Sustainable Development of GenAI in Japan Through Continual Pre-training

Institute of Science Tokyo
Rio Yokota

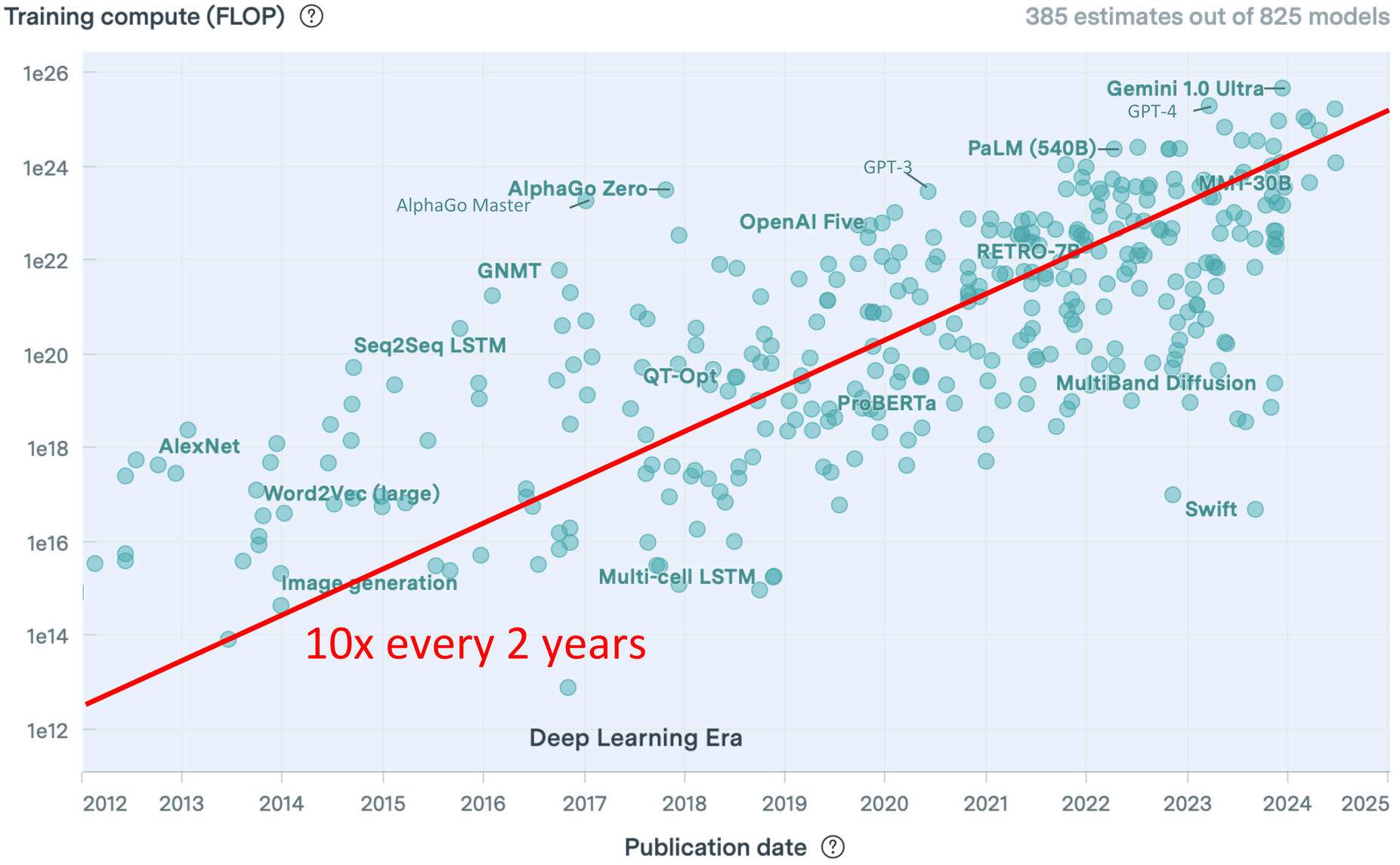
Generative AI: Pathways to Democratization,
Transparency and Sustainability
Nov 13, 2024

The ~~Scaling~~ Spending Law

2012: AlexNet
$500 GPU x 2
= $1K
5 days of training

2022: GPT-4 (prediction)
$8K GPU x 25,000
= $200M
90 days of training

Any growth beyond Moore's law is achieved only by spending more
2024: $1B
2026: $5B
2028: $25B
2030: $125B



https://epochai.org/data/notable-ai-models

Japanese Investment in LLMs

January 19, 2024

AWS plans to invest 2.26 trillion yen into its Japanese cloud infrastructure by 2027

AI IMPACT

SoftBank will reportedly invest nearly $1 billion in AI push, tapping Nvidia's chips

PUBLISHED TUE, APR 23 2024•3:35 AM EDT | UPDATED TUE, APR 23 2024•2:58 PM EDT

Microsoft to invest US$2.9 billion in AI and cloud infrastructure in Japan while boosting the nation's skills, research and cybersecurity

April 10, 2024 | Microsoft Source

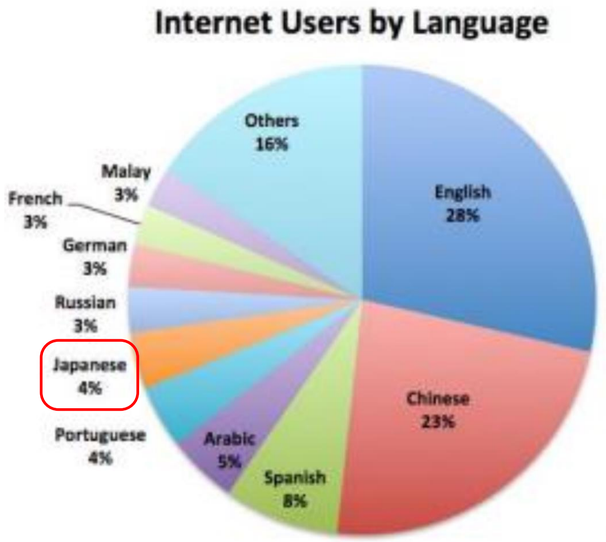
Sakana AI raising ~$100M at a $1B valuation led by NEA, Khosla, and Lux.

