LLMs for every language

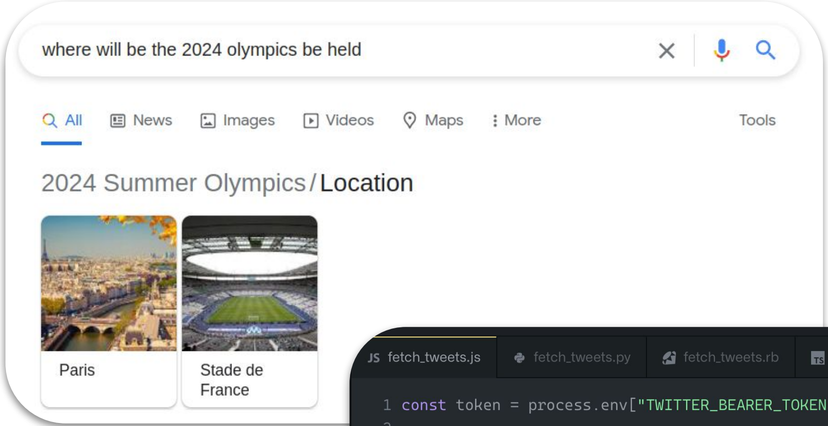
A how-to guide

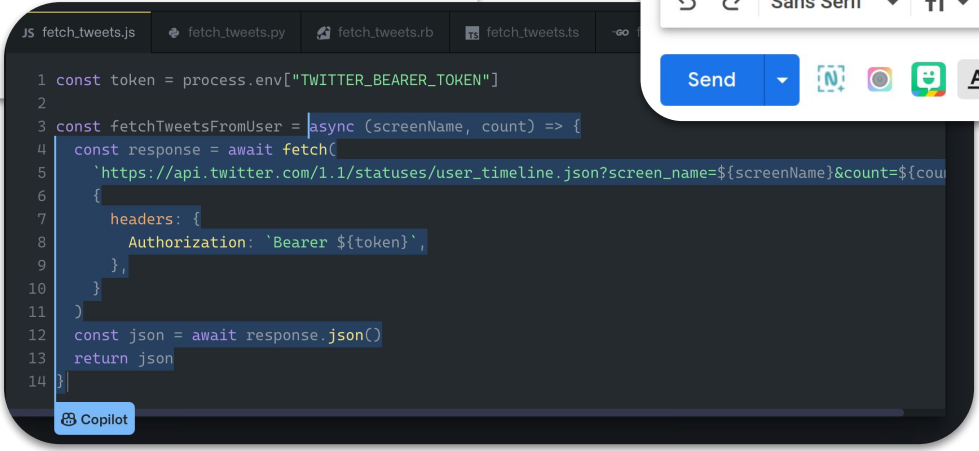
BigScience Collaboratively training a large multilingual language model

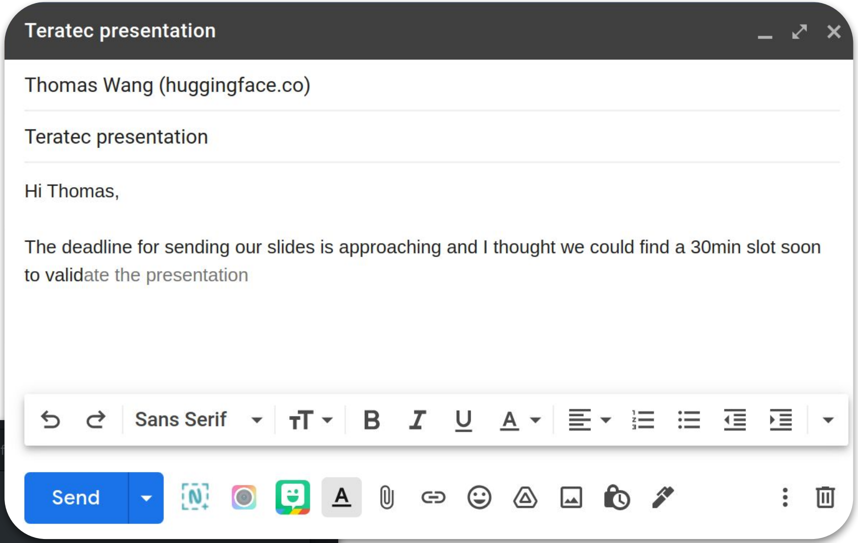
Teven Le Scao

What motivated us to do BigScience?

Language models





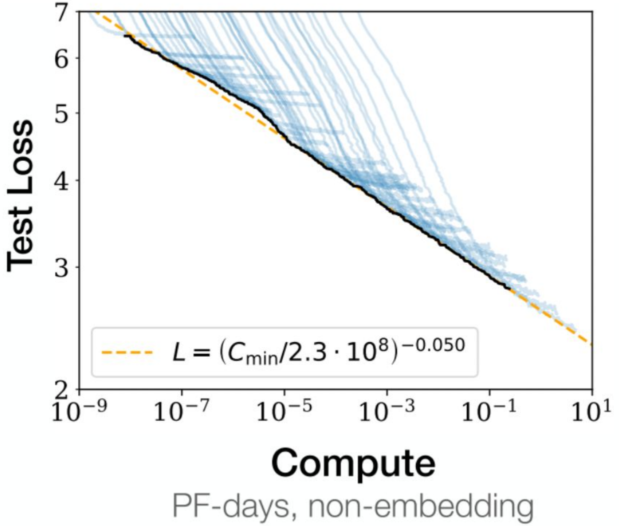


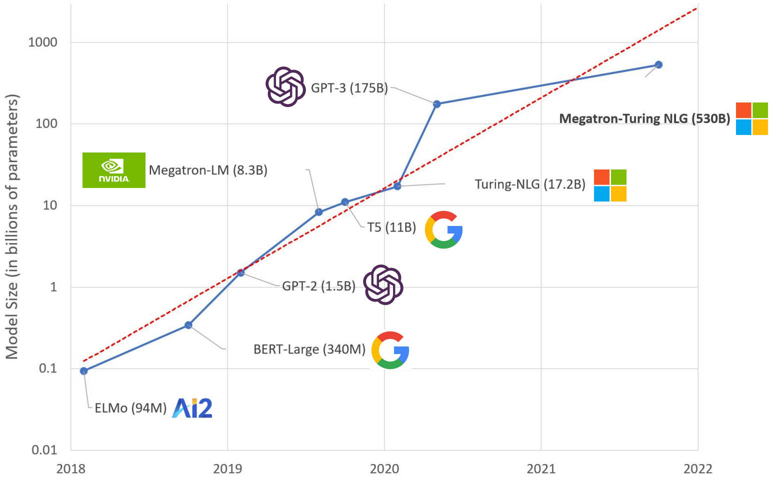
Language models

TE Are language models useful?

Yes, language models are useful for a variety of tasks in natural language processing (NLP), such as machine translation, speech recognition, and text summarization. These models are trained to understand and generate natural language, which allows them to perform tasks that involve understanding and generating human-like text. For example, a language model could be used to automatically generate human-like text responses to customer inquiries, or to help a machine translation system produce accurate and fluent translations. Additionally, language models can be used to improve the performance of other NLP models by providing them with a better understanding of the structure and nuances of natural language.

Scaling





GPT-3's generation example:

[...]

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.

Access

Training cost

- typically $2-5M
- million of gpu hours

Closed access for most of them

VB VentureBeat

Naver trained a 'GPT-3-like' Korean language model

Naver claims the system learned 6,500 times more Korean data than OpenAI's ... Some experts believe that while HyperCLOVA, GPT-3, PanGu-α, ...

1 Jun 2021

TechCrunch

Anthropic is the new AI research outfit from OpenAI's Dario Amodei, and it has $124M to burn

Anthropic, as it's called, was founded with his sister Daniela and its goal is to create "large-scale AI systems that are steerable, ...

28 May 2021

VB VentureBeat

AI21 Labs trains a massive language model to rival OpenAI's GPT-3

"AI21 Labs was founded to fundamentally change and improve the way people read and write. Pushing the frontier of language-based AI requires ...

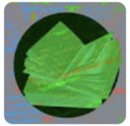
1 month ago

FC Fast Company

Ex-Googlers raise $40 million to democratize language AI

This story has been updated with more information about Cohere's approach to responsible AI. About the author. Fast Company Senior Writer Mark ...

2 days ago



BigScience

“During **one-year**, from May 2021 to May 2022, 1000+ researchers from 60 countries and more than 250 institutions are **creating together a very large multilingual neural network language model** and a **very large multilingual text dataset** on the 28 petaflops Jean Zay (IDRIS) supercomputer located near Paris, France.

During the workshop, the participants plan to investigate the dataset and the model from all angles: bias, social impact, capabilities, limitations, ethics, potential improvements, specific domain performances, carbon impact, general AI/cognitive research landscape.”

