How far can we get with one GPU in 100 hours?

CoAStaL at MultiIndicMT Shared Task

Aralikatte, Rahul; Murrieta Bello, Héctor Ricardo; Hershcovich, Daniel; Bollmann, Marcel; Søgaard, Anders

Published in:
Proceedings of the 8th Workshop on Asian Translation (WAT2021)

DOI:
10.18653/v1/2021.wat-1.24

Publication date:
2021

Document version
Publisher's PDF, also known as Version of record

Document license:
CC BY

Citation for published version (APA):
How far can we get with one GPU in 100 hours?
CoAStaL at MultiIndicMT Shared Task

Rahul Aralikatte  Héctor Ricardo Murrieta Bello  Miryam de Lhoneux
Daniel Hershcovich  Marcel Bollmann  Anders Søgaard
Department of Computer Science
University of Copenhagen
{rahul,ml,dh,marcel,soegaard}@di.ku.dk  xhd160@alumni.ku.dk

Abstract

This work shows that competitive translation results can be obtained in a constrained setting by incorporating the latest advances in memory and compute optimization. We train and evaluate large multilingual translation models using a single GPU for a maximum of 100 hours and get within 4-5 BLEU points of the top submission on the WAT 2021 leaderboard. We also benchmark standard baselines on the PMI corpus and re-discover well-known shortcomings of current translation metrics.

1 Introduction

Machine Translation is one of the few tasks in NLP which has the luxury of data. Due to the efforts of the community over the last two decades (Koehn, 2005; Tiedemann, 2012, 2020), most major languages of the world have millions of translated sentence pairs with English. With the introduction of sequence to sequence models (Sutskever et al., 2014; Cho et al., 2014), transformers (Vaswani et al., 2017), and large pre-trained language models (Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019; Liu et al., 2019), the accuracy of machine translation models has almost risen to that of humans (Wu et al., 2016). Yet, the ability to train such models is limited by the availability of compute. Today’s state-of-the-art models are trained by industry research labs, using large compute infrastructure which is usually unavailable or unaffordable to others. Such training is also shown to have large carbon footprints (Strubell et al., 2019).

In this work, we show that competitive translation performance can be achieved even with limited resources. We first train a statistical MT system that does not require GPUs, as a baseline. Next, we run inference on the best publicly available pre-trained models to benchmark their performance. Finally, we train graph2seq, seq2seq, and text2text models, which progressively perform better. All our experiments are constrained both in compute¹ and training time: we use one NVIDIA Titan RTX GPU for a maximum of 100 hours. Our main findings are: (i) pre-trained seq2seq and text2text models perform the best, especially when trained only on the PMI corpus, (ii) the benefits of pre-trained multilingual language models diminish for Indic language decoding due to their under-representation in pre-training data, and (iii) a small empirical evaluation on 2 languages shows that the prediction fluency and faithfulness start plateauing at 100 hours.

2 Data

The MultiIndicMT data is a combination of parallel corpora from different sources as shown in Tab. 1. It contains translations from 10 Indic languages to English. The Indic languages included are Bengali (BN), Gujarati (GU), Hindi (HI), Kannada (KN), Malayalam (ML), Marathi (MR), Oriya (OR), Punjabi (PA), Tamil (TA), and Telugu (TE). The training split contains about 11 million translation pairs.

Table 1: Sources of MultiIndicMT data.

