

WNGT 2020 Efficiency Shared Task

Kenneth Heafield,¹ 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.



Stanford NLP Group

@stanfordnlp



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 @lepikhin · 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?

Data Condition

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

$\approx 1s$ to translate \implies too small

Banned a team for memorizing known test set

Test Set

Last year

≈ 1 s to translate \implies too small

Banned a team for memorizing known test set

Before deadline

1 million sentences

≤ 100 space-separated words/sentence

Unspecified test set hidden in input

Test Set

Last year

≈1s to translate \implies too small

Banned a team for memorizing known test set

Before deadline

1 million sentences

≤ 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

<http://data.statmt.org/heffield/wngt20/test/>

Approximately Measuring Quality

Need a surprise evaluation set. WMT20 not ready yet.

→ Uh, average old WMT test sets?

→ WMT12 has sentences longer than 100 words.

→ **WMT1*:** average sacrebleu of WMT11, WMT13–19

See paper supplement for individual WMT scores.


Problem: participants likely tuned on WMT sets.

BLEU?

“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.

Hardware

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.meta1 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!

	GPU	CPU 1 core	CPU all core
NiuTrans	4	0	1
OpenNMT	4	4	4
UEdin	4	2	5

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.

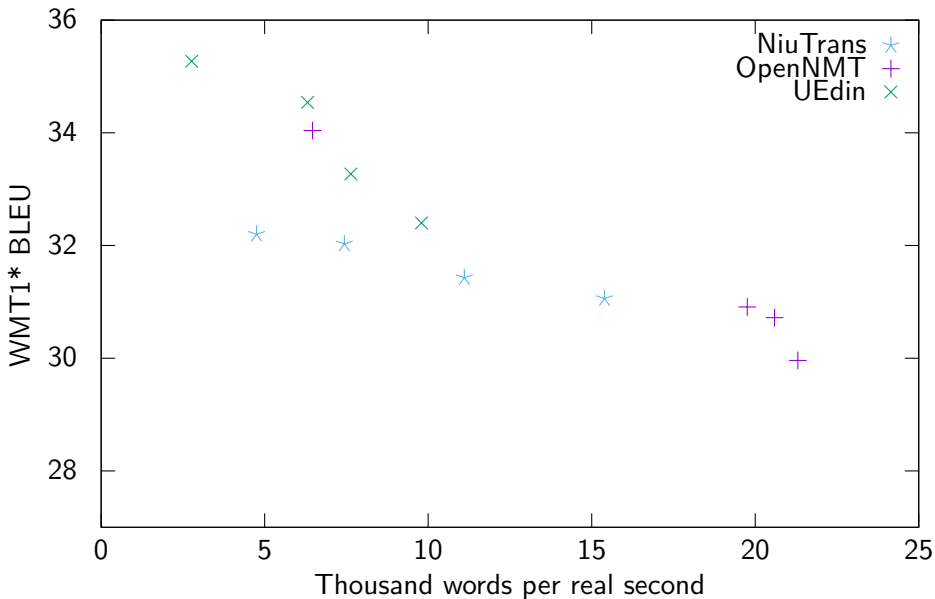
Speed

Primary: wall clock time.

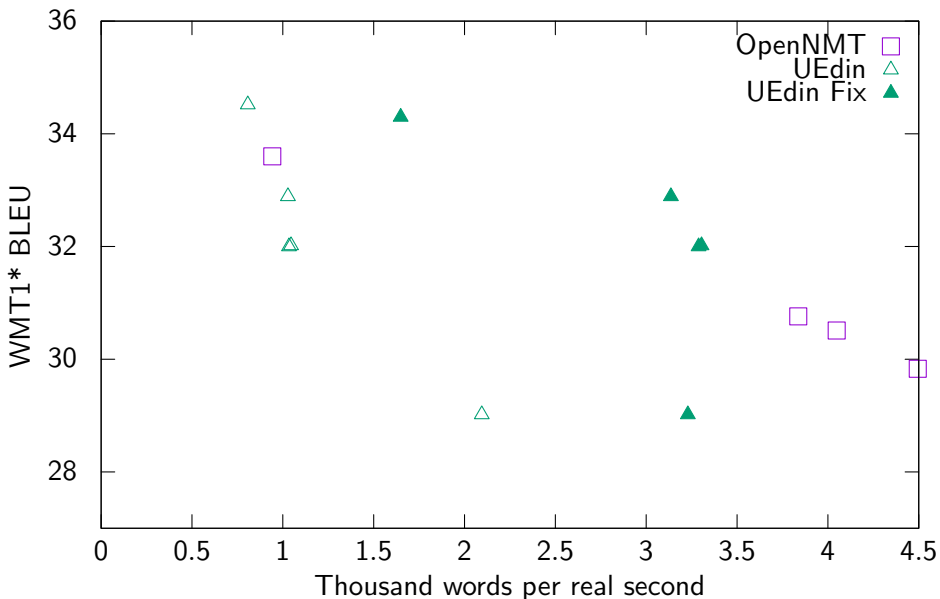
Words per second based on 15,048,961 untokenized words.