Spirit of the project

Make LLM research accessible

- Open-source, open-access
- Organize research around the models
- Make compute accessible

Create and share knowledge around the process

- Train in the open
- Freely discuss engineering problems and solutions

Artifacts: what came out of BigScience? (0/4)

```python
from transformers import AutoModel, AutoTokenizer

model_name = "bigscience/bloom"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)
```

https://huggingface.co/bigscience/bloom

Artifacts: what came out of BigScience? (1/4)

BigScience Workshop

Research workshop on large language models - The Summer of Language Models 21

https://bigscience.huggingface.co @BigScienceW bigscience-contact@googlegroups.com

Overview | Repositories 29 | Projects 4 | Packages | People 22

Popular repositories

promptsource Public
Toolkit for creating, sharing and using natural language prompts.
Python 660 144

bigscience Public
Central place for the engineering/scaling WG: documentation, SLURM scripts and logs, compute environment and data.
Shell 277 26

Megatron-DeepSpeed Public
Ongoing research training transformer language models at scale, including: BERT & GPT-2
Python 145 34

t-zero Public
Reproduce results and replicate training fo T0 (Multitask Prompted Training Enables Zero-Shot Task Generalization)
Python 143 26

biomedical Public
Tools for curating biomedical training data for large-scale language modeling
Python 125 69

evaluation Public
Code and Data for Evaluation WG
Python 36 23

People

View all

Top languages

Python Jupyter Notebook HTML Shell Makefile

Most used topics

machine-learning nlp

Artifacts: what came out of BigScience? (2/4)

What Language Model to Train if You Have One Million GPU Hours?

The BigScience Architecture & Scaling Group

Teven Le Scao[1*] Thomas Wang[1*] Daniel Hesslow[2*] Lucile Saulnier[1*] Stas Bekman[1*]
M Saiful Bari[3] Stella Biderman[4,5] Hady Elsahar[6] Jason Phang[7] Ofir Press[8] Colin Raffel[1]
Victor Sanh[1] Sheng Shen[9] Lintang Sutawika[10] Jaesung Tae[1] Zheng Xin Yong[11]
Julien Launay[2,12†] Iz Beltagy[13†]

[1] Hugging Face [2] LightOn [3] NTU, Singapore [4] Booz Allen [5] EleutherAI [6] Naver Labs Europe [7] New York University
[8] University of Washington [9] Berkeley University [10] Big Science [11] Brown University [12] LPENS [13] Allen Institute for AI

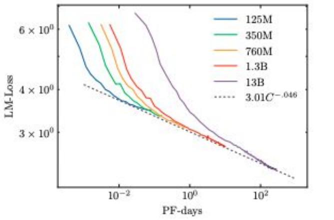


Figure 1: **Smooth scaling of language modeling loss as compute budget and model size increase.** We observe a power-law coefficient $\alpha_C \sim 0.046$, in-line with pre-

Multitask Prompted Training Enables Zero-Shot Task Generalization

Victor Sanh* Hugging Face | Albert Webson* Brown University | Colin Raffel* Hugging Face | Stephen H. Bach* Brown & Snorkel AI

Lintang Sutawika BigScience | Zaid Alyafeai KFUPM | Antoine Chaffin IRISA & IMATAG | Arnaud Stiegler Hyperscience | Teven Le Scao Hugging Face

Arun Raja I²R, Singapore | Manan Dey SAP | M Saiful Bari NTU, Singapore | Canwen Xu UCSD & Hugging Face | Urmish Thakker SambaNova Systems

Shanya Sharma Walmart Labs | Eliza Szczechla BigScience | Taewoon Kim VU Amsterdam | Gunjan Chhablani BigScience | Nihal V. Nayak Brown University

Debajyoti Datta University of Virginia | Jonathan Chang ASUS | Mike Tian-Jian Jiang ZEALS, Japan | Han Wang NYU | Matteo Manica IBM Research

Sheng Shen UC Berkeley | Zheng-Xin Yong Brown University | Harshit Pandey BigScience | Michael McKenna Parity | Rachel Bawden Inria, France

Thomas Wang Inria, France | Trishala Neeraj BigScience | Jos Rozen Naver Labs Europe | Abheesht Sharma BITS Pilani, India | Andrea Santilli University of Rome

...lan Fries ... & Snorkel AI | Ryan Teehan Charles River Analytics | Tali Bers Brown University

Leo Gao EleutherAI | Thomas Wolf Hugging Face | Alexander M. Rush Hugging Face

ABSTRACT

...ls have recently been shown to attain reasonable zero-shot ...verse set of tasks (Brown et al., 2020). It has been hypothe... ...sequence of implicit multitask learning in language models' ...t al., 2019). Can zero-shot generalization instead be directly ...ltitask learning? To test this question at scale, we develop

...deling methods ...ecture has been a ..., well-motivated ... transfer across ... impact of mod... ...h the emergence ...ameters models, ...reasingly expen... ...l train. Notably, ... how modeling ...ent capabilities, ...ise mainly from ...lingual language ...s scale, our goal ...nd training setup

Masader: Metadata Sourcing for Arabic Text and Speech Data Resources

Zaid Alyafeai[1], Maraim Masoud[2], Mustafa Ghaleb[1], and Maged S. Al-shaibani[1]

[1] King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
[2] Independent Researcher

Abstract

The NLP pipeline has evolved dramatically in the last few years. The first step in the pipeline is to find suitable annotated datasets to evaluate the tasks we are trying to solve. Unfortunately, most of the published datasets lack metadata annotations that describe their attributes. Not to mention, the absence of a public catalogue that indexes all the publicly available datasets related to specific regions or languages. When we consider low-resource dialectical languages, for example, this issue becomes more prominent. In this paper we create *Masader*, the largest public catalogue for Arabic NLP datasets, which consists of 200 datasets annotated with 25 attributes. Further-

and so on. This study attempts to identify the publicly available Arabic NLP datasets and to provide a catalogue of Arabic datasets to researchers. The catalogue will increase the discoverability and provide some key metadata that will help researchers identify the most suitable dataset for their research questions.

Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP

Sabrina J. Mielke [1,2] Zaid Alyafeai [3] Elizabeth Salesky [1]
Colin Raffel [2] Manan Dey [4] Matthias Gallé [5] Arun Raja [6]
Chenglei Si [7] Wilson Y. Lee [8] Benoît Sagot [9*] Samson Tan [10*]
BigScience Workshop Tokenization Working Group

[1]Johns Hopkins University [2]HuggingFace [3]King Fahd University of Petroleum and Minerals [4]SAP
[5]Naver Labs Europe [6]Institute for Infocomm Research, A*STAR Singapore [7]University of Maryland
[8]BigScience Workshop [9]Inria Paris [10]Salesforce Research Asia & National University of Singapore
sjm@sjmielke.com

Abstract

What are the units of text that we want to model? From bytes to multi-word expressions, text can be analyzed and generated at many granularities. Until recently, most natural language processing (NLP) models operated over words, treating those as discrete and atomic tokens, but starting with byte-pair encoding (BPE), subword-based approaches have become dominant in many areas, enabling small vocabularies while still allowing for fast inference. Is the end of the road character-level model or byte-level processing? In this survey, we connect several lines of work from the pre-neural and neural era, by showing how hybrid approaches

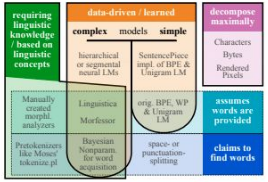


Figure 1: A taxonomy of segmentation and tokenization algorithms and research directions

What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?

The BigScience Architecture & Scaling Group

Thomas Wang[1*] Adam Roberts[2*]
Daniel Hesslow[3] Teven Le Scao[1] Hyung Won Chung[2]
Iz Beltagy[4] Julien Launay[3,5†] Colin Raffel[1†]

[1] Hugging Face [2]Google [3]LightOn
[4]Allen Institute for AI [5]LPENS, École Normale Supérieure

Abstract

Large pretrained Transformer language models have been shown to exhibit *zero-shot generalization*, i.e. they can perform a wide variety of tasks that they were not explicitly trained on. However, the architectures and pretraining objectives used across state-of-the-art models differ significantly, and there has been limited systematic comparison of these factors. In this work, we present a large-scale evaluation of modeling choices and their impact on zero-shot generalization. In particular, we focus on text-to-text models and experiment with three model architectures (causal/non-causal decoder-only and encoder-decoder), trained with two different pretraining objectives (autoregressive and masked language modeling), and evaluated with and without multitask prompted finetuning. We train

And many mores..

