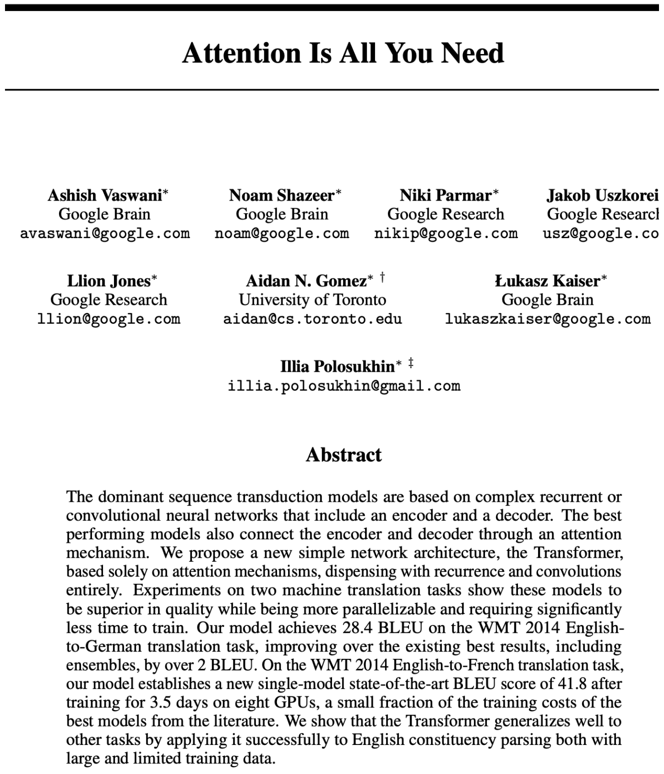
Fairseq En-De Translation Transformer Training on AiMOS vs AiMOSx

Jing Sha

09/2020

Metrics and Start-of-the-Art Performance of Transformer

- Transformer
  - A model architecture relying entirely on an attention mechanism to draw global dependencies between input and output
  - Allows for significant parallelization
- Reference paper:
  - https://arxiv.org/abs/1706.03762
- Implementation:
  - Fairseq: a sequence modeling toolkit written by FAIR in PyTorch that allows training custom models for translation, summarization, language modeling and other text generation tasks.
  - https://fairseq.readthedocs.io/en/latest/getting_started.html#training-a-new-model



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkorei Google Researcl usz@google.co

Llion Jones* Google Research llion@google.com

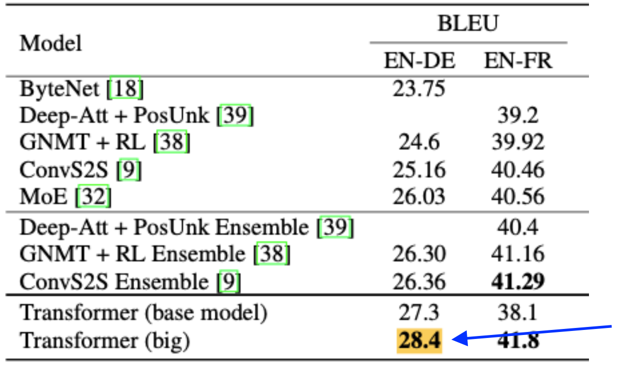
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



Model	BLEU	
	EN-DE	EN-FR
ByteNet [18]	23.75	
Deep-Att + PosUnk [39]		39.2
GNMT + RL [38]	24.6	39.92
ConvS2S [9]	25.16	40.46
MoE [32]	26.03	40.56
Deep-Att + PosUnk Ensemble [39]		40.4
GNMT + RL Ensemble [38]	26.30	41.16
ConvS2S Ensemble [9]	26.36	**41.29**
Transformer (base model)	27.3	38.1
Transformer (big)	**28.4**	**41.8**