<table>
<thead>
<tr>
<th>Source</th>
<th>Language(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVIT-PIB (2020)</td>
<td>BN, GU, HI, ML, MR, OR, TA, TE</td>
</tr>
<tr>
<td>JW (2019)</td>
<td>BN, GU, HI, KN, ML, MR, PA, TA, TE</td>
</tr>
<tr>
<td>TED (2012)</td>
<td>BN, GU, HI, KN, ML, MR, PA, TA, TE</td>
</tr>
<tr>
<td>PMIndia (2020)</td>
<td>BN, GU, HI, KN, ML, MR, OR, PA, TA, TE</td>
</tr>
<tr>
<td>Bible-uedin (2014)</td>
<td>GU, HI, KN, ML, MR, TE</td>
</tr>
<tr>
<td>OpenSubtitles (2016)</td>
<td>BN, HI, ML, TA, TE</td>
</tr>
<tr>
<td>WikiMatrix (2019)</td>
<td>HI, ML, MR, TA, TE</td>
</tr>
<tr>
<td>Wiki Titles (2021)</td>
<td>GU, TA</td>
</tr>
<tr>
<td>ALT (2016)</td>
<td>BN, HI</td>
</tr>
<tr>
<td>IITB 3.0 (2018)</td>
<td>HI</td>
</tr>
<tr>
<td>NLPC (2020)</td>
<td>TA</td>
</tr>
<tr>
<td>UFAL EriTam (2012)</td>
<td>TA</td>
</tr>
<tr>
<td>Uka Tarsadia (2019)</td>
<td>GU</td>
</tr>
<tr>
<td>MTEnglish2Odia (2018)</td>
<td>OR</td>
</tr>
<tr>
<td>OdiEnCorp 2.0 (2020)</td>
<td>OR</td>
</tr>
</tbody>
</table>

¹which in turn constrains the number of model parameters
from these languages. The development and test splits contain 1000 and 2390 11-way parallel sentences taken from the PMIndia corpus (Haddow and Kirefu, 2020), respectively.

Analysis To understand the data better, a small analysis is performed by randomly sampling 100 sentences from each language the authors can read (HI and KN). Overall, the translations are of high quality, except in a few sources where the parallel sentences are automatically extracted. For example, we found that JW (Agić and Vušić, 2019) has alignment issues, where a part of the translation is moved to the next line, thereby starting a chain of misalignments, as shown in Fig. 1. We manually annotate 100 translations for fluency and faithfulness on a scale of 0-5 and get a score of 4.01 for fluency and 3.54 for faithfulness.

3 Models

We train four types of models: (i) a phrase-based statistical model, (ii) a graph-to-text model, (iii) a sequence-to-sequence model, and (iv) a text-to-text model. Brief descriptions of the models are given below.

3.1 Moses

We train a statistical phrase-based model with Moses (Koehn et al., 2007) using default settings, following the guidelines for training a baseline.2 We prune words that occur less than three times in the corpus and use the same tokenizer as for the other models and de-tokenize predictions before evaluating. We train a separate model for each language pair and use the respective development set for tuning before translating the test set.

3.2 GRAPH-TO-TEXT model

We also train a graph2seq model with a GCN (Kipf and Welling, 2016) encoder and LSTM decoder. In addition to text, we input the source syntax trees obtained from a parser trained on Universal Dependencies (Nivre et al., 2016). We borrow hyperparameter settings from Bastings et al. (2017) and input a bag of source words to the encoder and expect subword units from the decoder. We train separate models for each language pair.

3.3 SEQ2SEQ model

For training multilingual models, we use pre-trained transformer-based language models to initialize the encoder and decoder of our seq2seq models. For English, we use standard uncased BERT-Base (Devlin et al., 2019) and for Indic languages, we use MuRIL (Khanuja et al., 2021). MuRIL’s architecture is similar to BERT and is pre-trained on 17 Indic languages including all ten required for our translation task. It is pre-trained on publicly available corpora from Wikipedia and Common Crawl. It also uses automatically translated and transliterated data for pre-training. We add cross-attention between the encoder and decoder following Rothe et al. (2020).

The model has 375M trainable parameters. When the decoder is multilingual, we follow previous works and force a language identifier as the BOS token. We use a learning rate of $5 \times 10^{-5}$ and a batch size of 12. We truncate sequences to a maximum length of 128 and use a cosine learning rate scheduler with a warmup of 10,000 steps. We denote our models as BERT2MURIL and MURIL2BERT when translating from and to En-
We report results in both English to Indic, and Indic to English. All other hyper-parameters remain unchanged. We train the model with a learning rate of $3 \times 10^{-5}$. (mT5-large; Xue et al., 2020) model on our translation task. This model is pre-trained on mC4, a multilingual version of the Common Crawl consisting of text from 101 languages. It contains 1.2B training examples of text from 101 languages. It contains 1.2B training examples. It contains 1.2B training. It contains 1.2B training.