Supplementary data: CPU time.

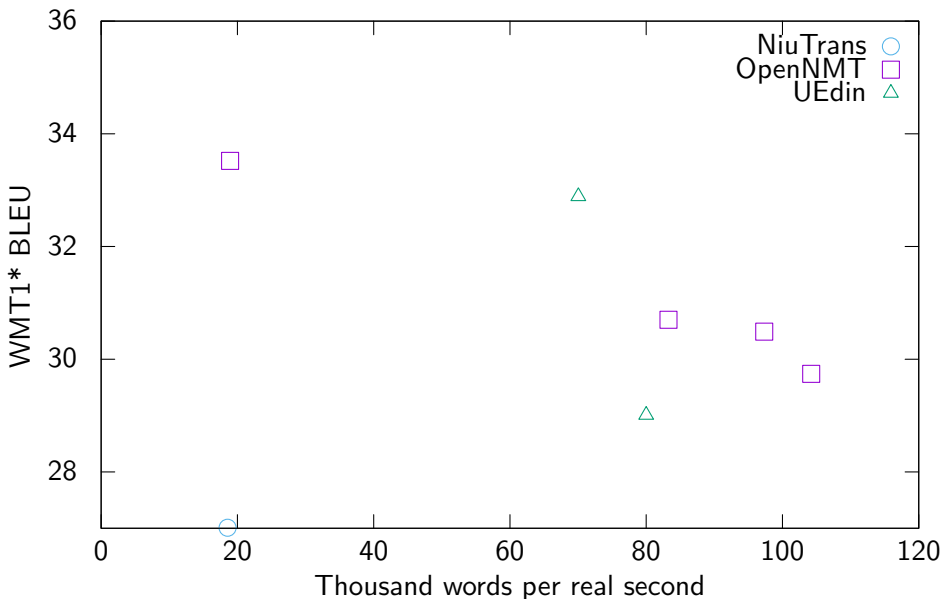
GPU speed



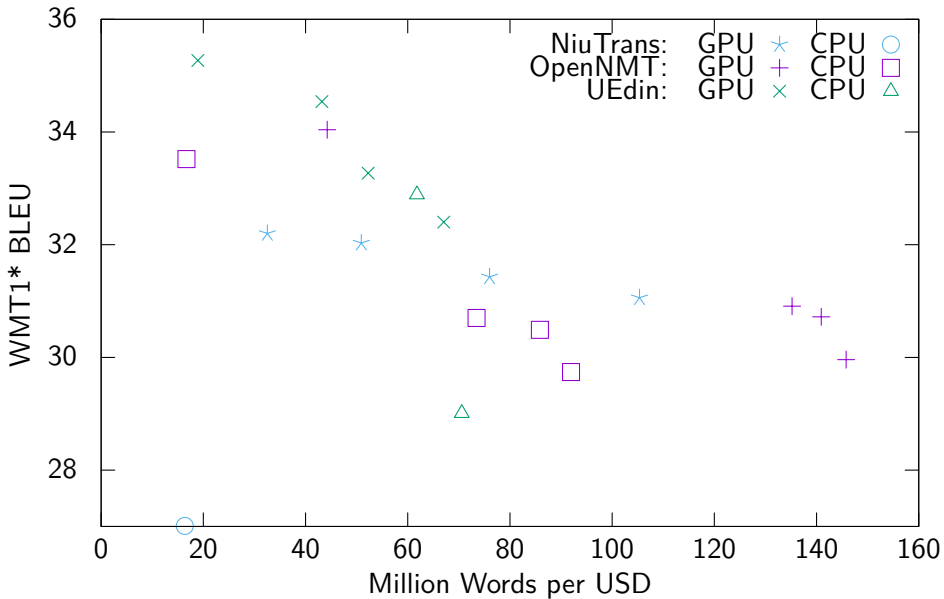
CPU single core speed



CPU all core speed



Cost

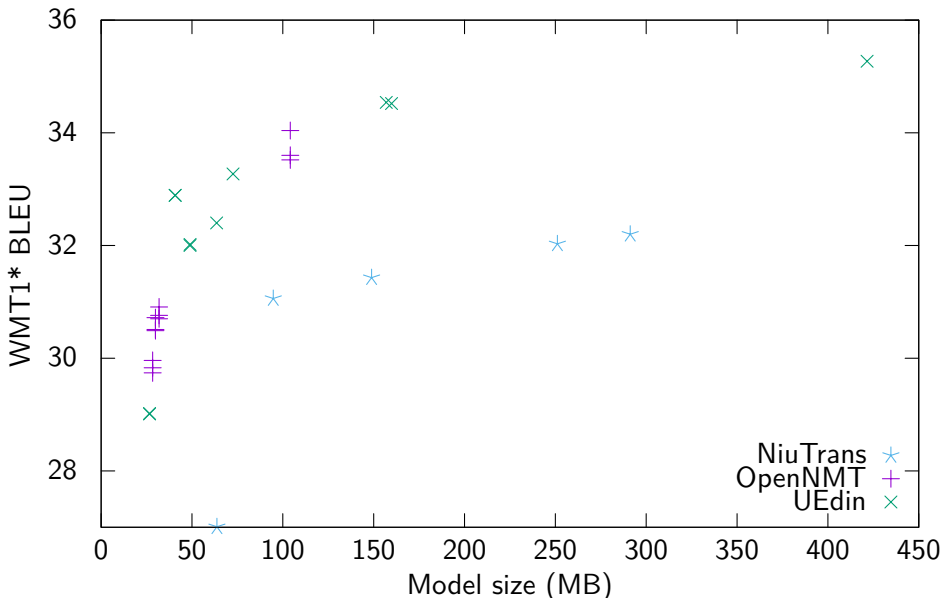


Disk