Artifacts: what came out of BigScience? (3/4)

BigScience

BigScience RAIL License v1.0

This is the home of the BigScience RAIL License v1.0.If you would like to download the license you can get it as **.txt**, **.docx**, or **.html** file.

https://huggingface.co/spaces/bigscience/license

Artifacts: what came out of BigScience? (4/4)

Stas Bekman
@StasBekman

The embed matrix with 250k multi-lingual vocab is on par in size with the transformer block, so rebalancing the pipeline to count embedding matrices as transformer blocks leads to even faster throughput and less memory usage on ranks 0 and -1

Benchmarks:

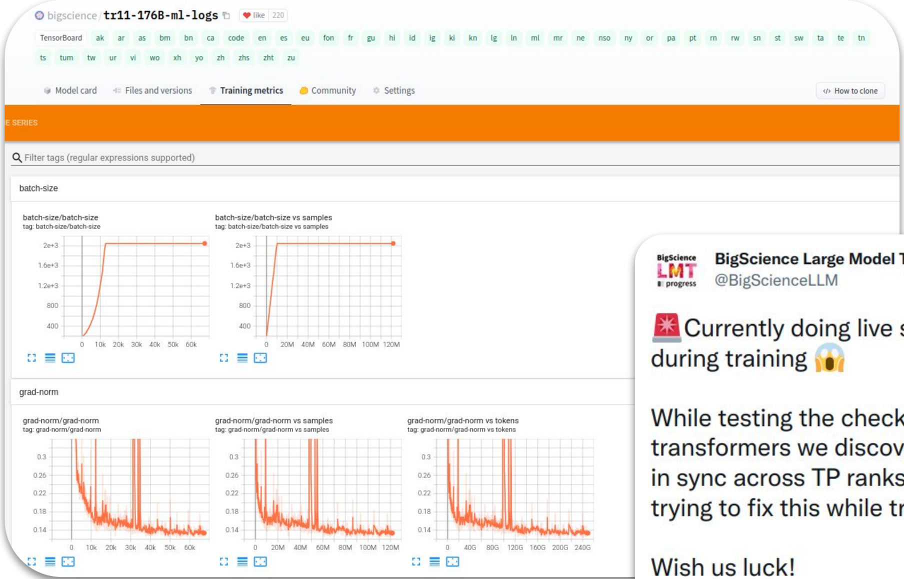
bigscience-workshop/
bigscience

Central place for the engineering/scaling WG: documentation, SLURM scripts and logs, compute environment and data.

13 Contributors · 2 Issues · 272 Stars · 24 Forks

github.com
bigscience/chronicles-prequel.md at master · bigscience-workshop/bigscience
Central place for the engineering/scaling WG: documentation, SLURM scripts and logs, compute environment and data. - bigscience/chronicles-prequel.md a...

6:34 AM · Mar 4, 2022 · Twitter Web App



BigScience Large Model Training
@BigScienceLLM

🚨Currently doing live surgery on the 176B model during training 😱

While testing the checkpoint weights for integration in transformers we discovered that layer norms were not in sync across TP ranks contrary to what expected - trying to fix this while training 🤞

Wish us luck!

12:38 PM · Mar 25, 2022 · Twitter Web App

https://huggingface.co/bigscience/tr11-176B-ml-logs

https://github.com/bigscience-workshop/bigscience/blob/master/train/lessons-learned.md

Training a 176B model

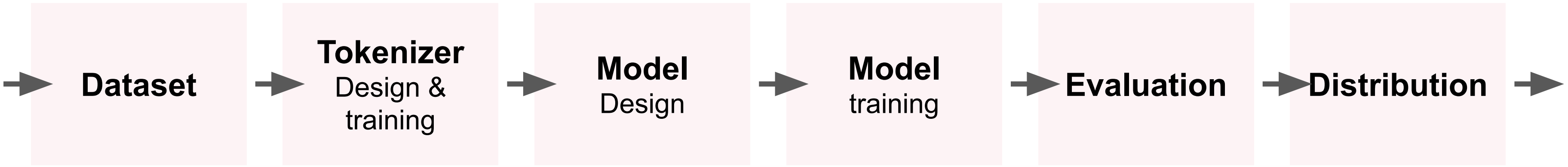
Jean Zay

- This work was granted access to the HPC resources of *Institut du développement et des ressources en informatique scientifique* (IDRIS) du *Centre national de la recherche scientifique* (CNRS) under the allocation 2021-A0101012475 made by *Grand équipement national de calcul intensif* (GENCI) - Thank you!
- Compute grant:
    - 2.5M V100 hours
    - 1.25M A100 hours: a reserved allocation of 416 A100 (80GB)
    - and a ton of CPU
- Technical support - Thanks Rémi Lacroix!

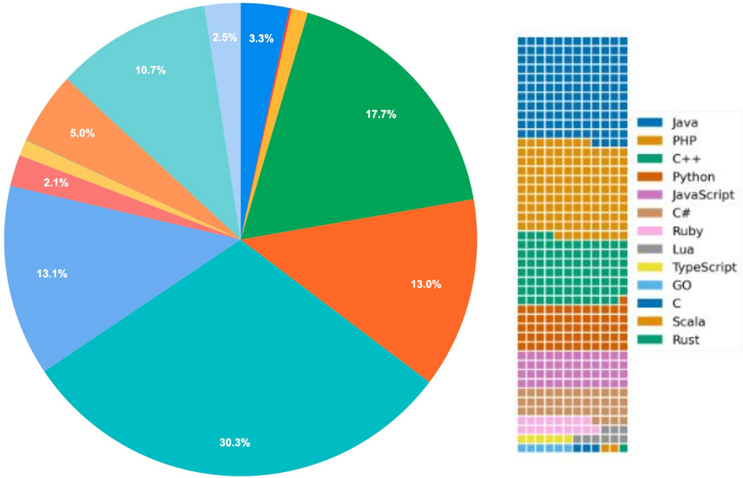
INSTITUT DU DÉVELOPPEMENT ET DES RESSOURCES EN INFORMATIQUE SCIENTIFIQUE

GENCI

Pipeline


Dataset
Tokenizer
Design &
training
Model
Design
Model
training
Evaluation
Distribution

Dataset: 1.6T multilingual



- Arabic (3,3%)
- Basque (0,2%)
- Catalan (1,1%)
- Chinese (17,7%)
- Code (13%)
- English (30,3%)
- French (13,1%)
- Indic (2,1%)
- Indonesian (1,1%)
- Niger Congo (0,03%)
- Portuguese (5%)
- Spanish (10,7%)
- Vietnamese (2,5%)

https://openreview.net/forum?id=UoEw6KigkUn

Sources: catalogue

Crowdsourced multilingual datasets from BigScience participants

~60% of data in tokens

Lessons:

- Some filtering and deduplication required, lots of templates to remove
- A lot of those are *not* clean
- LMs require specific corpora: unsupervised, diverse text with long documents

Web crawl

Our filtered and deduped version of OSCAR-v1

~35% of data in tokens

Lessons:

- Heavy filtering needed, but you can find good data
- Hard to assess deduplication at TB-scale

Web crawl filtering

7 simple filters

AR	EU	BN	CA	ZH	EN	FR	HI	ID	PT	UR	VI	ES
20.3	5.2	48.8	21.1	23.1	17.2	17.0	25.7	10.4	12.6	15.8	21.3	16.9

Table 1: Percentage of documents removed by the filtering per language (ISO 639-1 code).

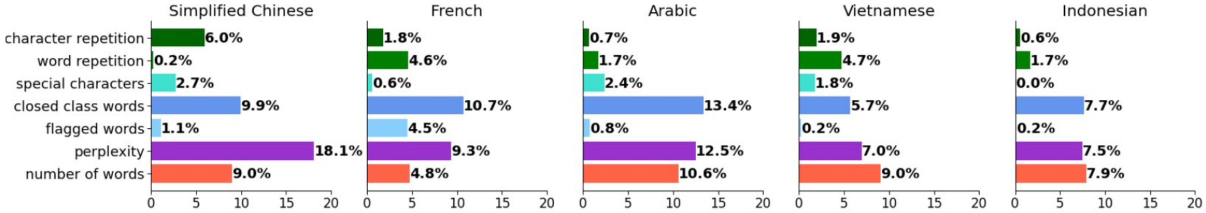


Figure 3: Percentage of documents discarded by each filter independently for 5 languages

Sources: pseudo-crawl

List of domains from native speakers, corresponding Common Crawl WARC files

5% of data

Lessons:

- Doing your own WARC parsing is rewarding but a project in itself
- We probably should have crawled the sites directly
- Tons of processing/deduplication/template removal needed

Deduplication

For catalogue and pseudo-crawl

- Line by line within every document
- Across all documents

And for the web crawl:

- MinHash near-dedup (0.7% of data removed, wary of false +)
- Suffix tree exact dedup over long documents (21.67% duplication, stringent)

Tokenizer choices

Metric: **fertility**

- Minimize vocab size
- Minimize tokens per byte of text

Tokenizer choices

Objective: no more than 10%+ fertility compared to monolingual tokenizers

Reached at **250k tokens**

Tokenizer	**fr**	**en**	**es**	**zh**	**hi**	**ar**
Monolingual	1.30	1.15	1.12	1.50	1.07	1.16
BLOOM	1.17 (-11%)	1.15 (+0%)	1.16 (+3%)	1.58 (+5%)	1.18 (+9%)	1.34 (+13%)

Table 2: Fertilities obtained on Universal Dependencies treebanks on languages with existing monolingual tokenizers. The monolingual tokenizers we used were the ones from CamemBERT (Martin et al., 2020), GPT-2 (Radford et al., 2019), `DeepESP/gpt2-spanish`, `bert-base-chinese`, `monsoon-nlp/hindi-bert` and Arabic BERT (Safaya et al., 2020), all available on the HuggingFace Hub.