8 GPUs, 3.5 days

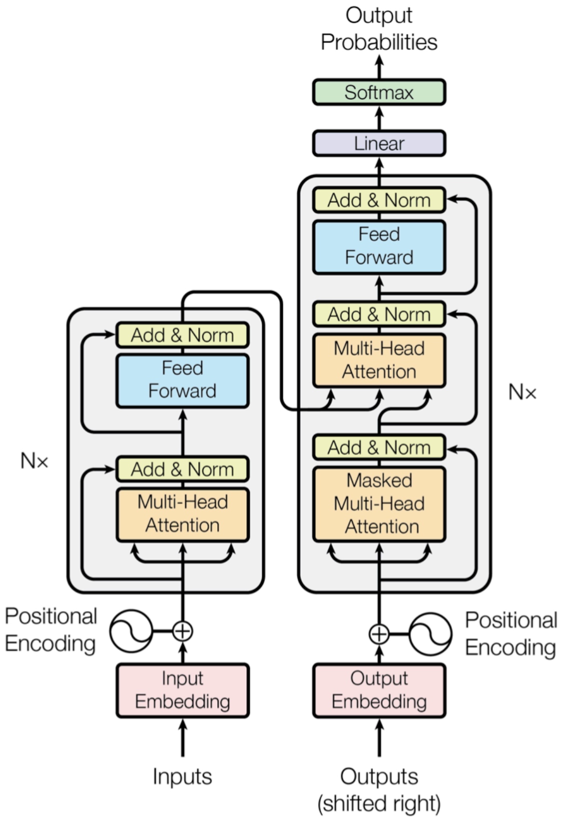


Figure 1: The Transformer - model architecture.

Experiment Setup

- Dataset
  - WMT ’14 en2de
- Hyperparameters
  - Almost same as in “Attention Is All You Need” (NeurIPS 2017)
  - Do not change with number of nodes/GPUs
- Machines
  - AiMOS (power)
  - AiMOSx (x86)
- Nodes
  - 1, 2, 4, 8, 16, 32 (6, 12, 24, 48, 96, 192 GPUs)
- Evaluation
  - Loss (on training/validation data)
  - BLEU score (on test data)

run-fs-transformer.sh

```
# This script runs the slurm batch job with the correct parameters
#!/bin/bash -x
#SBATCH -J fs_32
#SBATCH -o fs_%j.out
#SBATCH -e fs_%j.err
#SBATCH --nodes=32
#SBATCH --gres=gpu:6
#SBATCH --ntasks-per-node=6
#SBATCH --time=06:00:00

# world size = number of gpus
export WZ=192
export LOGDIR="32nodes"
[[ ! -d $LOGDIR ]] && mkdir -p $LOGDIR

export DATA="/gpfs/u/locker/200/CADS/datasets/wmt17_en_de/"

srun --output $LOGDIR/train.log.node%t --error $LOGDIR/train.stderr.node%t.%j fairseq-train $DATA --arch
transformer_vaswani_wmt_en_de_big --optimizer adam --adam-betas '(0.9,0.98)' --clip-norm 0.0 --lr 5e-4 --min-lr 1e-09 --lr-
scheduler inverse_sqrt --warmup-init-lr 1e-07 --warmup-updates 4000 --dropout 0.3 --criterion label_smoothed_cross_entropy --
max-tokens 2048 --fp16 --lazy-load --update-freq 4 --keep-interval-updates 100 --save-interval-updates 3000 --log-interval 50 --
tensorboard-logdir $LOGDIR/logs --save-dir $LOGDIR/checkpoints --distributed-world-size $WZ --distributed-port 9218
```