Morphic Research June 15, 2024 at 9:15 AM

April 14, 2024

Introducing OpenAI Japan

We are excited to announce our first office in Asia and we're releasing a GPT-4 custom model optimized for the Japanese language.



World Internet Users Statistics, 2022

Japanese LLMs

LLM-jp

Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (？,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

Members:
Tokyo Tech., AIST

System:
ABCI (350,000 A100 hours)

Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

LLM-JP

Japanese corpora:
Wikipedia: 1.4B tokens (1.3M documents)
mC4: 136B tokens (75M documents)
Common Crawl: 380B tokens (300M documents)
NDL WARP: 250B tokens (160M URL → 50M PDF → 39M documents)
JST J-STAGE: 3B tokens

English corpora:
Wikipedia: 5.1B tokens
Pile: 176B tokens
Stack: 148B tokens
SlimPajama: 627B tokens
RefinedWeb: 600B tokens
Dolma: 3T tokens
FineWeb: 15T tokens

Groups:
1. Data (crawling)
2. Data (cleaning)
3. Data (papers/books)
4. Architectures
5. Pre-training
6. Instruct/Fine-tuning
7. Evaluation
8. Safety

Computational resource:
MDX (600K A100 hours)
ABCI (900K A100 hours)
GCP (？M H100 hours)
TSUBAME4.0 (720K H100 hours)

Universities:
The University of Tokyo (Imaizumi, Ozeki, Kawahara, Tsuruoka, Baba, Matsuo, Miyao, Yanaka, Yoshinaga, Hanaoka, Kawazoe, Kodera, Taura), Tohoku University (Inui, Suzuki, Sakaguchi), Tokyo Institute of Technology (Okazaki, Arase, Yokota, Endo, Okumura), Waseda University (Kawahara), Ochanomizu University (Kobayashi), Nagoya University (Takeda, Sasano), Kyoto University (Kurohashi), Osaka University (Onizuka), Hokkaido University (Rafal), Tsukuba (Ochiai), Ochanomizu University (Kobayashi), Sophia University (Fukazawa), UEC (Yanai), Hitotsubashi University (Keyaki), Tokyo Metropolitan University (Hirasawa), Musashino University (Watanabe), Keio University (Ohara), Nara Institute of Science and Technology (Aramaki, Watanabe), Kyushu Institute of Technology (Okita), OIST (Yamada)

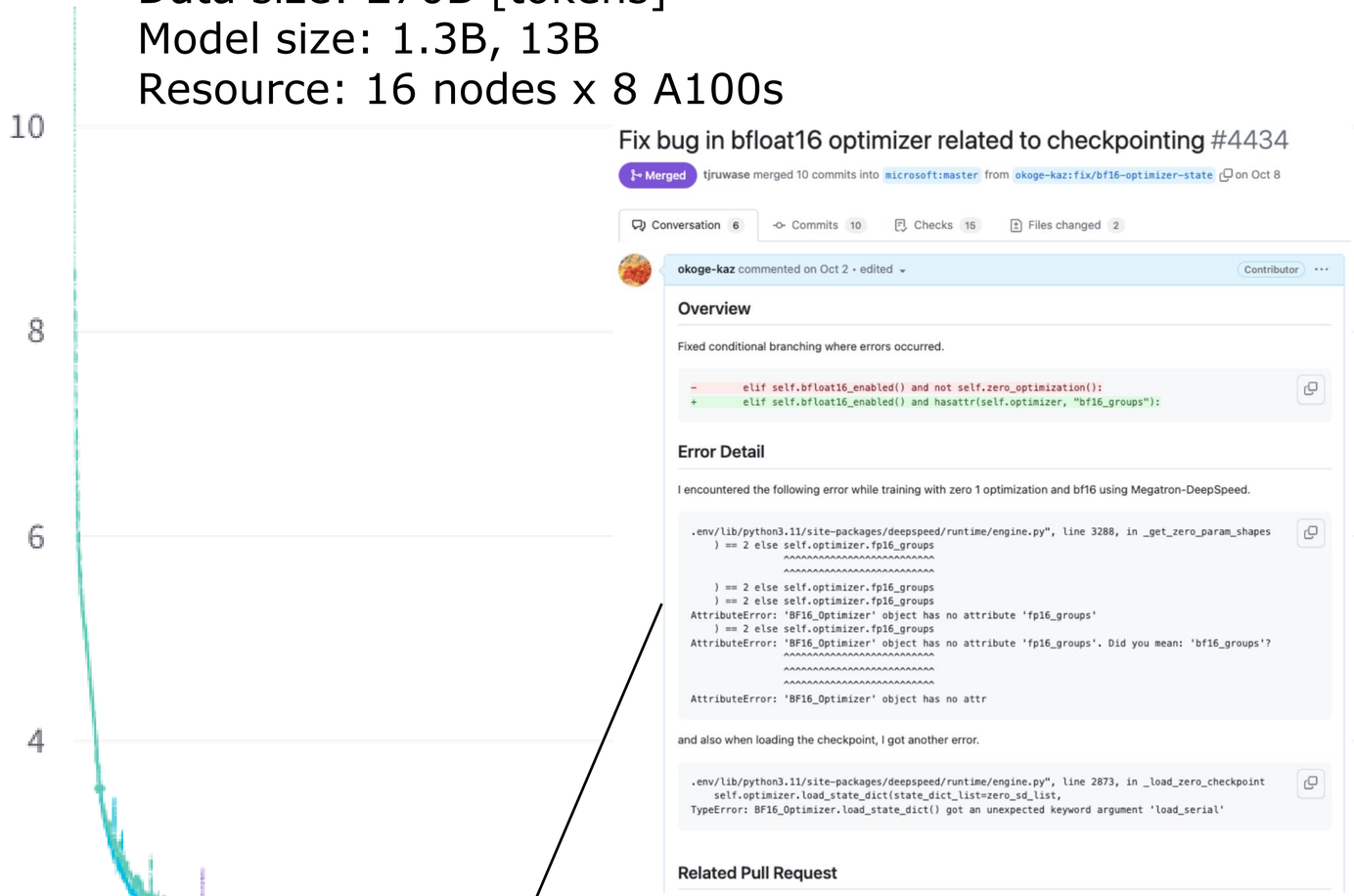
National Laboratories:
RIKEN AIP, RIKEN CCS, RIKEN GRP, AIST, NII, NICT, JST, JAMSTEC