Lessons:

- Weird tokens (e.g. URLs) are good indicators of training data pathologies
- Massively multilingual vocabs are hard… Still messed up for Devanagari

GPT-style autoregressive

GPT3

RESEARCH

Democratizing access to large-scale language models with OPT-175B

May 3, 2022

Publication

An empirical analysis of compute-optimal large language model training

Download | View publication

Google AI Blog

The latest from Google Research

Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance

Monday, April 4, 2022

Announcing AI21 Studio and Jurassic-1 Language Models

AI21 Labs' new developer platform offers instant access to our 178B-parameter language model, to help you build sophisticated text-based AI applications at scale

Research

Language modelling at scale: Gopher, ethical considerations, and retrieval

December 8, 2021

YUAN 1.0: LARGE-SCALE PRE-TRAINED LANGUAGE MODEL IN ZERO-SHOT AND FEW-SHOT LEARNING

Shaohua Wu* | Xudong Zhao | Tong Yu
Rongguo Zhang | Chong Shen | Hongli Liu | Feng Li
Hong Zhu | Jiangang Luo | Liang Xu | Xuanwei Zhang

Announcing GPT-NeoX-20B

Announcing GPT-NeoX-20B, a 20 billion parameter model trained in collaboration with CoreWeave.

February 2, 2022 · Connor Leahy

Code stack

```
~checkpoints/tr11-176B-ml/checkpoints/main> du -h global_step63600/
2.3T  global_step63600/
```

🏋 Including optimizer states and checkpoints

microsoft / **Megatron-DeepSpeed** Public

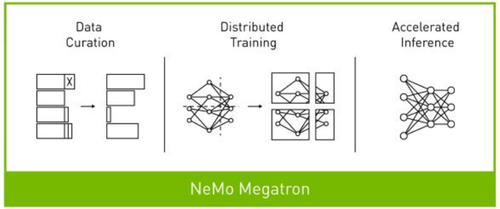
forked from NVIDIA/Megatron-LM

Code stacks I would use now

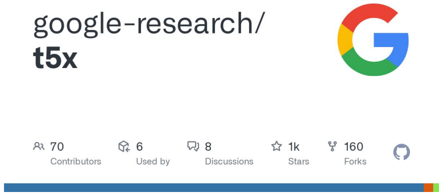
- If you have high interconnect

deepspeed

- If not

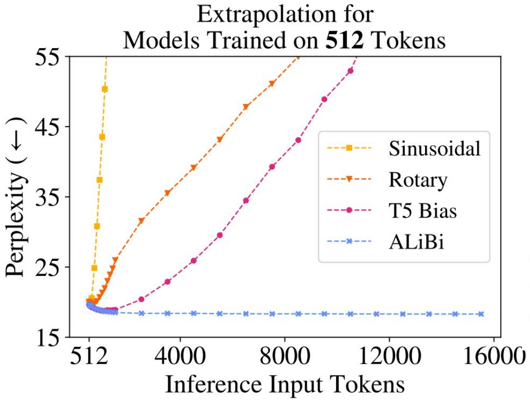


- If you have TPUs



Modeling adjustments

- **ALiBi** positional embeddings which allow long-sequence extrapolation



Positional Embedding	Average EAI Results
None	41.23
Learned	41.71
Rotary	41.46
ALiBi	**43.70**

Table 2: **ALiBi significantly outperforms other embeddings for zero-shot generalization.** All models are trained on the OSCAR dataset for 112 billion tokens.

- **Embed LayerNorm** for stability (there's a perf tradeoff though!)

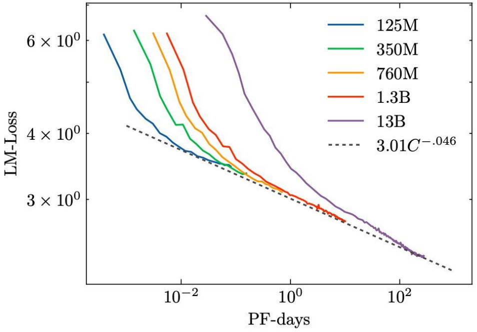
Scaling

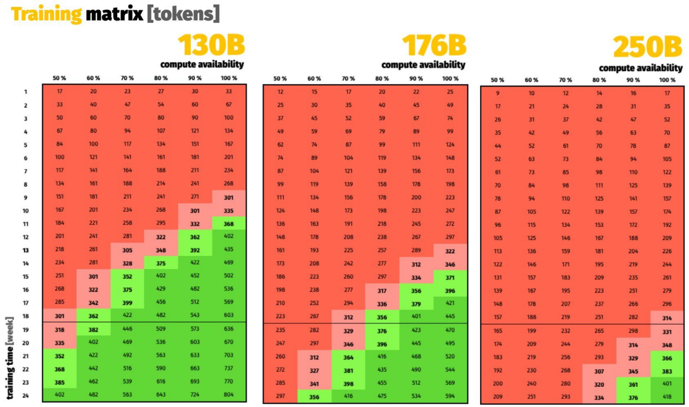
Compute (total operations) = k * Data * Parameters

As C (compute) goes up, how much should go into D (data size) vs N (params)?

Scaling

Scaling laws on English





Fixed budget scenarios for different model sizes

Model	Size [Bparams.]	Pretraining [Btokens]	Budget [PF-days]	Layers	Hidden dim.	Attention heads num.	Attention heads dim.
LaMDA (Thoppilan et al., 2022)	137	432	4,106	64	8,192	128	64
GPT-3 (Brown et al., 2020)	175	300	3,646	96	12,288	96	128
J1-Jumbo (Lieber et al., 2021)	178	300	3,708	76	13,824	96	144
PanGu-α (Zeng et al., 2021)	207	42	604	64	16,384	128	128
Yuan (Wu et al., 2021)	245	180	3,063	76	16,384		
Gopher (Rae et al., 2021)	280	300	4,313	80	16,384	128	128
MT-530B (Smith et al., 2022)	530	270	9,938	105	20,480	128	160

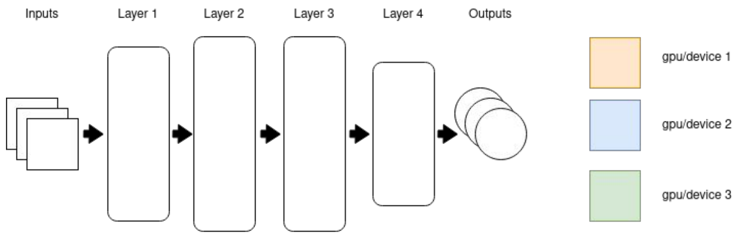
arxiv.org/abs/2001.08361

arxiv.org/abs/2006.12467

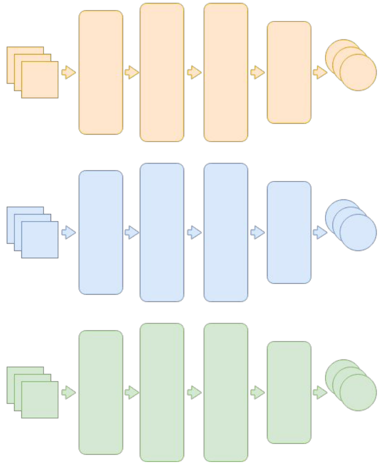
Model	Size [params.]	Layers	Hidden dim.	Attention heads num.	Attention heads dim.	Memory [GB]	Performance [sec/iter.]	Performance [TFLOPs]
(1)	178	82	13,312	64	208	63	104	152
(2)	178	82		128	104	60	109	146
(3)	**176**	**70**	**14,336**	**112**	**128**	**59**	**105**	**150**

Engineering constraints

Parallelism:
how to use a cluster wisely for DL?



