WNGT 2020 Efficiency Shared Task
Kenneth Heafield, Yusuke Oda, Graham Neubig

https://www.aclweb.org/anthology/2020.ngt-1.1
https://sites.google.com/view/wngt20/efficiency-task

¹Corruptly, both organizer and participant.
In case you haven’t heard, the new unit for measuring computation runtime is TPU core years. But, if you missed that memo, since the numbers are already in the hundreds, you may as well get ahead of the game and start quoting your runtimes in TPU core centuries.

#NLProc

Dmitry (Dima) Lepikhin @lepihin · Jul 1

arxiv.org/abs/2006.16668

We scaled the Transformer model with Sparsely-Gated Mixture-of-Experts using GShard, and trained a 600B multilingual translation model in about 4 days (for 100 languages) achieving 13.5 BLEU gain compared to the baseline.
Goal: Efficient Machine Translation

Present task: inference $\rightarrow$ production
Future task: efficient training?
WMT 2019 English–German constrained news task.

State-of-the-art systems submit to the latest WMT
⇒ There is no such thing as state-of-the-art on WMT14!
Also, recycle WMT 2019 systems as teachers.
Awkward timing with WMT

2020 training data not final at start, test set unavailable at end. Root cause: WNGT at ACL, WMT at EMNLP.
Coordinate with WMT more?
Test Set

Last year

≈1s to translate \implies too small
Banned a team for memorizing known test set
Test Set

Last year

≈1s to translate $\implies$ too small
Banned a team for memorizing known test set

Before deadline

1 million sentences
$\leq$100 space-separated words/sentence
Unspecified test set hidden in input
## Test Set

### Last year

\[ \approx 1 \text{s to translate} \implies \text{too small} \]

Banned a team for memorizing known test set

### Before deadline

- 1 million sentences
- \( \leq 100 \) space-separated words/sentence
- Unspecified test set hidden in input

### After deadline

- WMT plus filler: EMEA, Tatoeba, German Federal
- Shuffled, also score parallel filler data

Need a surprise evaluation set. WMT20 not ready yet.
→ Uh, average old WMT test sets?
→ WMT12 has sentences longer than 100 words.
→ \textbf{WMT1*: average sacrebleu of WMT11, WMT13–19}

See paper supplement for individual WMT scores.
Problem: participants likely tuned on WMT sets.
“use human evaluation to verify claims in experiments that use metrics such as BLEU” –Reviewer of my EU project

“BLEU has been surpassed by various other metrics”
–Mathur et al, ACL 2020

→ Submitted fast Czech systems to WMT20 with Charles University.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825303.
Recent hardware with 8-bit optimization:

**GPU**  NVidia T4
g4dn.xlarge on Amazon Web Services $0.526/hr

**CPU**  Intel Xeon Platinum 8275CL (Cascade Lake) dual socket
c5.metal on Amazon Web Services $4.08/hr
Single-core and all-core tracks (48 physical cores)

Provided credits for participants to develop with.

Amazon, Intel, and NVidia have contributed to my research.
Three teams

Multiple submission encouraged!

<table>
<thead>
<tr>
<th></th>
<th>GPU</th>
<th>CPU 1 core</th>
<th>CPU all core</th>
</tr>
</thead>
<tbody>
<tr>
<td>NiuTrans</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>OpenNMT</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>UEdin</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

UEdin’s CPU submissions had a memory leak → shown with/without fix.
Pareto Comparison

Submissions have varying quality and efficiency. Unclear how much quality loss to tolerate.

Pareto comparison: quality $\geq$ baseline and efficiency $\geq$ baseline.

More efficient with same quality
... or better quality with same efficiency.
Primary: wall clock time.
Words per second based on 15,048,961 untokenized words.

Supplementary data: CPU time.
Model size: parameters, BPE, shortlists, etc.

Total Docker size: model, part of Ubuntu, code
OpenNMT won Docker with 122–308 MB; others 432–933 MB.
Model size, all platforms

![Graph showing model size vs WMT1* BLEU scores across different platforms (NiuTrans, OpenNMT, UEdin). The x-axis represents model size in MB, and the y-axis shows WMT1* BLEU scores. The graph includes markers for each model, with different colors representing different platforms.](attachment:image.png)
Peak RAM usage

GPU: polling nvidia-smi
CPU: memory.max_usage_in_bytes
Efficiency Task

All participants had something Pareto optimal.

System descriptions:
https://sites.google.com/view/wngt20/programme

I am opening the task for rolling submission.
What’s missing

Allowed batching in all conditions
→ What about latency?