Industry:
Microsoft Japan, AWS Japan, NVIDIA Japan, Intel, IBM Japan, Sakana AI, Stability AI Japan, SB Institutions, LINE/Yahoo, Sony, DeNA, Toshiba, Fujitsu, NTT, NTT Communications, KDDI, Toyota, Turing, Preferred Networks, Cyberagent, ELYZA, OmronScinicX, Studio Ousia, Precision, ZENKIGEN, Legalscape, Miraihonyaku, Megagon Labs, Stockmark, Matsuri Technologies, First Accounting, Baobab, Polaris.ai, Money Forward, Mercari, Asteras, Pasco, Rakuten, Lightblue, GMO, Advance Soft, Laboro.AI, Algomatic, Brainpad, IHI, Mizuho Bank, Retrieva, Fixstars, neoAI, and many more.

LLM-jp 1.3B and 13B models

Data size: 270B [tokens]
Model size: 1.3B, 13B
Resource: 16 nodes x 8 A100s

	1.3B	13B
lr (min_lr)	2.0e-4 (1.0e-6)	1.0e-4 (1.0e-6)
init_std	0.013	0.008
num_layers	24	40
hidden_size	2048	5120
num_attn_heads	16	40
lr_warmup	2000	2000
lr_decay	cosine	cosine
lr_decay_tokens	1080B	1080B
Optimizer (decay)	Adam (0.1)	Adam (0.1)
Adam beta1, beta2	0.9, 0.95	0.9, 0.95
batch_size	1536	1536
clip_grads	1.0	1.0
use_flassh_attn	yes	yes
zero stage	1	1
Seq_len	2048	2048



Fix bug in bfloat16 optimizer related to checkpointing #4434

Merged tjruwase merged 10 commits into microsoft:master from okoge-kaz:fix/bf16-optimizer-state on Oct 8

Conversation 6 Commits 10 Checks 15 Files changed 2

okoge-kaz commented on Oct 2 • edited Contributor

Overview

Fixed conditional branching where errors occurred.

```
-       elif self.bfloat16_enabled() and not self.zero_optimization():
+       elif self.bfloat16_enabled() and hasattr(self.optimizer, "bf16_groups"):
```

Error Detail

I encountered the following error while training with zero 1 optimization and bf16 using Megatron-DeepSpeed.

```
.env/lib/python3.11/site-packages/deepspeed/runtime/engine.py", line 3288, in _get_zero_param_shapes
    ) == 2 else self.optimizer.fp16_groups
                 ^^^^^^^^^^^^^^^^^^^^^^^^^
                 ^^^^^^^^^^^^^^^^^^^^^^^^^
    ) == 2 else self.optimizer.fp16_groups
    ) == 2 else self.optimizer.fp16_groups
AttributeError: 'BF16_Optimizer' object has no attribute 'fp16_groups'
    ) == 2 else self.optimizer.fp16_groups
AttributeError: 'BF16_Optimizer' object has no attribute 'fp16_groups'. Did you mean: 'bf16_groups'?
                 ^^^^^^^^^^^^^^^^^^^^^^^^^
                 ^^^^^^^^^^^^^^^^^^^^^^^^^
                 ^^^^^^^^^^^^^^^^^^^^^^^^^
AttributeError: 'BF16_Optimizer' object has no attr
```

and also when loading the checkpoint, I got another error.

```
.env/lib/python3.11/site-packages/deepspeed/runtime/engine.py", line 2873, in _load_zero_checkpoint
    self.optimizer.load_state_dict(state_dict_list=zero_sd_list,
TypeError: BF16_Optimizer.load_state_dict() got an unexpected keyword argument 'load_serial'
```

Related Pull Request

Train Loss

LLM-jp 175B model

Data size: Stopped at 70B [tokens]
Model size: 175B (GPT)
Resource: 49 nodes x 8 A100s → 42,300 [token/sec]

LLM-jp 172B model

Data size: EN 950B, JP 634B, code 114B [tokens]
Model size: 172B (Llama2)
Resource: ?? nodes x 8 H100s → 260,000 token/sec