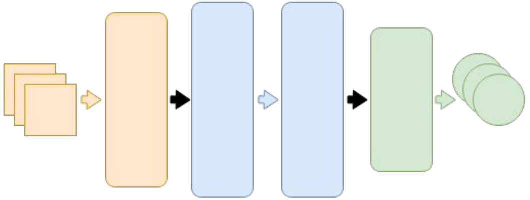
Data parallelism
to accelerate the training speed



Each device has a replica of the model and receives a different batch of training data on which it performs a forward and backward pass

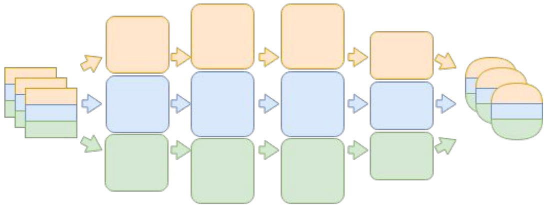
Model parallelism
to train models that don't fit in the memory of one device

Pipeline parallelism



Only one or several consecutive layers of the model are placed on a single GPU

Tensor parallelism

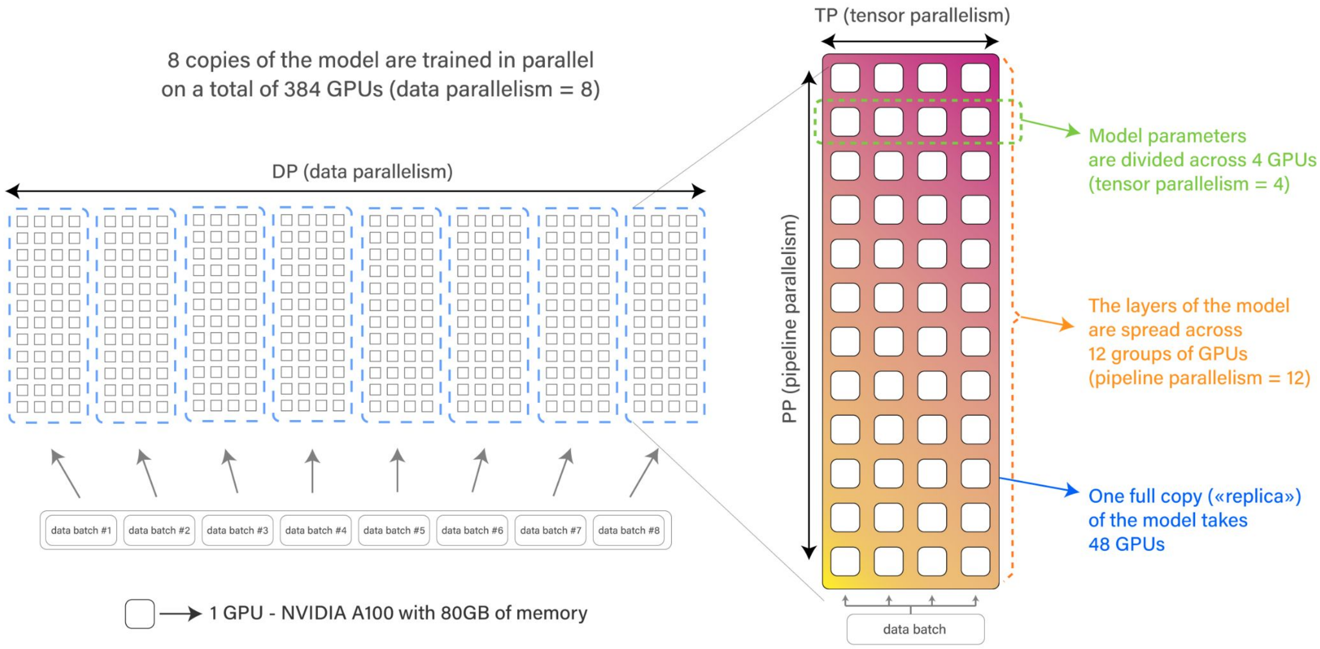


Each tensor is divided into several pieces so that instead of having the whole tensor residing on a single GPU each piece of the tensor resides on a different GPU

Parallelism:

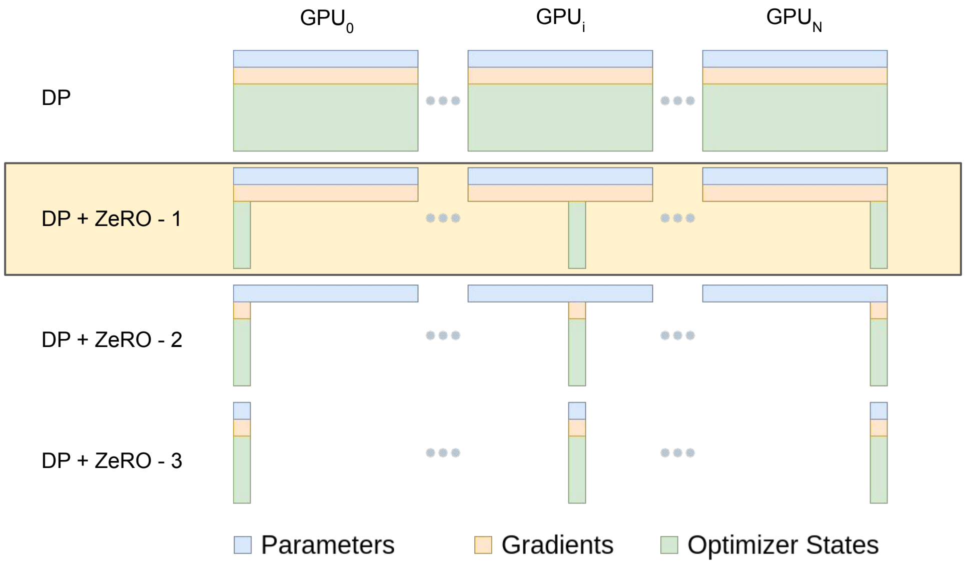
DP?
TP?
PP?

All 3 techniques were used!



ZeRO data parallelism

- instead of replicating everything each GPU stores only a slice of it
- free the GPUs for larger batch sizes or more layers



Memory Consumption		Comm Volume
Formulation	Specific Example K=12 Ψ=7.5B N_d=64	
$(2+2+K)*\Psi$	120GB	1x
$2\Psi+2\Psi+\frac{K*\Psi}{N_d}$	31.4GB	1x
$2\Psi+\frac{(2+K)*\Psi}{N_d}$	16.6GB	1x
$\frac{(2+2+K)*\Psi}{N_d}$	1.9GB	1.5x

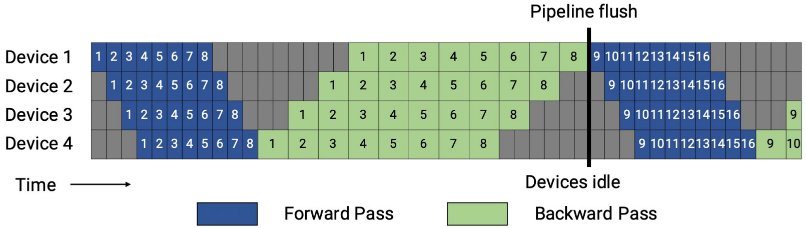
https://huggingface.co/docs/transformers/parallelism#zero-data-parallelism

Pipeline scheduling

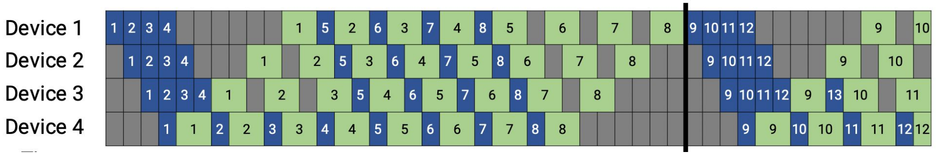
👇The one we used

All forward, all backward

Reduce memory →

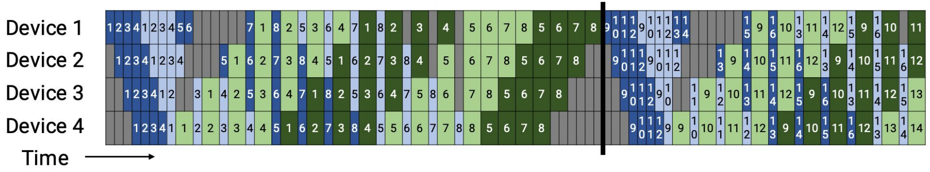


One forward, one backward (1f1b)