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

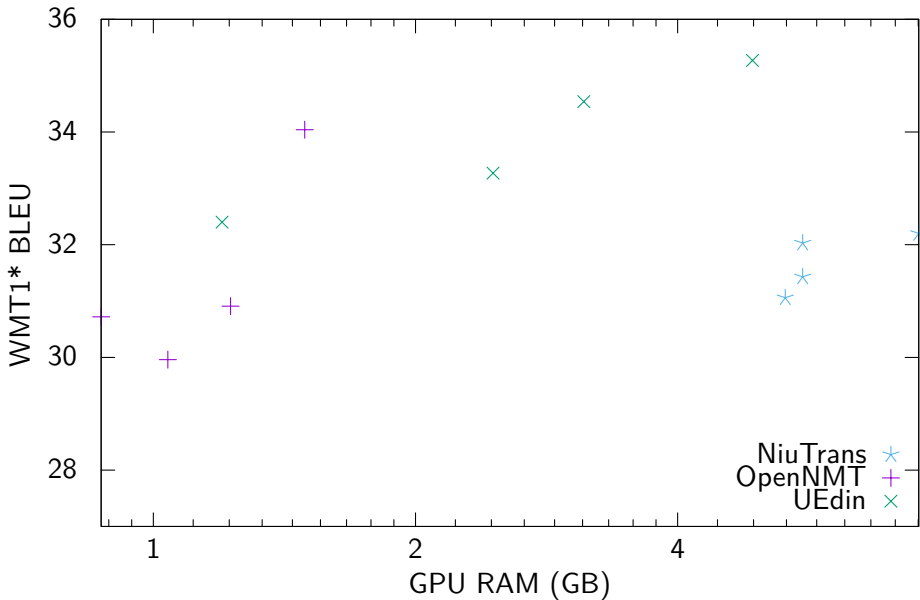


Peak RAM usage

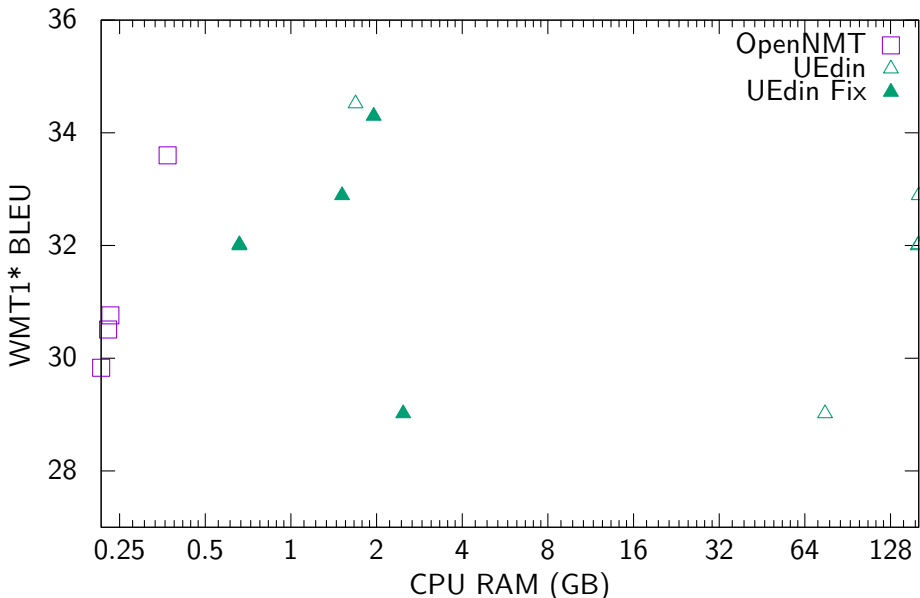
GPU: polling `nvidia-smi`

CPU: `memory.max_usage_in_bytes`

GPU RAM



CPU single core RAM