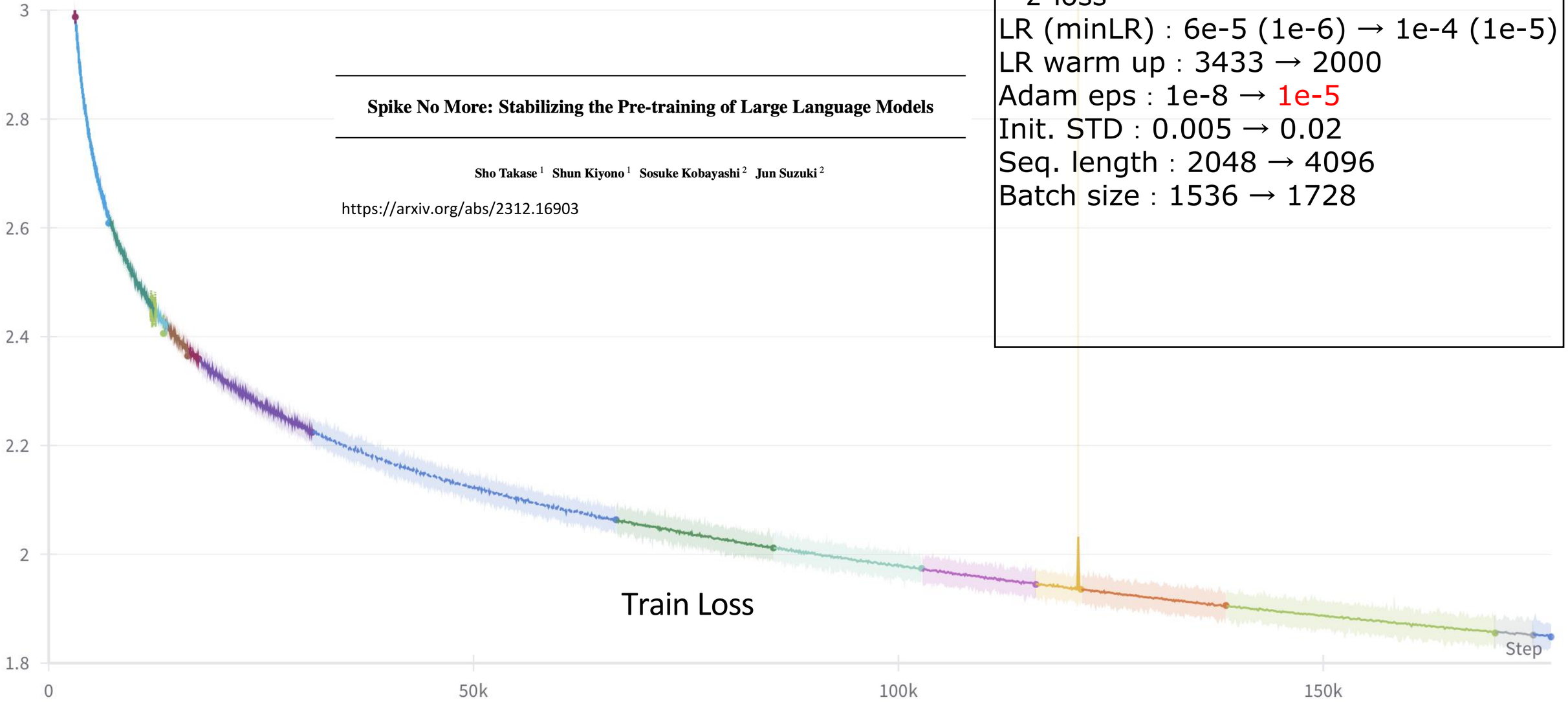
Things we changed:
GPT → Llama2
- pre-norm
- RMS norm
- scaled embedding
- z-loss

LR (minLR) : 6e-5 (1e-6) → 1e-4 (1e-5)
LR warm up : 3433 → 2000
Adam eps : 1e-8 → 1e-5
Init. STD : 0.005 → 0.02
Seq. length : 2048 → 4096
Batch size : 1536 → 1728