Where are the non-autoregressive people?
→ Non-autoregressive: a case study in poor evaluation.
Latency

Latency is average time to translate one sentence. Experiments with Edinburgh’s systems; sorry I asked too late.
Batching is Important for Speed

Task Definition
Efficiency Results
Latency
Non-autoregressive
Recommendation
Latency: 10.3–71.7 ms!

![Graph showing latency results for CPU and GPU, with WMT1* BLEU scores on the y-axis and latency in ms on the x-axis. The graph includes data points for both CPU and GPU, with different symbols indicating different efficiency results.]
Autoregressive MT latency is 10.3–71.7 ms, often <30 ms.

So what’s up with this table from Jiatao Gu et al (2018)?

<table>
<thead>
<tr>
<th>Models</th>
<th>WMT14</th>
<th>WMT16</th>
<th>IWSLT16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>En→De</td>
<td>De→En</td>
<td>En→Ro</td>
</tr>
<tr>
<td>NAT</td>
<td>17.35</td>
<td>20.62</td>
<td>26.22</td>
</tr>
<tr>
<td>NAT (+FT)</td>
<td>17.69</td>
<td>21.47</td>
<td>27.29</td>
</tr>
<tr>
<td>NAT (+FT + NPD s = 10)</td>
<td>18.66</td>
<td>22.41</td>
<td>29.02</td>
</tr>
<tr>
<td>NAT (+FT + NPD s = 100)</td>
<td>19.17</td>
<td>23.20</td>
<td>29.79</td>
</tr>
<tr>
<td>Autoregressive (b = 1)</td>
<td>22.71</td>
<td>26.39</td>
<td>31.35</td>
</tr>
<tr>
<td>Autoregressive (b = 4)</td>
<td>23.45</td>
<td>27.02</td>
<td>31.91</td>
</tr>
</tbody>
</table>

Task Definition | Efficiency Results | Latency | Non-autoregressive | Recommendation |
Replicating Gu et al (2018)’s setup

Do not try this at home or work.

WMT14
State-of-the-art is latest WMT.
Replicating Gu et al (2018)’s setup

Do not try this at home or work.

**WMT14**
State-of-the-art is latest WMT.

**Tokenized BLEU**
Tokenization differences $\Rightarrow$ BLEU scores are not comparable.
But many non-autoregressive papers compare anyway.
Use sacrebleu instead.

P100, latency on IWSLT 2016 en-de dev.
Real baselines for Gu et al (2018)

Latency (ms) on P100, IWSLT 2016 dev

Gu et al (2018) autoregressive
Gu et al (2018) non-autoregressive
Marian 2018

Weirdly Tokenized WMT14 BLEU

Task Definition
Efficiency Results
Latency
Non-autoregressive
Recommendation
Weirdly Tokenized WMT14 BLEU

Latency (ms) on P100, IWSLT 2016 dev

Gu et al (2018) autoregressive
Gu et al (2018) non-autoregressive
Marian 2018
Marian 2019

Task Definition
Efficiency Results
Latency
Non-autoregressive
Recommendation
Also, in my view, non-autoregressive approaches may or may not be useful in the end, as it has both potentials and limitations. I think it is still a developing area. I am not sure we should limit ourselves by asking all papers to compare with the highly optimized system so far.

Research doesn’t have to be state-of-the-art. Just mention stronger baselines, not 60x weaker straws.
Arguably this is what the shared task explores. Here are some easy things you could have done:

1. Model distillation for autoregressive, since it’s used for non-autoregressive
2. Use 1–2 decoder layers in autoregressive models
3. Averaged attention network
Recommendations
Use sacrebleu.
You can’t compare against a paper that didn’t.
Don’t have to be state-of-the-art. Just cite it or put it in your table. Strawman baselines are misleading.
Lots of baselines

1. Fewer parameters or layers
2. Quantize
3. Prune
4. Beam size
5. Shortlisting
6. Simplify architecture
7. Model distillation
8. Early exit
9. Non-autoregressive

Show your method is a better trade-off via Pareto optimality.

Don’t trust papers that get X speedup for “small” Y BLEU loss!
Currently no evidence that non-autoregressive is competitive.

Kyunghyun Cho
@kchonyc

Replying to @marian_nmt

if anyone ever proves it, it'll be either you or @zngu, and people will come back to this tweet and like it.

6:58 PM · Jun 30, 2020 · Twitter Web App

We’re implementing it in Marian.

WNGT 2020 efficiency task is rolling, send me dockers!