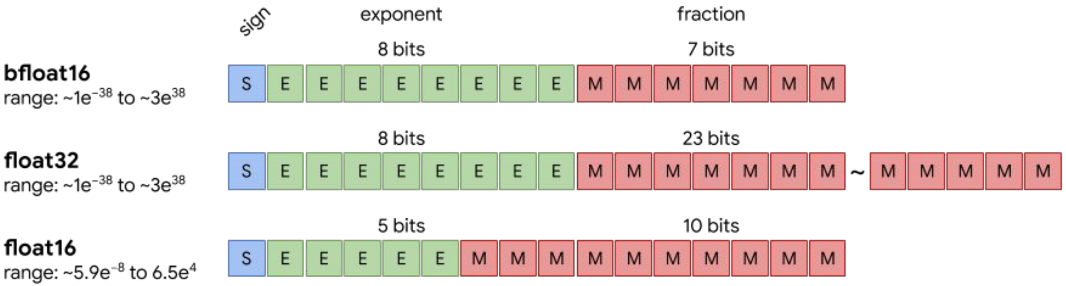


Reduce bubble at the cost of communication

Interleaved 1f1b



BF16



Source: https://moocaholic.medium.com/fp64-fp32-fp16-bfloat16-tf32-and-other-members-of-the-zoo-a1ca7897d407

Evaluation: is hard

🚘 **Extrinsic Evaluation**: Focus on downstream, user-facing tasks

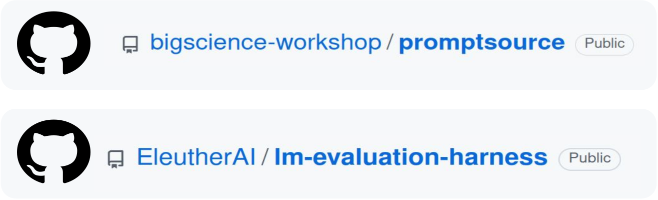
⚙️ **Intrinsic Evaluation**: Focus on encoding of linguistic and world knowledge

🌈 **Bias/Social Impact**: Quantify encoding of stereotypes and risk of user harm

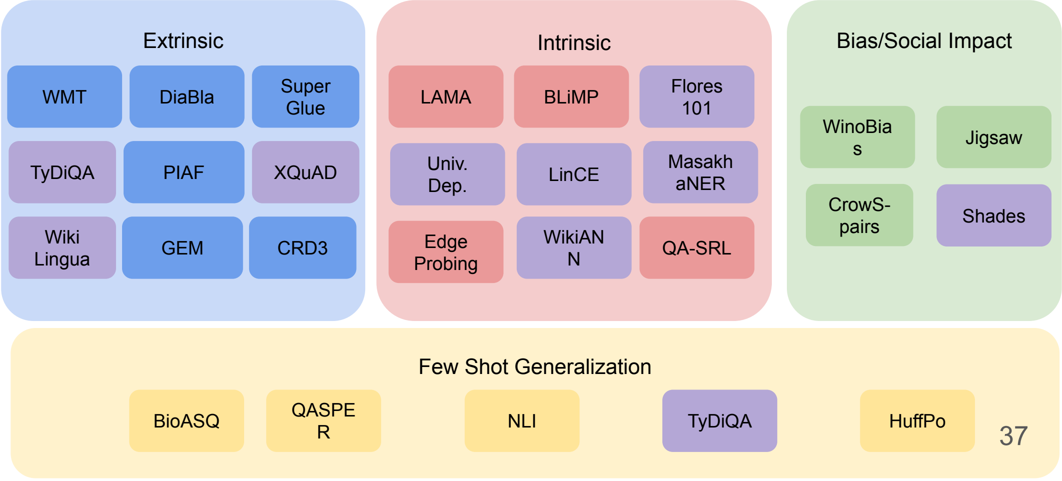
🗣️ **Multilingualism**: Ensure coverage of training and unseen language in all evaluations

🍎 🍊**Few-Shot Generalization**: Focus on evaluation on distributions not seen in pretraining

Code Bases

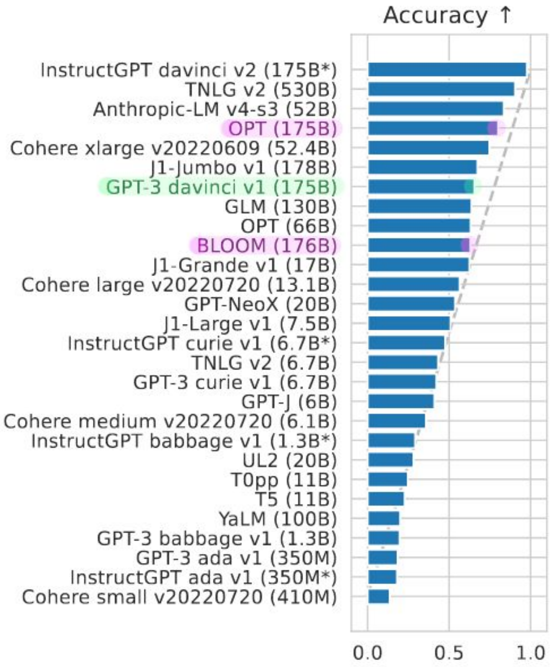


Benchmark



After training

HELM



In 2022, open source was ~1.5 years behind closed-source

Zero-shot

Checkpoint: 65k steps (240B tokens)

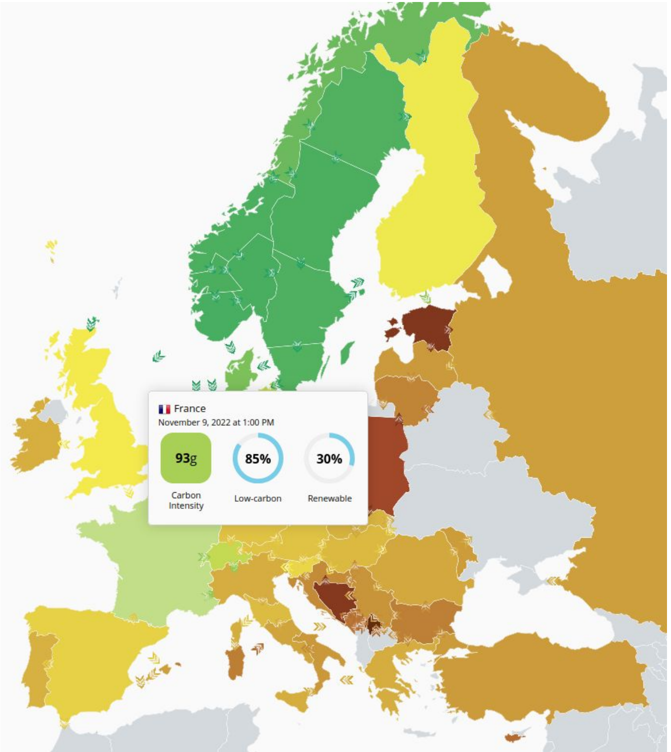
```
To do a "farduddle" means to jump up and down
really fast. An example of a sentence that uses
the word farduddle is:
```

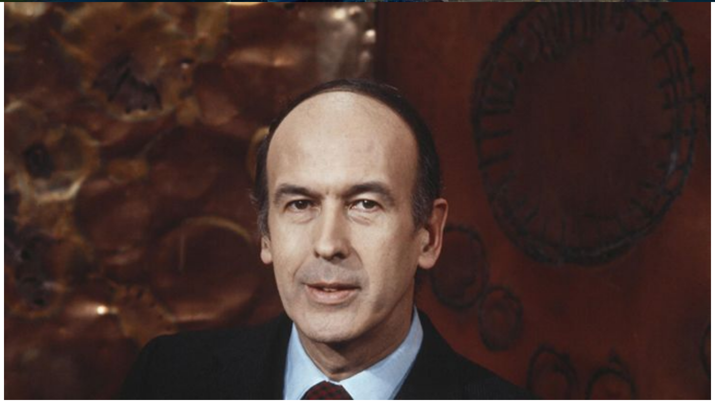
Prompt

`We were on a Kindrex toyshop hop and we got to see if Kindrex toys were built to withstand constant jumping, lots of jumping! To do this part, I had to farduddle to simulate jumping.`