Spike No More: Stabilizing the Pre-training of Large Language Models

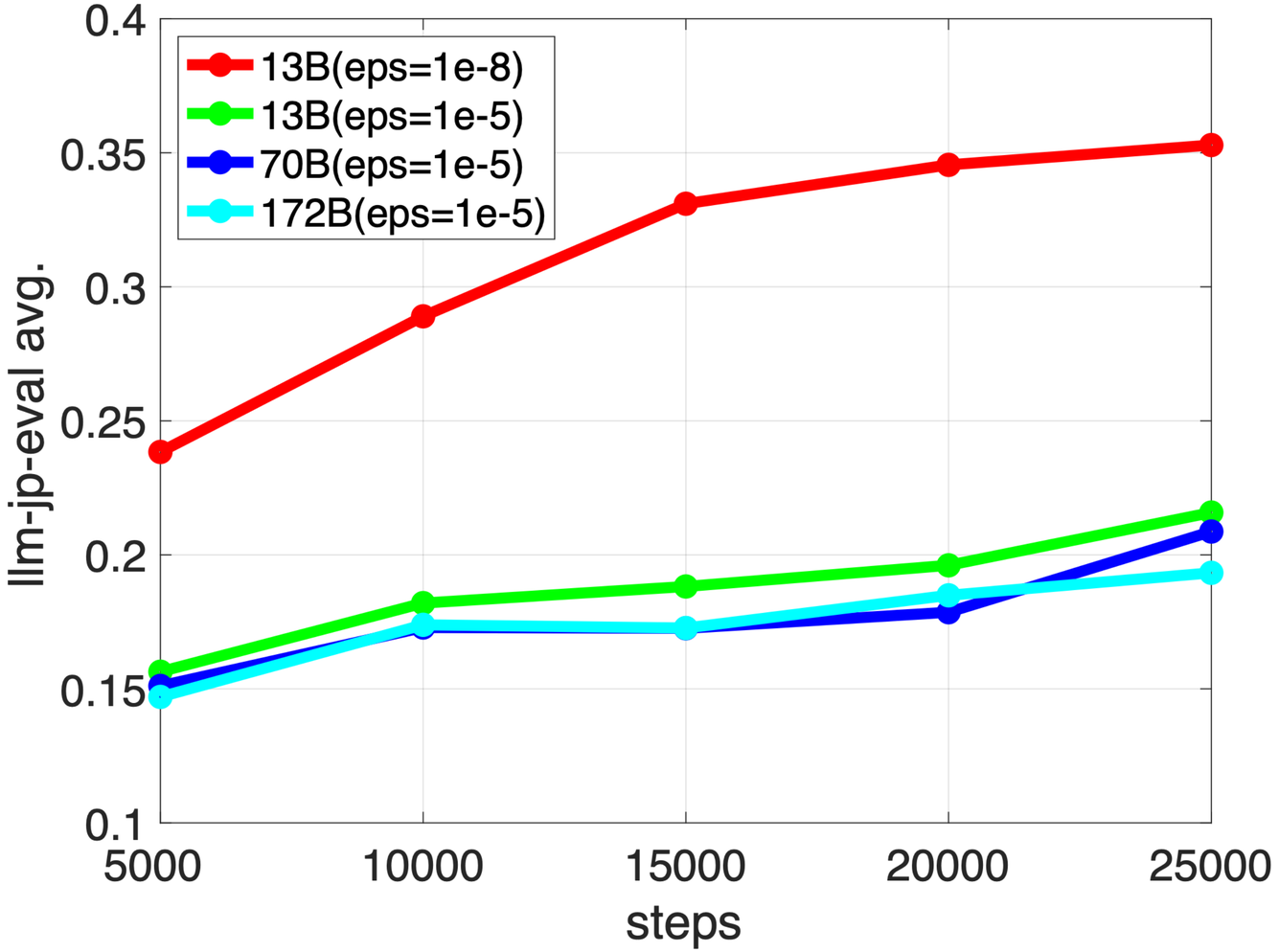
Sho Takase [1] Shun Kiyono [1] Sosuke Kobayashi [2] Jun Suzuki [2]

https://arxiv.org/abs/2312.16903



Train Loss

Adam eps=1e-5?



Things we changed:
GPT → Llama2
- pre-norm
- RMS norm
- scaled embedding
- z-loss

LR (minLR) : 6e-5 (1e-6) → 1e-4 (1e-5)
LR warm up : 3433 → 2000
Adam eps : 1e-8 → 1e-5
Init. STD : 0.005 → 0.02
Seq. length : 2048 → 4096
Batch size : 1536 → 1728

Hyperparameters. We trained using the AdamW optimizer (Loshchilov and Hutter, 2017), with $\beta_1 = 0.9, \beta_2 = 0.95, \text{eps} = 10^{-5}$. We use a cosine learning rate schedule, with warmup of 2000 steps, and decay final learning rate down to 10% of the peak learning rate. We use a weight decay of 0.1 and gradient clipping of 1.0. Figure 5 (a) shows the training loss for LLAMA 2 with these hyperparameters.

Japanese LLMs

LLM-jp

Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (？,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

Members:
Tokyo Tech., AIST

System:
ABCI (350,000 A100 hours)

Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

Continual Pre-training vs Pre-training from Scratch

Continual Pre-training

Advantages

- Leverages all the training data used to train the original model

Disadvantages

- Unclear what data the model was trained on

Pre-training from Scratch

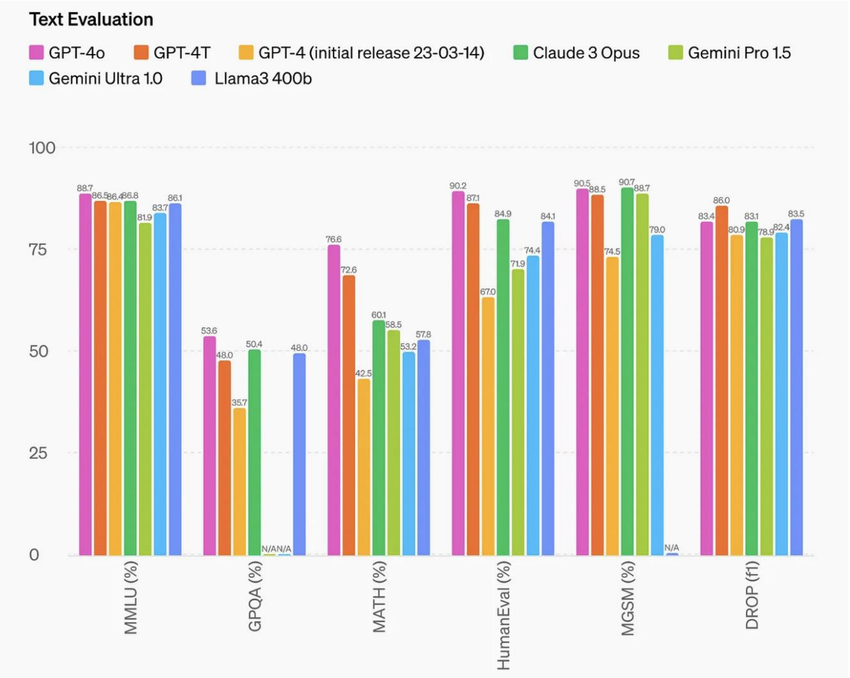
Advantages

- Total control over what data the model is trained on

Disadvantages

- Need enormous data and computer resources

https://medium.com/@lars.chr.wiik/gpt-4o-vs-gpt-4-vs-gemini-1-5-performance-analysis-6bd207a2c580



