 39 (5) + K_{test} = 10              | 32.7      | 24.1      | 16.3  | .539 | .481 | .394      | 826 |
| 40 (5) + N_{test} = 4k              | 30.8      | 22.3      | 15.2  | .519 | .463 | .382      | 868 |
| 6 Base 12-2                         | 33.5      | 24.5      | 16.3  | .546 | .485 | .395      | 1203 |
| 7 (6) + K_{test} = 10               | 33.4      | 24.3      | 16.2  | .545 | .484 | .394      | 1539 |
| 8 (6) + N_{test} = 4k               | 31.5      | 22.5      | 15.2  | .523 | .466 | .382      | 1457 |
| 41 Hybrid 12-2                      | 32.9      | 23.7      | 16.2  | .544 | .476 | .394      | 1724 |
| 42 Hybrid 12-3                      | 32.9      | 23.8      | 16.2  | .543 | .476 | .395      | 1533 |
| + BPE filtering (N_{train} = 4k)    |           |           |       |           |           |       |     |
| 9 Base 6-6                          | 32.9      | 24.2      | 16.3  | .541 | .481 | .394      | 789 |
| 43 (9) + dec. lang code             | 32.7      | 23.9      | 16.2  | .540 | .478 | .393      | 891 |
| 44 Base 6-2                         | 32.5      | 23.4      | 15.6  | .537 | .472 | .386      | 1595 |
| 10 Base 12-2                        | 33.3      | 24.3      | 16.3  | .546 | .481 | .395      | 1552 |
| 45 (10) + dec. lang code            | 32.5      | 23.0      | 15.7  | .540 | .471 | .389      | 1514 |
| 11 Hybrid 12-2                      | 32.8      | 23.5      | 16.1  | .543 | .474 | .393      | 2422 |
| 12 Hybrid 12-3                      | 32.9      | 23.7      | 16.1  | .543 | .475 | .394      | 2145 |
| + Multi-decoder                     |           |           |       |           |           |       |     |
| 13 (2) → 6-2                        | 33.0      | 24.2      | 16.0  | .543 | .480 | .391      | 1608 |
| 46 (36) → 6-2                       | 33.3      | 24.5      | 16.2  | .545 | .482 | .394      | 1548 |
| 14 (3) → 12-2                       | 33.8      | 25.1      | 16.7  | .551 | .490 | .401      | 1614 |
| 47 (14) w/o lang code               | 33.6      | 24.3      | 16.6  | .550 | .481 | .399      | 1567 |

Table 20: Test BLEU, chrF scores and decoding speed of **TED Talks models** of various depths. SOTA’s “best multi” is a Transformer Small 6-6 multi-parallel model with adapter layers. WPS: speed in words per second for →EN translation. Scores in parentheses are obtained by pivot translation through English. (47) does not use any language code during the multi-decoder finetuning stage.
| Model | TED2020 test chrF | FLORES devtest spBLEU | WPS |
|-------|-----------------|----------------------|-----|
|       | →EN ←EN / EN    | →EN ←EN / EN        |     |
|       |                 |                      |     |
| Goyal et al. (2021) | – – – | 32.4 31.9 25.7 | – |
| **English-centric** | | | |
| 15 | Base 6-6 | .551 .545 .399 (.456) | 34.0 31.9 16.0 (22.5) | 656 |
| 16 | Big 6-6 | .580 .569 .401 (.483) | 38.8 36.4 18.5 (27.0) | 623 |
| 37 | Big 6-2 | .572 .565 .373 (.477) | 37.7 35.3 15.3 (25.8) | 1077 |
| 17 | Big 12-2 | .585 .575 .435 (.488) | 39.6 37.1 21.1 (27.6) | 1033 |
| 48 | Wide 12-2 | .590 .581 (.494) | 40.7 38.9 19.0 (29.0) | 870 |
| **+ Multi-parallel** | | | |
| 19 | Big 6-6 | .574 .564 .481 | 37.9 35.6 26.8 | 595 |
| 49 | Big 6-6 (T = 5) | .573 .564 .481 | 37.6 35.6 26.8 | 608 |
| 20 | Big 12-2 | .581 .570 .486 | 39.0 36.2 27.6 | 1030 |
| 21 (20) + $N_{test} = 16k$ | .580 .570 .486 | 38.4 35.8 27.2 | 1328 |
| 22 (20) + $N_{test} = 8k$ | .563 .560 .478 | 33.5 33.4 25.3 | 1283 |
| 50 | Hybrid 12-3 | .576 .564 .482 | 38.2 35.4 26.9 | 1313 |
| **+ Multi-decoder** | | | |
| 23 | Big 6-6 | .572 .564 .480 | 37.5 35.5 26.6 | 679 |
| 51 | Big 6-2 | .565 .558 .472 | 36.2 34.0 25.1 | 1305 |
| 24 | Big 12-2 | .580 .570 .486 | 38.8 36.2 27.4 | 1261 |
| 25 | Hybrid 12-2 | .573 .562 .481 | 37.9 34.9 26.5 | 1796 |
| 52 (25) + 200k steps | .576 .564 .483 | 38.2 34.9 26.8 | 1861 |
| 26 | Hybrid 12-3 | .574 .563 .482 | 38.0 34.8 26.6 | 1659 |
| 53 (26) + 200k steps | .577 .565 .483 | 38.3 35.6 27.1 | 1770 |
| **+ BPE filtering (N_{train} = 8k)** | | | |
| 27 (16) → 6-2 | .573 .566 .483 | 38.0 36.3 27.3 | 1333 |
| 54 (27) + 200k steps | .575 .569 .484 | 38.1 36.7 27.5 | 1360 |
| 55 (23) → 6-2 | .573 .566 .483 | 37.9 36.4 27.3 | 1372 |
| 56 (37) → 6-2 | .574 .566 .483 | 38.0 36.2 27.4 | 1295 |
| 57 (51) → 6-2 | .574 .567 .484 | 37.9 36.1 27.4 | 1323 |
| 28 (17) → 12-2 | .583 .573 .490 | 39.5 37.4 28.7 | 1270 |
| 58 (28) + 200k steps | .585 .574 .492 | 39.8 37.5 28.9 | 1221 |
| 59 (24) → 12-2 | .583 .572 .491 | 39.5 37.2 28.6 | 1274 |
| 60 (28) + freeze enc. | .584 .574 .478 | 39.5 37.5 26.5 | 1238 |
| 61 (59) + freeze enc. | .581 .574 .489 | 39.2 37.3 28.3 | 1219 |
| 62 (28) w/o lang code | .583 .573 .491 | 39.3 37.3 28.6 | 1215 |
| 63 (62) + 200k steps | .585 .575 .493 | 39.7 37.7 29.0 | 1242 |
| 64 (28) + shared embed | .582 .572 .489 | 39.3 37.2 28.4 | 1224 |

Table 21: TED2020 test chrF, FLORES devtest spBLEU (Goyal et al., 2021) and decoding speed of ParaCrawl models of various depths. WPS: speed in words per second for $\rightarrow$EN translation. Scores in parentheses are obtained by pivot translation through English. (48) uses the same architecture as (17) but with a feed-forward dimension of 8192 and is trained for 1M steps on ParaCrawl v8. (62) is fine-tuned like (28) but without language codes, similarly to Lyu et al. (2020) (while the pre-trained model has language codes). (60, 61) freeze the shared encoder parameters during the multi-decoder training stage. (64) is trained with shared target embeddings and $N_{train} = 8k$ (instead of training one separate embedding matrix per decoder).