Task Definition
oooooooooooo

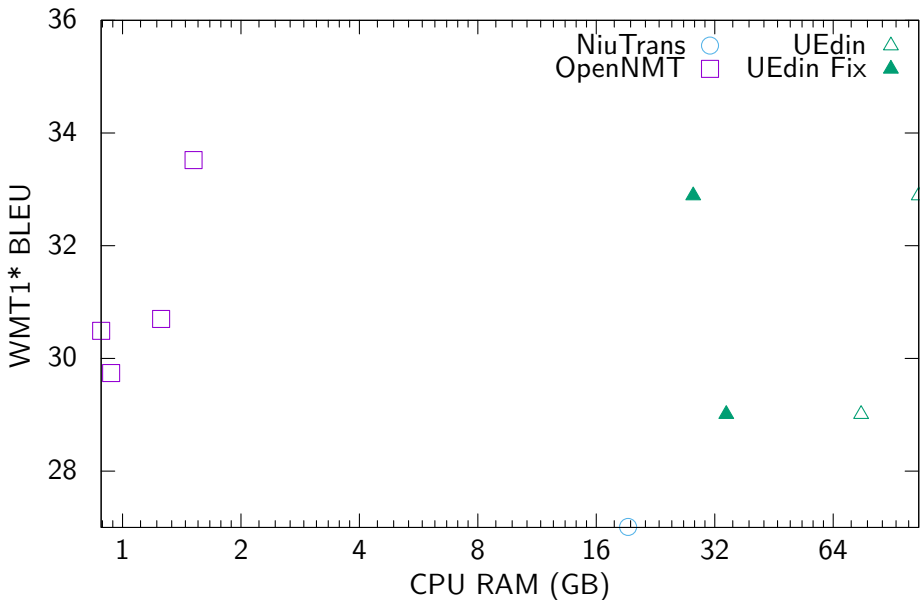
Efficiency Results
oooooooo●oooo

Latency
oooo

Non-autoregressive
ooooo

Recommendation
ooooo

CPU all core RAM



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?