Open-source models will always be trailing not so far behind closed models

How to leverage these models and adapt them to novel languages / modalities is something worth investigating

Llama3-400B is pretty competitive with GPT-4 on MMLU and DROP

Qwen2-72B is even better than Llama3-70B

	Qwen2-72B	Llama3-70B	Mixtral-8x22B
MMLU	**84.2**	79.5	77.8
MMLU-Pro	**55.6**	52.8	49.5
GPQA	**37.9**	36.3	34.3
TheoremQA	**43.1**	32.3	35.9
BBH	**82.4**	81.0	78.9
HumanEval	**64.6**	48.2	46.3
MBPP	**76.9**	70.4	71.7
MultiPL-E	**59.6**	46.3	46.7
GSM8K	**89.5**	83.0	83.7
MATH	**51.1**	42.5	41.7
C-Eval	**91.0**	65.2	54.6
CMMLU	**90.1**	67.2	53.4
Multi-Exam	**76.6**	70.0	63.5
Multi-Understanding	**80.7**	79.9	77.7
Multi-Mathematics	**76.0**	67.1	62.9

https://www.reddit.com/r/singularity/comments/1d9mi13/new_alibabas_llm_qwen_2_72b_surpasses_llama_3_70b/

Continual Training on Japanese Vocabulary

English
Characters not in the vocabulary are broken down into UTF-8 bytes, consuming as many as three tokens per character

Tokens 22 **Characters** 114

```
Characters not in the vocabulary are broken down into UTF-8 bytes,
 consuming as many as three tokens per character
```

語彙に含まれない文字はUTF-8のバイト列に分解され1文字が3トークン程度も消費することとなる

Tokens 60 **Characters** 47

```
����に��まれない����はUTF-8のバイト��に�����され1����が3トークン�����も������することとなる
```

Language	Tokens
English	1x
Japanese	3x
Chinese	3x
Korean	5x

Data filtering

Step 1	**Downloading WARC files**	63,352,266,406 pages in Common Crawl
Step 2	**Rapid Japanese detection**	This step reduces processing time for Steps 3 and 4
Step 3	**Text extraction**	Extract text from HTML (Trafilatura)
Step 4	**Precise Japanese detection**	2,686,080,919 Japanese pages extracted
Step 5	**Quality filtering**	Find high-quality text based on several rules
Step 6	**Deduplication**	Remove duplicated text (to avoid overfitting)
Step 7	**Filtering by hostnames**	Remove pages that may be unuseful to LLMs
Step 8	**Normalizing punctuations**	Normalize Japanese punctuations into “、” and “。”
Step 9	**Removing footersd**	Remove footers that were left at Step 3

Effect of vocab. extension

English: SQuAD2, TriviaQA, OpenBookQA, xwinograd, HellaSwag, GSM8K
Japanese: MC, QA, RC, mgsm_ja, XLSUM_ja, WMT20-ja-en, WMT20-en-ja

					Japanese	Eng
Meta	Llama-2-7b-hf	7B	N/A	No	0.3111	**0.4883**
Cyberagent	calm2-7b	7B	N/A	No	0.2819	0.4026
Stability AI Japan	japanese-stablelm-base-beta-7b	7B	No	Yes (Llama2)	0.321	0.4736
Stability AI Japan	japanese-stablelm-base-gamma-7b	7B	No	Yes (mistral)	**0.4186**	0.486
Stability AI Japan	japanese-stablelm-base-ja_vocab-beta-7b	7B	Yes	Yes (Llama2)	0.2729	0.4545
Tokyo Tech + AIST	Swallow (w/o vocab ext.)	7B	No	Yes (Llama2)	0.3847	0.4385
Tokyo Tech + AIST	Swallow (w vocab ext.)	7B	Yes	Yes (Llama2)	0.3714	0.4399
Meta	Llama-2-13b-hf	13B	N/A	No	0.3893	**0.5381**
LLM-JP	llm-jp-13b-v1.0	13B	N/A	No	0.2684	0.3893
Stockmark	stockmark-13b	13B	N/A	No	0.2427	0.2881
Tokyo Tech + AIST	Swallow (w vocab ext.)	13B	Yes	Yes (Llama2)	**0.4467**	0.4922
Meta	Llama-2-70b-hf	70B	N/A	No	0.4862	**0.6267**
Stability AI Japan	japanese-stablelm-base-beta-70b	70B	No	Yes (Llama2)	0.5089	N/A
Tokyo Tech + AIST	Swallow (w/o vocab ext.)	70B	No	Yes (Llama2)	**0.5414**	N/A
Tokyo Tech + AIST	Swallow (w vocab ext.)	70B	Yes	Yes (Llama2)	0.5371	N/A