Generated

Carbon footprint





Carbon footprint

ESTIMATING THE CARBON FOOTPRINT OF BLOOM, A 176B PARAMETER LANGUAGE MODEL

Alexandra Sasha Luccioni
Hugging Face
sasha.luccioni@hf.co

Sylvain Viguier
Graphcore
sylvainv@graphcore.ai

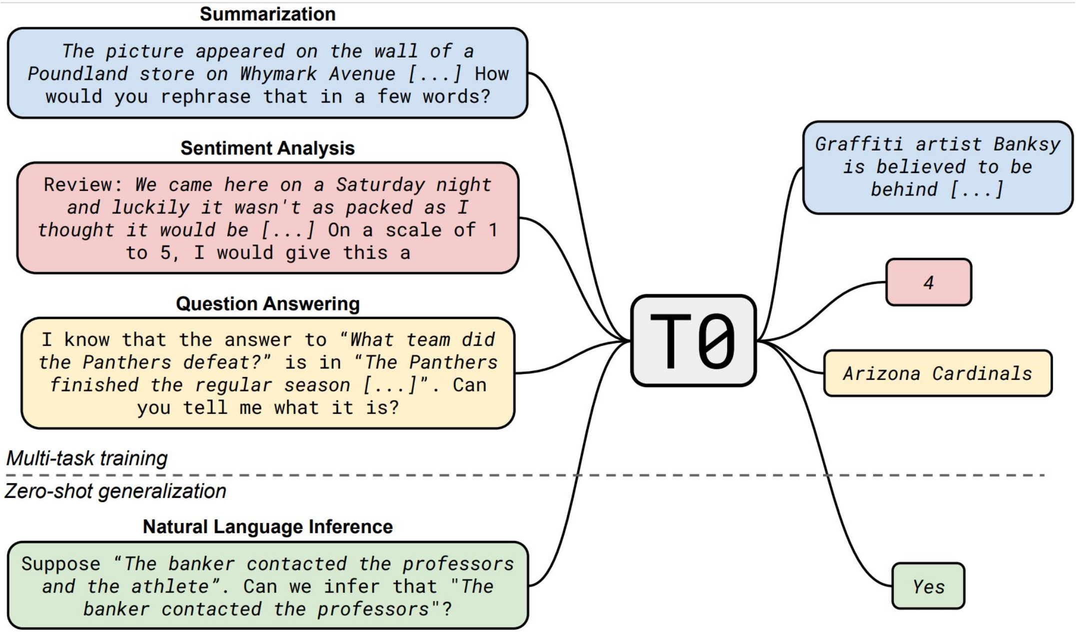
Anne-Laure Ligozat
LISN & ENSIIE
anne-laure.ligozat@lisn.upsaclay.fr

Model name	**Number of parameters**	**Datacenter PUE**	**Carbon intensity of grid used**	**Power consumption**	**CO_2eq emissions**	**CO_2eq emissions $\times$ PUE**
GPT-3	175B	1.1	429 gCO_2eq/kWh	1,287 MWh	*502 tonnes*	552 tonnes
Gopher	280B	1.08	330 gCO_2eq/kWh	*1,066 MWh*	*352 tonnes*	380 tonnes
OPT	175B	*1.09*[2]	*231 gCO_2eq/kWh*	*324 MWh*	70 tonnes	*76.3 tonnes*[3]
BLOOM	176B	1.2	57 gCO_2eq/kWh	433 MWh	25 tonnes	30 tonnes

~ 20 Paris <-> NYC return trips

Example generations

Multi-task fine-tuning



Code models

BigCode is an open scientific collaboration working on the responsible development of large language models for code

Learn more...

Supported by:

servicenow

Hugging Face

https://www.bigcode-project.org/

Monolingual EU LMs?

Scaling

Compute (total operations) = k * Data * Parameters

As C (compute) goes up, how much should go into D (data size) vs N (params)?

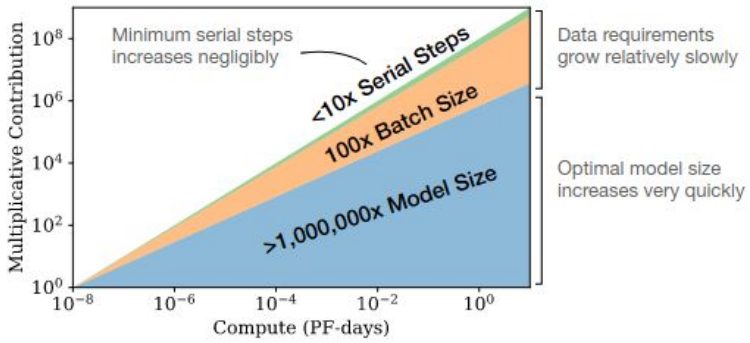
The scaling controversy

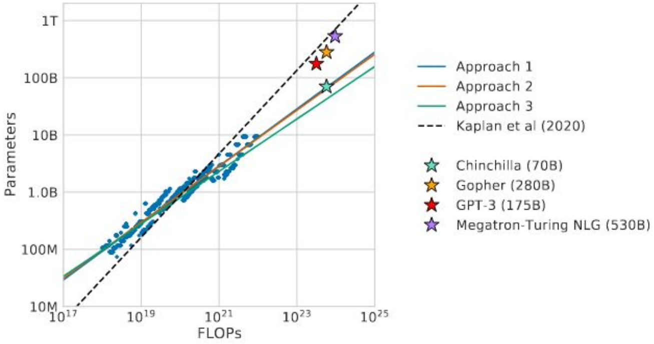
Kaplan '20:

D ~ $C^{0.27}$, N ~ $C^{0.73}$

Hoffman '22:

D ~ $C^{0.50}$, N ~ $C^{0.50}$





Data, data, data

BLOOM: 384 A100s for 111 days

-> 5.5E+23 operations

Chinchilla optimality:

59B parameters, for 1,18T tokens

BLOOM:

176B parameters, for 366B tokens

Data, data, data

So where do you find 1T tokens? ~4TB text, 800B words.

Data, data, data

So where do you find 1T tokens? ~4TB text, 800B words.

- Even in English, finding 1T quality tokens is non-trivial
- Norwegian OSCAR: 2.8GB

Data, data, data

So where do you find 1T tokens? ~4TB text, 800B words.

- Even in English, finding 1T quality tokens is non-trivial
- Norwegian OSCAR: 2.8GB

Libraries, silver bullet? The French National Library contains roughly 800B words

Our current work

Niklas Muennighoff, HF

Nouamane Tazi, HF

Sampo Pyysalo, Turku

Thom Wolf, HF

Ola Piktus, HF

Teven Le Scao, HF

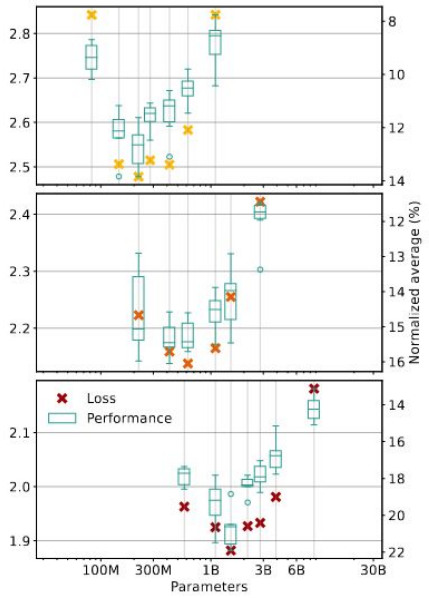
UNIVERSITY OF TURKU

LUMI

The Supercomputer of the North

Val loss vs downstream zero-shot perf

Chinchilla-optimal models for loss are also optimal for 0-shot performance.



Is repeating data bad?