Environment and Run Setup

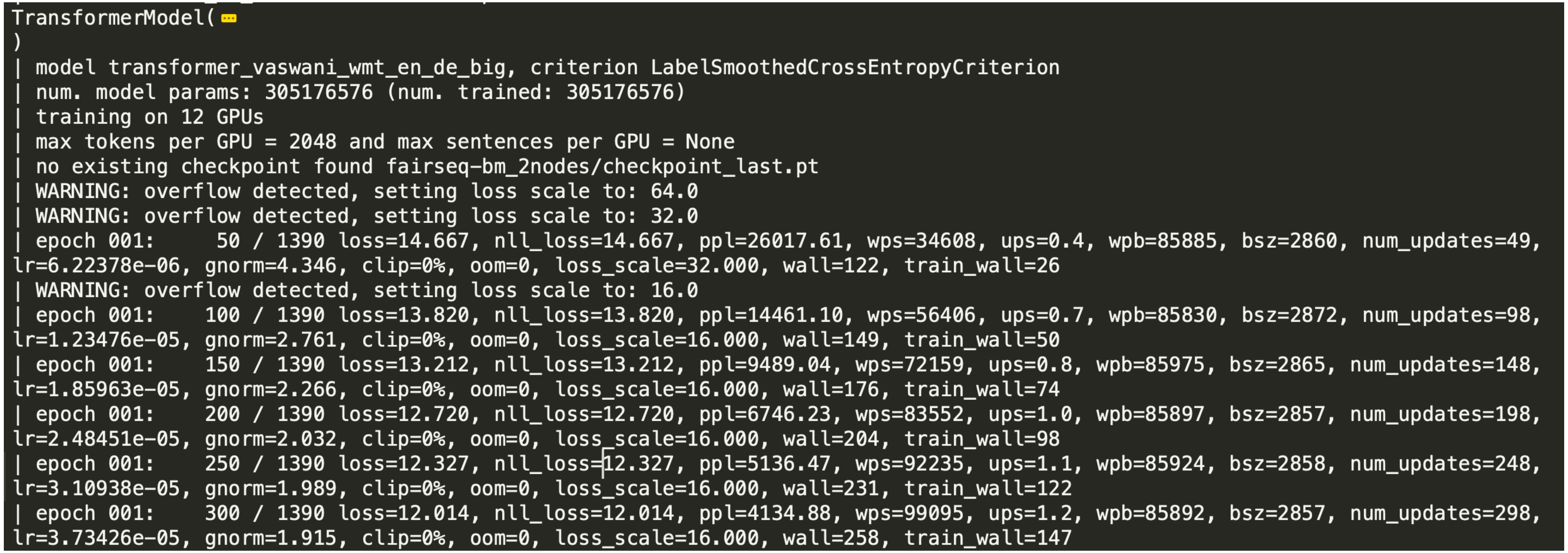
1. ssh into one of the frontend nodes, *e.g.* dcsfen01 (AiMOS) and nplfen01 (AiMOSx)

2. Set up a Conda environment (do once)
   - Install Conda and set up `~/.condarc`
     - For how to see https://ibm-ai-hardware-center.github.io/AiMOS/docs/aimos-workload.html#install-anaconda
   - `conda create -n poweraifs python=3.7`
   - `conda activate poweraifs`
   - `conda install fairseq`
   - `pip install tensorboardx` (for exporting events files for tensorboard)

3. Run script
   - Copy the run-fs-transformer.sh to `~/scratch-shared/fairseq`
   - Update environment variables and number of GPUs in `run-fs-transformer.sh`
   - Submit a distributed learning job: `sbatch run-fs-transformer.sh`