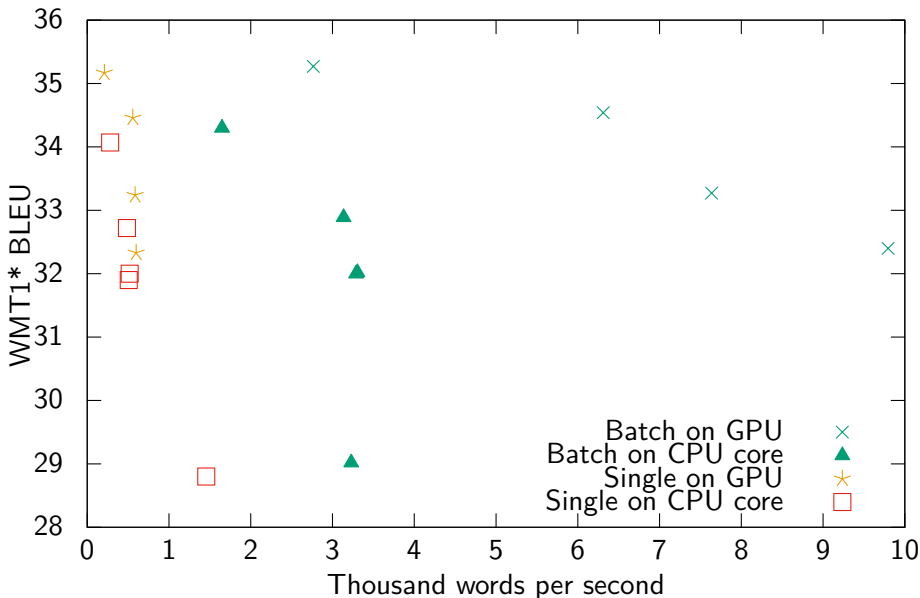
→ Non-autoregressive: a case study in poor evaluation.

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
○○○○○○○○○○

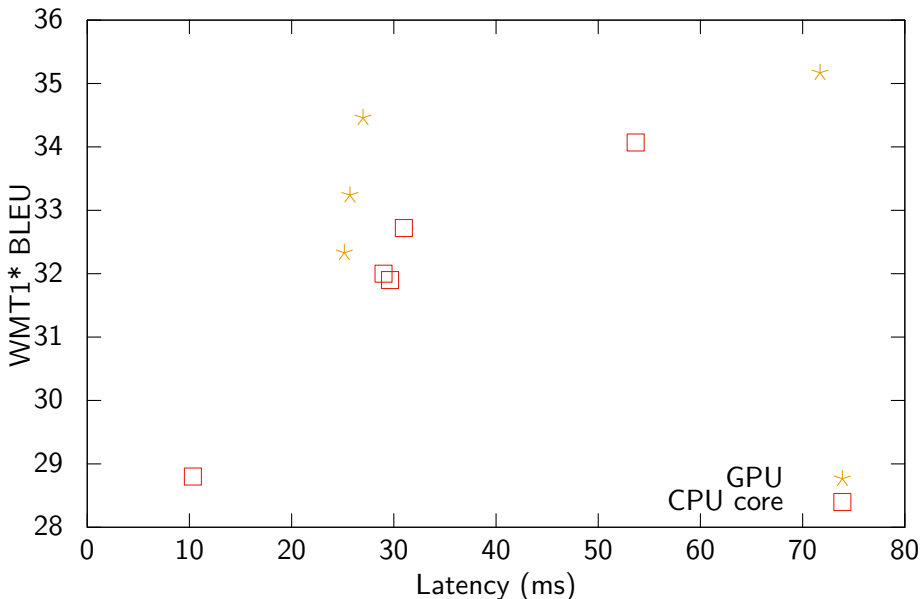
Efficiency Results
○○○○○○○○○○○○

Latency
●○○

Non-autoregressive
○○○○

Recommendation
○○○○

Latency: 10.3–71.7 ms!



Task Definition
○○○○○○○○○○

Efficiency Results
○○○○○○○○○○○○

Latency
○○●○

Non-autoregressive
○○○○○

Recommendation
○○○○○

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)?

Models	WMT14		WMT16		IWSLT16		
	En→De	De→En	En→Ro	Ro→En	En→De	Latency / Speedup	
NAT	17.35	20.62	26.22	27.83	25.20	39 ms	15.6×
NAT (+FT)	17.69	21.47	27.29	29.06	26.52	39 ms	15.6×
NAT (+FT + NPD $s = 10$)	18.66	22.41	29.02	30.76	27.44	79 ms	7.68×
NAT (+FT + NPD $s = 100$)	19.17	23.20	29.79	31.44	28.16	257 ms	2.36×
Autoregressive ($b = 1$)	22.71	26.39	31.35	31.03	28.89	408 ms	1.49×
Autoregressive ($b = 4$)	23.45	27.02	31.91	31.76	29.70	607 ms	1.00×

Replicating Gu et al (2018)'s setup

Do not try this at home or work.

WMT14

State-of-the-art is latest WMT.

Just don't claim state-of-the-art like Wang et al (2018) did.

Replicating Gu et al (2018)'s setup

Do not try this at home or work.

WMT14

State-of-the-art is latest WMT.

Just don't claim state-of-the-art like Wang et al (2018) did.