<table>
<thead>
<tr>
<th>Model</th>
<th>Bn chrF</th>
<th>bleu</th>
<th>Gu chrF</th>
<th>bleu</th>
<th>Hi chrF</th>
<th>bleu</th>
<th>Kn chrF</th>
<th>bleu</th>
<th>Ml chrF</th>
<th>bleu</th>
<th>Mr chrF</th>
<th>bleu</th>
<th>Or chrF</th>
<th>bleu</th>
<th>Pa chrF</th>
<th>bleu</th>
<th>Ta chrF</th>
<th>bleu</th>
<th>Te chrF</th>
<th>bleu</th>
</tr>
</thead>
<tbody>
<tr>
<td>m2m100 (418M)</td>
<td>0.30</td>
<td>2.98</td>
<td>0.11</td>
<td>0.40</td>
<td>0.48</td>
<td>21.21</td>
<td>0.15</td>
<td>0.05</td>
<td>0.21</td>
<td>0.69</td>
<td>0.31</td>
<td>3.96</td>
<td>0.05</td>
<td>0.06</td>
<td>0.11</td>
<td>0.66</td>
<td>0.20</td>
<td>1.43</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>m2m100 (1.2B)</td>
<td>0.35</td>
<td>4.48</td>
<td>0.16</td>
<td>1.35</td>
<td>0.49</td>
<td>22.22</td>
<td>0.18</td>
<td>0.15</td>
<td>0.28</td>
<td>1.29</td>
<td>0.35</td>
<td>6.19</td>
<td>0.05</td>
<td>0.04</td>
<td>0.16</td>
<td>1.84</td>
<td>0.23</td>
<td>1.26</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Moses (PMI)</td>
<td>0.38</td>
<td>4.90</td>
<td>0.46</td>
<td>12.4</td>
<td>0.48</td>
<td>15.7</td>
<td>0.44</td>
<td>8.00</td>
<td>0.39</td>
<td>2.60</td>
<td>0.41</td>
<td>7.50</td>
<td>0.42</td>
<td>8.40</td>
<td>0.44</td>
<td>14.1</td>
<td>0.42</td>
<td>5.02</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>Moses (all)</td>
<td>0.40</td>
<td>5.20</td>
<td>0.48</td>
<td>14.3</td>
<td>0.50</td>
<td>17.1</td>
<td>0.44</td>
<td>9.10</td>
<td>0.36</td>
<td>2.10</td>
<td>0.41</td>
<td>7.30</td>
<td>0.40</td>
<td>8.90</td>
<td>0.46</td>
<td>16.7</td>
<td>0.48</td>
<td>8.20</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>GCN (PMI)</td>
<td>0.40</td>
<td>5.20</td>
<td>0.48</td>
<td>14.3</td>
<td>0.50</td>
<td>17.1</td>
<td>0.44</td>
<td>9.10</td>
<td>0.36</td>
<td>2.10</td>
<td>0.41</td>
<td>7.30</td>
<td>0.40</td>
<td>8.90</td>
<td>0.46</td>
<td>16.7</td>
<td>0.48</td>
<td>8.20</td>
<td>0.35</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Character F1 and BLEU scores of English to Indic translations.