Japanese LLM benchmark (by Weights & Biases)

	model_name	GLP	ALT	Total AVG
15	anthropic.claude-3-5-sonnet-20240620-	0.7618	0.9027	0.8322
9	gpt-4o-2024-05-13	0.7451	0.8796	0.8123
38	gpt-4-turbo-2024-04-09	0.7019	0.8467	0.7743
16	anthropic.claude-3-opus-20240229-v1:0	0.7262	0.82	0.7731
18	gemini-1.5-pro-001	0.6882	0.8199	0.754
1	gpt-4o-mini-2024-07-18	0.7046	0.7788	0.7417
39	gpt-4-0613	0.6762	0.7973	0.7367
13	gemini-1.5-flash-001	0.6365	0.817	0.7268
12	anthropic.claude-3-sonnet-20240229-	0.6235	0.8176	0.7205
17	anthropic.claude-3-haiku-20240307-v1:0	0.6214	0.7564	0.6889
34	meta-llama/Meta-Llama-3-70B-Instruct	0.6213	0.7499	0.6856
4	cyberagent/calm3-22b-chat	0.625	0.7164	0.6707
2	tokyotech-llm/Llama-3-Swallow-70B-	0.5802	0.7556	0.6679
35	Qwen/Qwen2-7B-Instruct	0.5516	0.7114	0.6315
6	karakuri-ai/karakuri-lm-8x7b-chat-v0.1	0.532	0.7121	0.6221
8	elyza/Llama-3-ELYZA-JP-8B	0.5248	0.7192	0.622
25	microsoft/Phi-3-medium-128k-instruct	0.5484	0.6919	0.6202
3	tokyotech-llm/Llama-3-Swallow-8B-	0.507	0.7283	0.6176
14	solar-1-mini-chat-ja	0.5259	0.7063	0.6161
7	karakuri-ai/karakuri-lm-8x7b-instruct-	0.5181	0.6914	0.6048
37	gpt-3.5-turbo-0125	0.5616	0.6156	0.5886
33	meta-llama/Meta-Llama-3-8B-Instruct	0.5021	0.6585	0.5803

GLP : General Language Processing

ALT : Alignment

Total AVG = (Avg. GLP + Avg. ALT)/2

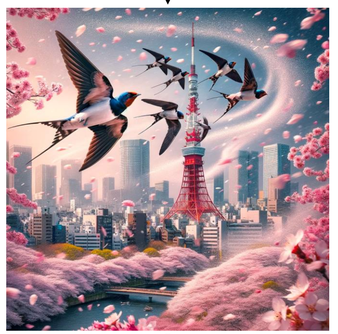
Swallow by Tokyo Tech.