Tokenized BLEU

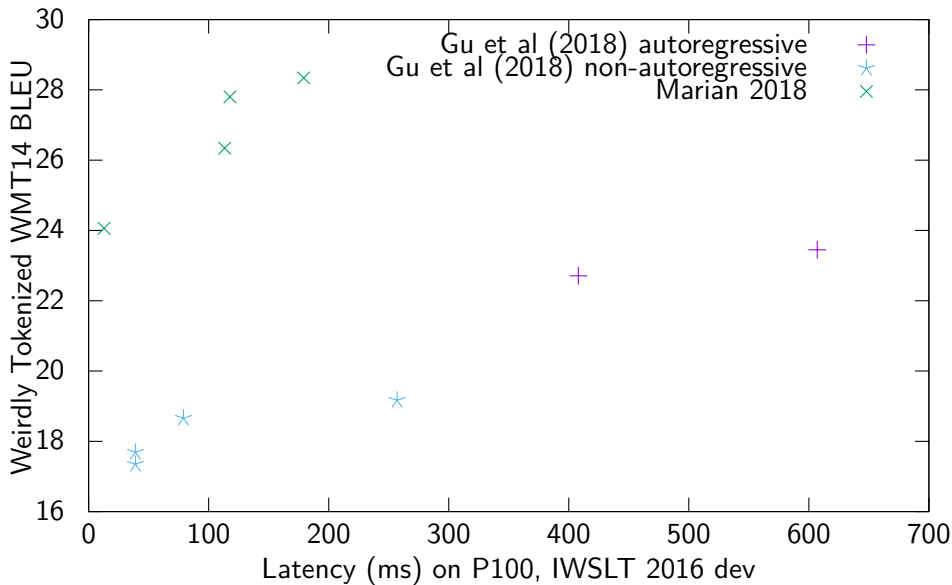
Tokenization differences \implies BLEU scores are not comparable.

But many non-autoregressive papers compare anyway.

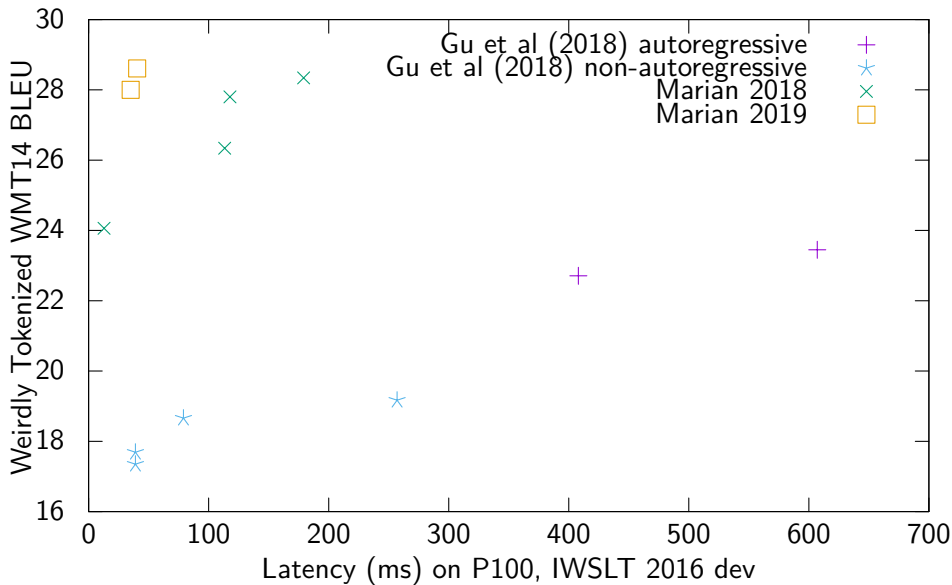
Use sacrebleu instead.

P100, latency on IWSLT 2016 en-de dev.

Real baselines for Gu et al (2018)



Real baselines for Gu et al (2018)





Jiatao Gu
@thoma_gu



Replying to @zngu @odashi_t and 4 others

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.



Jiatao Gu @thoma_gu · Jun 29

Replying to @zngu @raphaelshu and 3 others

I think it really depends on what model (# of parameters, etc) are you using and what language/framework are you running. It is not a fair comparison to show the difference on the absolute values.



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!

Conclusion

Currently no evidence that non-autoregressive is competitive.



We're implementing it in Marian.

WNGT 2020 efficiency task is rolling, send me dockers!