<table>
<thead>
<tr>
<th>Model</th>
<th>Bn chrF</th>
<th>bleu</th>
<th>Gu chrF</th>
<th>bleu</th>
<th>Hi chrF</th>
<th>bleu</th>
<th>Kn chrF</th>
<th>bleu</th>
<th>Ml chrF</th>
<th>bleu</th>
<th>Mr chrF</th>
<th>bleu</th>
<th>Or chrF</th>
<th>bleu</th>
<th>Pa chrF</th>
<th>bleu</th>
<th>Ta chrF</th>
<th>bleu</th>
<th>Te chrF</th>
<th>bleu</th>
</tr>
</thead>
<tbody>
<tr>
<td>m2m100 (418M)</td>
<td>0.43</td>
<td>13.8</td>
<td>0.10</td>
<td>0.18</td>
<td>0.55</td>
<td>26.0</td>
<td>0.10</td>
<td>0.08</td>
<td>0.32</td>
<td>7.45</td>
<td>0.40</td>
<td>13.4</td>
<td>0.15</td>
<td>0.75</td>
<td>0.36</td>
<td>9.65</td>
<td>0.25</td>
<td>2.44</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>m2m100 (1.2B)</td>
<td>0.44</td>
<td>14.7</td>
<td>0.10</td>
<td>0.16</td>
<td>0.55</td>
<td>26.7</td>
<td>0.09</td>
<td>0.16</td>
<td>0.36</td>
<td>11.0</td>
<td>0.41</td>
<td>14.1</td>
<td>0.15</td>
<td>0.42</td>
<td>0.36</td>
<td>9.08</td>
<td>0.23</td>
<td>1.26</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Moses (PMI)</td>
<td>0.37</td>
<td>6.00</td>
<td>0.46</td>
<td>10.6</td>
<td>0.49</td>
<td>12.6</td>
<td>0.42</td>
<td>8.30</td>
<td>0.39</td>
<td>5.90</td>
<td>0.41</td>
<td>7.50</td>
<td>0.42</td>
<td>8.40</td>
<td>0.44</td>
<td>14.1</td>
<td>0.42</td>
<td>5.02</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>Moses (all)</td>
<td>0.40</td>
<td>8.00</td>
<td>0.48</td>
<td>12.5</td>
<td>0.52</td>
<td>16.1</td>
<td>0.43</td>
<td>8.80</td>
<td>0.43</td>
<td>8.50</td>
<td>0.44</td>
<td>10.4</td>
<td>0.43</td>
<td>10.6</td>
<td>0.48</td>
<td>13.0</td>
<td>0.42</td>
<td>9.60</td>
<td>0.43</td>
<td></td>
</tr>
<tr>
<td>GCN (PMI)</td>
<td>0.40</td>
<td>8.30</td>
<td>0.49</td>
<td>12.8</td>
<td>0.54</td>
<td>14.8</td>
<td>0.44</td>
<td>10.9</td>
<td>0.48</td>
<td>11.5</td>
<td>0.48</td>
<td>13.5</td>
<td>0.46</td>
<td>7.20</td>
<td>0.45</td>
<td>12.6</td>
<td>0.43</td>
<td>14.3</td>
<td>0.45</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Character F1 and BLEU scores of Indic to English translations.

3This is the largest model we could train on our GPU without using optimization tricks.

4Note that we report local evaluation metrics which do not exactly match with the leaderboard numbers because of the differences in tokenization. We do this to avoid uploading multiple prediction files and overloading the evaluation server.

34

4 Results

We report results in both English to Indic, and Indic to English directions. We use character F1 and BLEU (Papineni et al., 2002), which are standard metrics to evaluate translations. We train two variants of all models: (i) only on the PMI corpus, and (ii) on the full training data. The English to Indic results are shown in Tab. 2 and the Indic to English results, in Tab. 3.

m2m100 We first benchmark the performance of the Many-to-Many multilingual model (m2m100; Fan et al., 2020) which is trained on non-English centric translation. It can translate to and from all Indic languages in our task, except Telugu. As expected, with no finetuning, both the small (418M parameters) and large (1.2B parameters) models perform poorly, on all languages except Hindi. This is expected as the other languages are under-represented in the mC4 dataset.

Moses We see that simple phrase-based translation works relatively well. Though significantly worse than the best model, Moses produces results comparable to that of mT5-large (all) in both directions. Although this can be attributed to mT5-large being under-trained, it gives us an insight into
the ability of simpler models to learn quickly in constrained environments. We also note that just training on the PMI corpus gives a result that is almost on par with the results obtained by training on the entire training split. The model trained on PMI even surpasses the other model, on Kannada indicating a strong in-domain training bias.

**GCN** In this setup, we only train on the PMI corpus due to time constraints. We find that while it comfortably surpasses Moses, it also comes close to the much bigger models, especially when translating to Indic languages. It is to be noted that, this small gap in results can be mainly attributed to the lack of convergence of the bigger models, as discussed next.