Llama2-based

Swallow-7B
Swallow-13B
Swallow-70B

Mistral-based
Swallow-MS-7B

Mixtral-based
Swallow-MX-8x7B

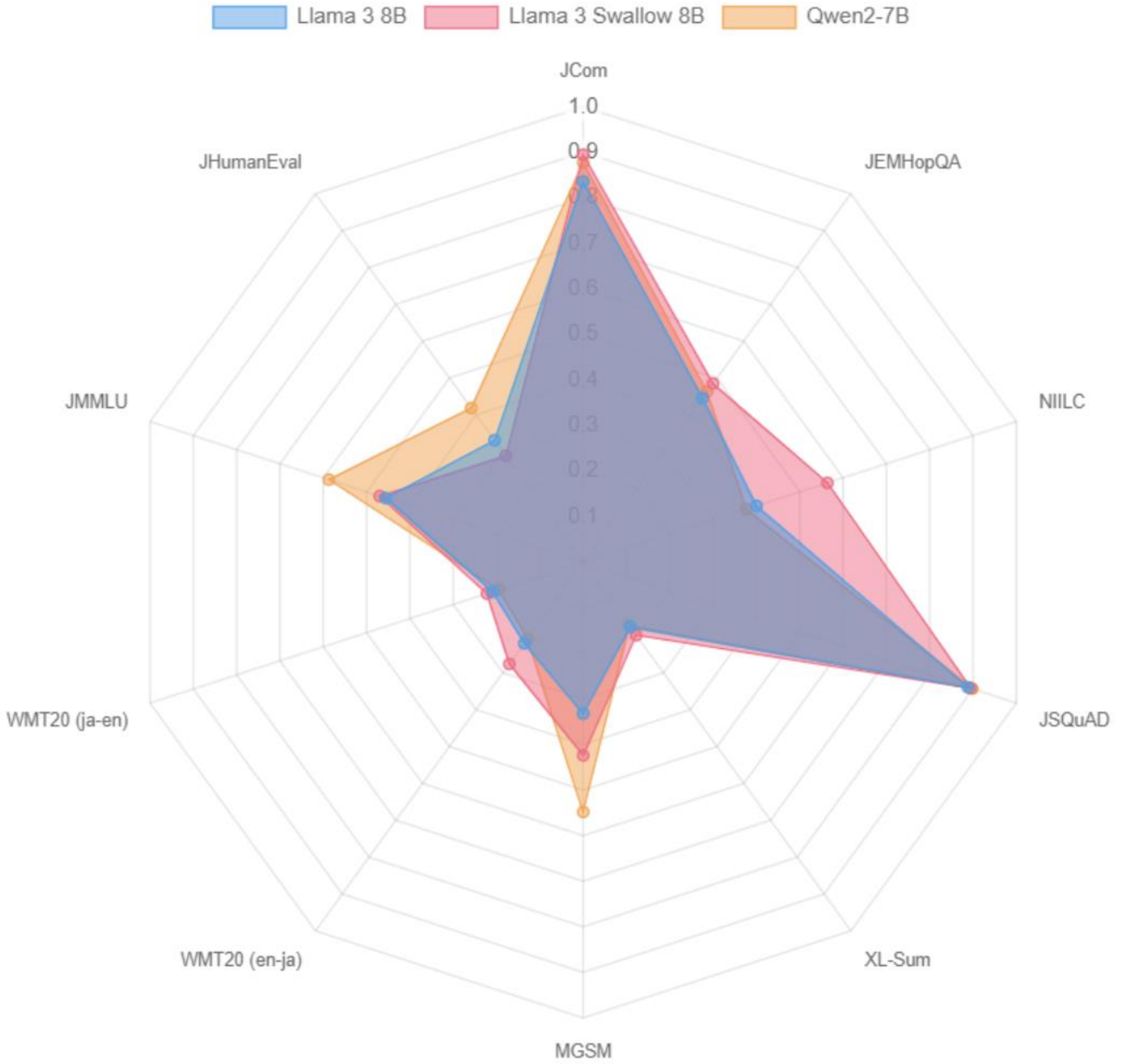


Llama3-based

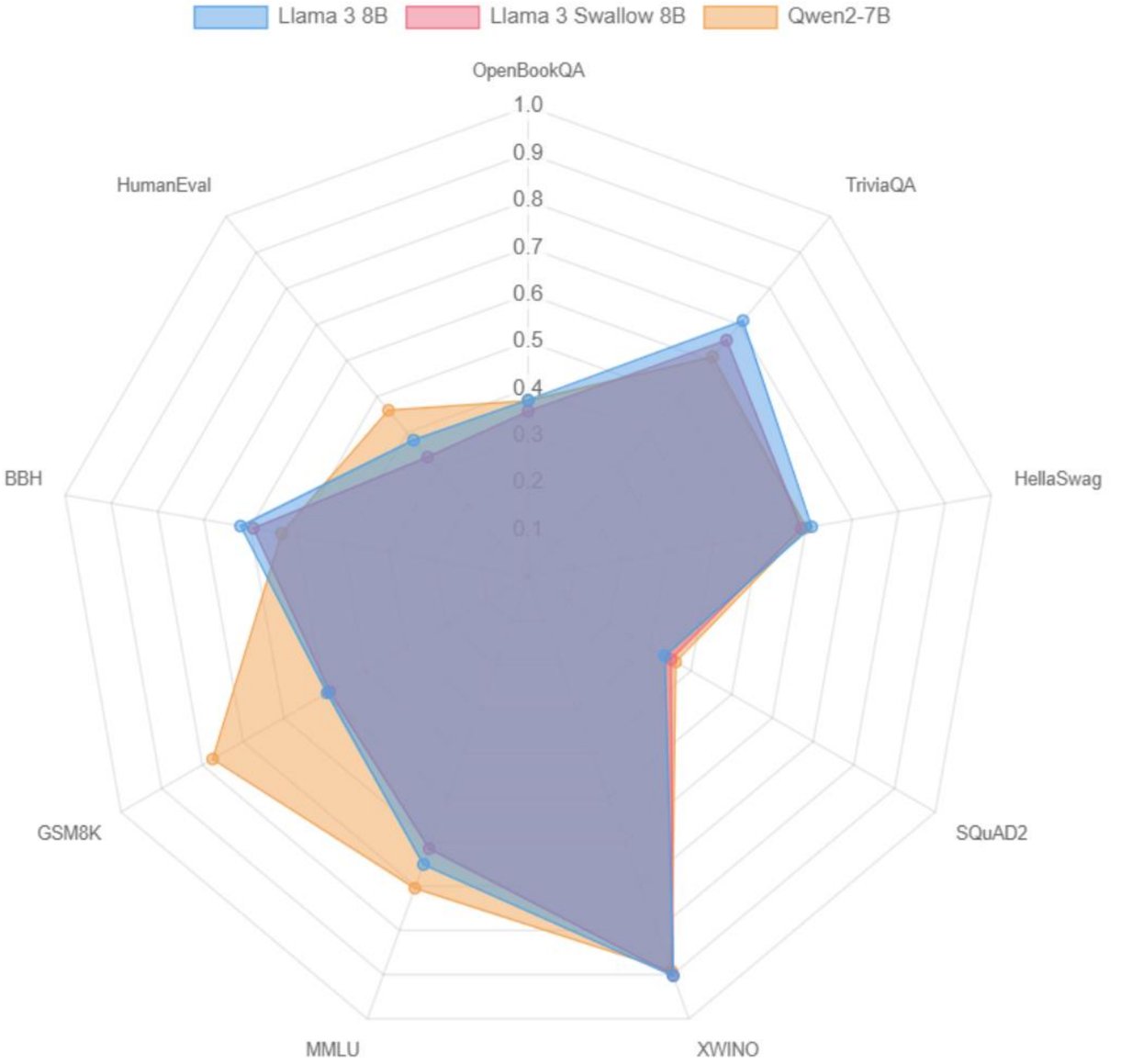
Llama3-Swallow-8B
Llama3-Swallow-70B
Llama3.1-Swallow-8B
Llama3.1-Swallow-70B

https://wandb.ai/wandb-japan/llm-leaderboard3/reports/Nejumi-LLM-3--Vmlldzo3OTg2NjM2

Comparison with Qwen2



Japanese



English

Summary and outlook

Training of larger models is more unstable than smaller models
- Newer transformers have mechanisms like pre-norm, RMS-norm, scaled embedding
- Llama-2 is more stable than GPT

Continual pre-training on non-English languages is an effective approach
- Continual pre-training requires much less data than pre-training from scratch
- Extending the vocabulary can improve token efficiency, but slightly degrades the performance

Two papers presented at COLM'24

Building a Large Japanese Web Corpus for Large Language Models

Naoaki Okazaki, Kakeru Hattori, Hirai Shota, Hiroki Iida, Masanari Ohi, Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Rio Yokota, Sakae Mizuki

Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, Naoaki Okazaki