Validation loss: yes

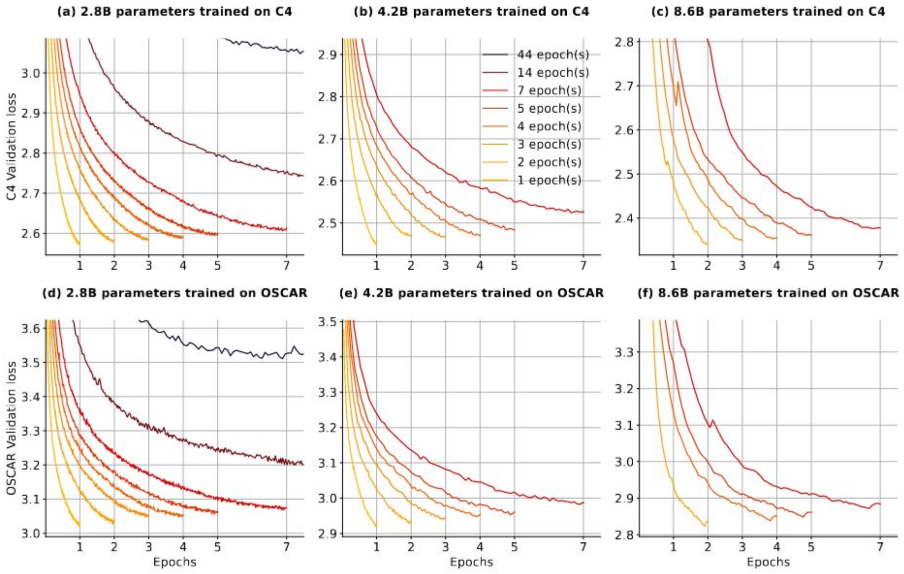


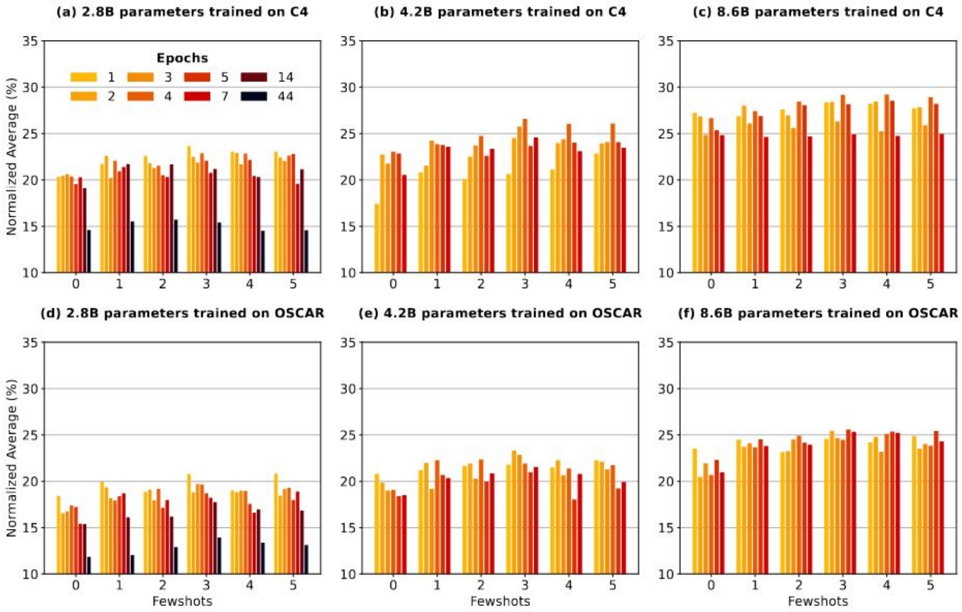
Figure 3: Epochs

Is repeating data bad?

Zero- or few-shot downstream perf:

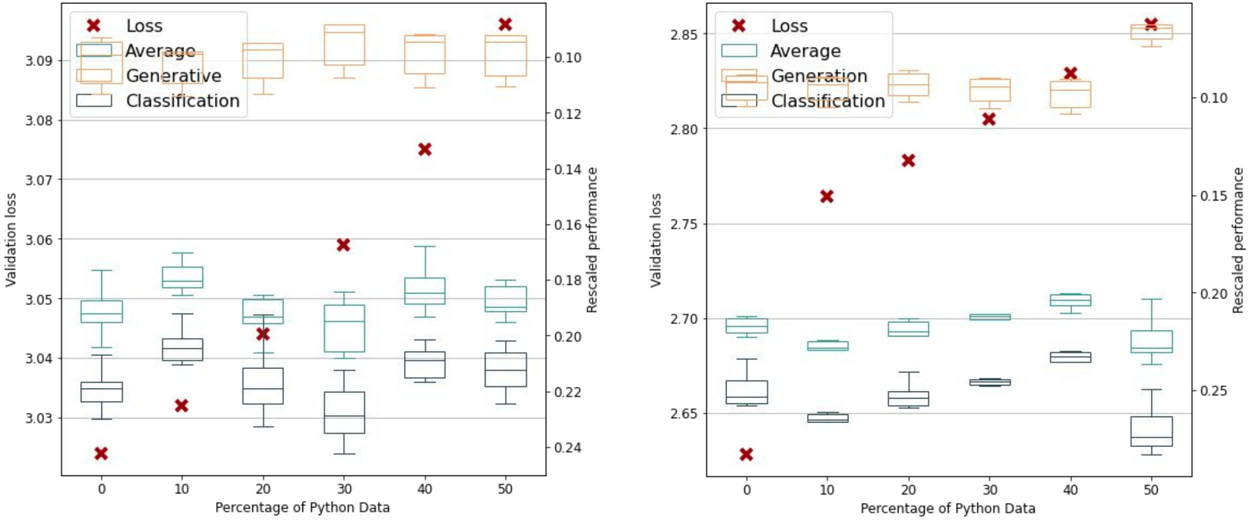
Not really…..

You can probably get away with x5-8



Adding code data

50% Python data: no difference



Would it work for non-en languages?

Multilinguality: when does it help?

Scaling Laws for Generative Mixed-Modal Language Models

Armen Aghajanyan[*,†], **Lili Yu**[*,†], **Alexis Conneau**[†], **Wei-Ning Hsu**[†]

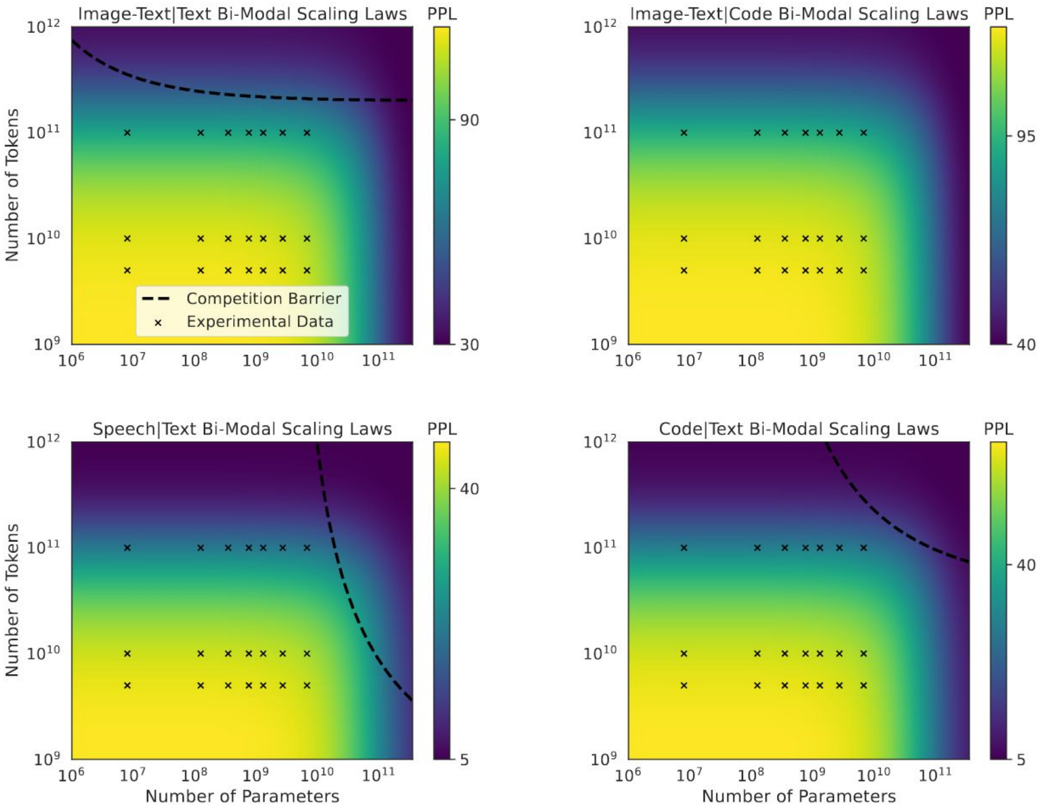
Karen Hambardzumyan[◇], **Susan Zhang**[†], **Stephen Roller**[†], **Naman Goyal**[†]

Omer Levy[†] **& Luke Zettlemoyer**[†,♡]

FAIR[†], University of Washington[♡], YerevaNN[◇]
armenag@meta.com

Abstract

Generative language models define distributions over sequences of tokens that can represent essentially any combination of data modalities (e.g., any permutation of image tokens from VQ-VAEs, speech tokens from HuBERT, BPE tokens for language or code, and so on). To better understand the scaling properties of such mixed-modal models, we conducted over 250 experiments using seven different modalities and model sizes ranging from 8 million to 30 billion, trained on 5-100 billion tokens. We report new mixed-modal scaling laws that unify the contributions of individual modalities and the interactions between them. Specifically, we explicitly model the optimal synergy and competition due to data and model size as an additive term to previous uni-modal scaling laws. We also find four empirical phenomena observed during the training, such as emergent coordinate-ascent style training that naturally alternates between modalities, guidelines for selecting critical hyper-parameters, and connections between mixed-modal competition and training stability. Finally, we test our scaling law by training a 30B speech-text model, which significantly outperforms the corresponding unimodal models. Overall, our research provides valuable insights into the design and training of mixed-modal generative models, an important new class of unified models that have unique distributional properties.



Multilinguality: when does it help?

Different languages ~ very close modalities

Find close languages, and compute competition barrier

Or even start from a code or other language model, then fine-tune it, the same way ChatGPT is descended from a Python LM

RLHF

Instruct, Sparrow: ~50k human annotations

As easy in any language as in English

Questions?