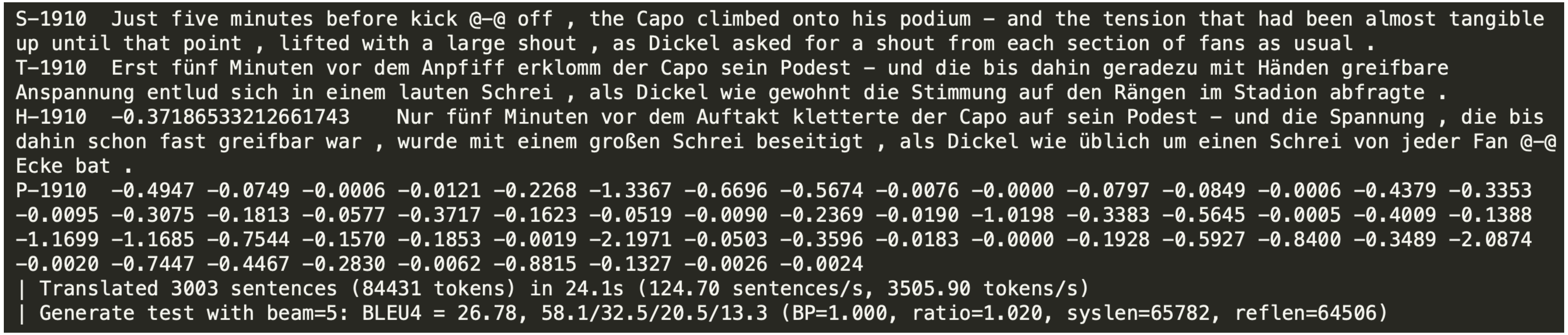
Sample Experiment Output

Training log:



```
TransformerModel(⋯
)
| model transformer_vaswani_wmt_en_de_big, criterion LabelSmoothedCrossEntropyCriterion
| num. model params: 305176576 (num. trained: 305176576)
| training on 12 GPUs
| max tokens per GPU = 2048 and max sentences per GPU = None
| no existing checkpoint found fairseq-bm_2nodes/checkpoint_last.pt
| WARNING: overflow detected, setting loss scale to: 64.0
| WARNING: overflow detected, setting loss scale to: 32.0
| epoch 001:     50 / 1390 loss=14.667, nll_loss=14.667, ppl=26017.61, wps=34608, ups=0.4, wpb=85885, bsz=2860, num_updates=49,
lr=6.22378e-06, gnorm=4.346, clip=0%, oom=0, loss_scale=32.000, wall=122, train_wall=26
| WARNING: overflow detected, setting loss scale to: 16.0
| epoch 001:    100 / 1390 loss=13.820, nll_loss=13.820, ppl=14461.10, wps=56406, ups=0.7, wpb=85830, bsz=2872, num_updates=98,
lr=1.23476e-05, gnorm=2.761, clip=0%, oom=0, loss_scale=16.000, wall=149, train_wall=50
| epoch 001:    150 / 1390 loss=13.212, nll_loss=13.212, ppl=9489.04, wps=72159, ups=0.8, wpb=85975, bsz=2865, num_updates=148,
lr=1.85963e-05, gnorm=2.266, clip=0%, oom=0, loss_scale=16.000, wall=176, train_wall=74
| epoch 001:    200 / 1390 loss=12.720, nll_loss=12.720, ppl=6746.23, wps=83552, ups=1.0, wpb=85897, bsz=2857, num_updates=198,
lr=2.48451e-05, gnorm=2.032, clip=0%, oom=0, loss_scale=16.000, wall=204, train_wall=98
| epoch 001:    250 / 1390 loss=12.327, nll_loss=12.327, ppl=5136.47, wps=92235, ups=1.1, wpb=85924, bsz=2858, num_updates=248,
lr=3.10938e-05, gnorm=1.989, clip=0%, oom=0, loss_scale=16.000, wall=231, train_wall=122
| epoch 001:    300 / 1390 loss=12.014, nll_loss=12.014, ppl=4134.88, wps=99095, ups=1.2, wpb=85892, bsz=2857, num_updates=298,
lr=3.73426e-05, gnorm=1.915, clip=0%, oom=0, loss_scale=16.000, wall=258, train_wall=147
```

Evaluation log:



```
S-1910  Just five minutes before kick @-@ off , the Capo climbed onto his podium - and the tension that had been almost tangible
up until that point , lifted with a large shout , as Dickel asked for a shout from each section of fans as usual .
T-1910  Erst fünf Minuten vor dem Anpfiff erklomm der Capo sein Podest - und die bis dahin geradezu mit Händen greifbare
Anspannung entlud sich in einem lauten Schrei , als Dickel wie gewohnt die Stimmung auf den Rängen im Stadion abfragte .
H-1910  -0.37186533212661743    Nur fünf Minuten vor dem Auftakt kletterte der Capo auf sein Podest - und die Spannung , die bis
dahin schon fast greifbar war , wurde mit einem großen Schrei beseitigt , als Dickel wie üblich um einen Schrei von jeder Fan @-@
Ecke bat .
P-1910  -0.4947 -0.0749 -0.0006 -0.0121 -0.2268 -1.3367 -0.6696 -0.5674 -0.0076 -0.0000 -0.0797 -0.0849 -0.0006 -0.4379 -0.3353
-0.0095 -0.3075 -0.1813 -0.0577 -0.3717 -0.1623 -0.0519 -0.0090 -0.2369 -0.0190 -1.0198 -0.3383 -0.5645 -0.0005 -0.4009 -0.1388
-1.1699 -1.1685 -0.7544 -0.1570 -0.1853 -0.0019 -2.1971 -0.0503 -0.3596 -0.0183 -0.0000 -0.1928 -0.5927 -0.8400 -0.3489 -2.0874
-0.0020 -0.7447 -0.4467 -0.2830 -0.0062 -0.8815 -0.1327 -0.0026 -0.0024
| Translated 3003 sentences (84431 tokens) in 24.1s (124.70 sentences/s, 3505.90 tokens/s)
| Generate test with beam=5: BLEU4 = 26.78, 58.1/32.5/20.5/13.3 (BP=1.000, ratio=1.020, syslen=65782, reflen=64506)
```