**mT5** mT5 can translate to and from all Indic languages required by our task, except Oriya. We note that the model trained only on the PMI corpus is always better than the model trained on the complete data. We postulate that 100 hours is not enough time for the model to converge on the full data. We also see that mT5’s performance is far superior compared to all other models for Indic to English translation. This may be expected as the model is pre-trained to generate fluent English text. For English to Indic translation, mT5 performs on-par or slightly worse than bert2muril finetuned on PMI data, except for Hindi and Tamil, where it is better.

**MuRIL and BERT** Following the mT5 models, these models also perform better when trained only on the PMI corpus as it fails to converge on the larger data in the given time. As an additional step, we finetune these under-fit models on the PMI data for 5 hours and see a significant performance gain in the English to Indic direction (bert2muril). The model outperforms the much bigger mT5 on 7 languages with Gujarati, Hindi, and Tamil being the exceptions. However, finetuning does not seem to have a major effect in the other direction (muril2bert). As in the case of mT5, we believe that the BERT decoder’s pre-training subsumes any gains from extra finetuning.

**Official Evaluation** Since Tab. 2 and 3 show BLEU scores obtained by evaluating the generated predictions locally, they do not exactly match the official scores on the leaderboard. For a fair comparison, we present both local and official BLEU scores of our best submissions in Tab. 4. We see that the scores are similar when translating from Indic languages to English. But when translating from English, the official scores are often significantly higher. This is a result of our use of minimal tokenization (mteval-v13a) before computing BLEU, while the official evaluation uses the Indic-tokenizer (Kunchukuttan, 2020).

### Table 4: Comparison of BLEU scores obtained during local and official evaluations.

<table>
<thead>
<tr>
<th>Language</th>
<th>En→*</th>
<th>*→En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>8.89</td>
<td>11.1</td>
</tr>
<tr>
<td>Gujarati</td>
<td>20.8</td>
<td>20.4</td>
</tr>
<tr>
<td>Hindi</td>
<td>26.5</td>
<td>31.7</td>
</tr>
<tr>
<td>Kannada</td>
<td>16.0</td>
<td>16.1</td>
</tr>
<tr>
<td>Malayalam</td>
<td>5.91</td>
<td>6.27</td>
</tr>
<tr>
<td>Marathi</td>
<td>14.3</td>
<td>14.5</td>
</tr>
<tr>
<td>Oriya</td>
<td>15.3</td>
<td>15.7</td>
</tr>
<tr>
<td>Punjabi</td>
<td>24.1</td>
<td>27.2</td>
</tr>
<tr>
<td>Tamil</td>
<td>10.1</td>
<td>10.0</td>
</tr>
<tr>
<td>Telugu</td>
<td>6.54</td>
<td>12.9</td>
</tr>
</tbody>
</table>

Figure 2: Increase in BLEU score across languages when trained on the full training data, at different intervals of time.
hours is not enough time for the models to converge when trained on the full training set. Fig 2 shows the BLEU scores of BERT2MuRIL model after 80 and 100 hours of training, respectively. We see that the model gets significantly better in the last 20 hours. A 5 hour finetuning with the PMI corpus, further increases its performance. This clearly shows that the model would become more accurate if it is trained for a longer period or with more compute.

To establish whether an increase in BLEU scores corresponds to an increase in the fluency and faithfulness of the translations, we manually annotate 50 Hindi and Kannada test predictions from the best model to find that the increase in both cases is marginal. In the 20 additional training hours, the fluency and faithfulness increased by 0.005 and 0.01 respectively which suggests that BLEU may not be the best metric to quantify the goodness of translation systems, as shown in works like Zhang et al. (2004); Callison-Burch et al. (2006).

6 Conclusion

In this work, we show that it is possible to get competitive translation results using a single GPU for a limited amount of time by carefully selecting and training large pre-trained encoder-decoder models. We also show that we can train models which have more than $10^9$ trainable parameters using the latest advances in GPU resource optimization. Finally, through a small empirical study, we find that while longer training can increase BLEU scores, it may not increase their fluency and faithfulness.

Acknowledgements

We thank the reviewers for their valuable feedback. Rahul Aralikatte and Anders Søgaard were funded by a Google Focused Research Award. Miryam de Lhoneux was funded by the Swedish Research Council (grant 2020-00437). Marcel Bollmann was funded from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 845995.

References


210


Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In LREC.