IMPERIAL

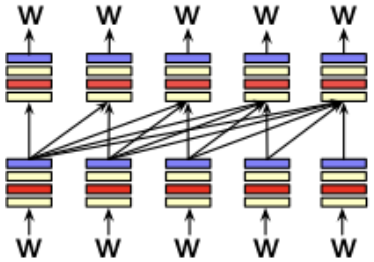
Pre-training Strategies (2)

09/12/2025

Shamsuddeen Muhammad
Google DeepMind Academic Fellow,
Imperial College London
https://shmuhammadd.github.io/

Idris Abdulmumin
Postdoctoral Research Fellow,
DSFSI, University of Pretoria
https://abumafrim.github.io/

Decoders



What most people think of when we say LLM

- GPT, Claude, Llama, DeepSeek, Mistral
- A generative model
- It takes as input a series of tokens, and iteratively generates an output token one at a time.
- Left to right (causal, autoregressive)

The progress on transformer **decoder** models has been spearheaded to a large extent by OpenAI. These models are exceptionally good at predicting the next word in a sequence and are thus mostly used for text generation tasks.

Their progress has been fueled by using **larger datasets** and **scaling the language models to larger and larger sizes**.

Decoders Model

Decoder-only architectures (e.g., GPT models) use **causal self-attention**, meaning:

- Each word can only attend to **previous** words.
- It cannot look at future tokens.
- The model processes the sequence *from left to right*.

This perfectly matches the structure of a language model.

Decoders Model: Causal Modeling

Decoder-only transformers apply a **triangular attention mask**:

Token t can attend only to 1,2,...,t.Token t can attend only to 1,2,...,t.

This mask blocks all future positions:

- token 1 → sees only itself
- token 2 → sees token 1
- token 3 → sees tokens 1, 2
- …and so on.

Thus, the model **never cheats** by looking at the answer.

Decoders Model: Causal Modeling

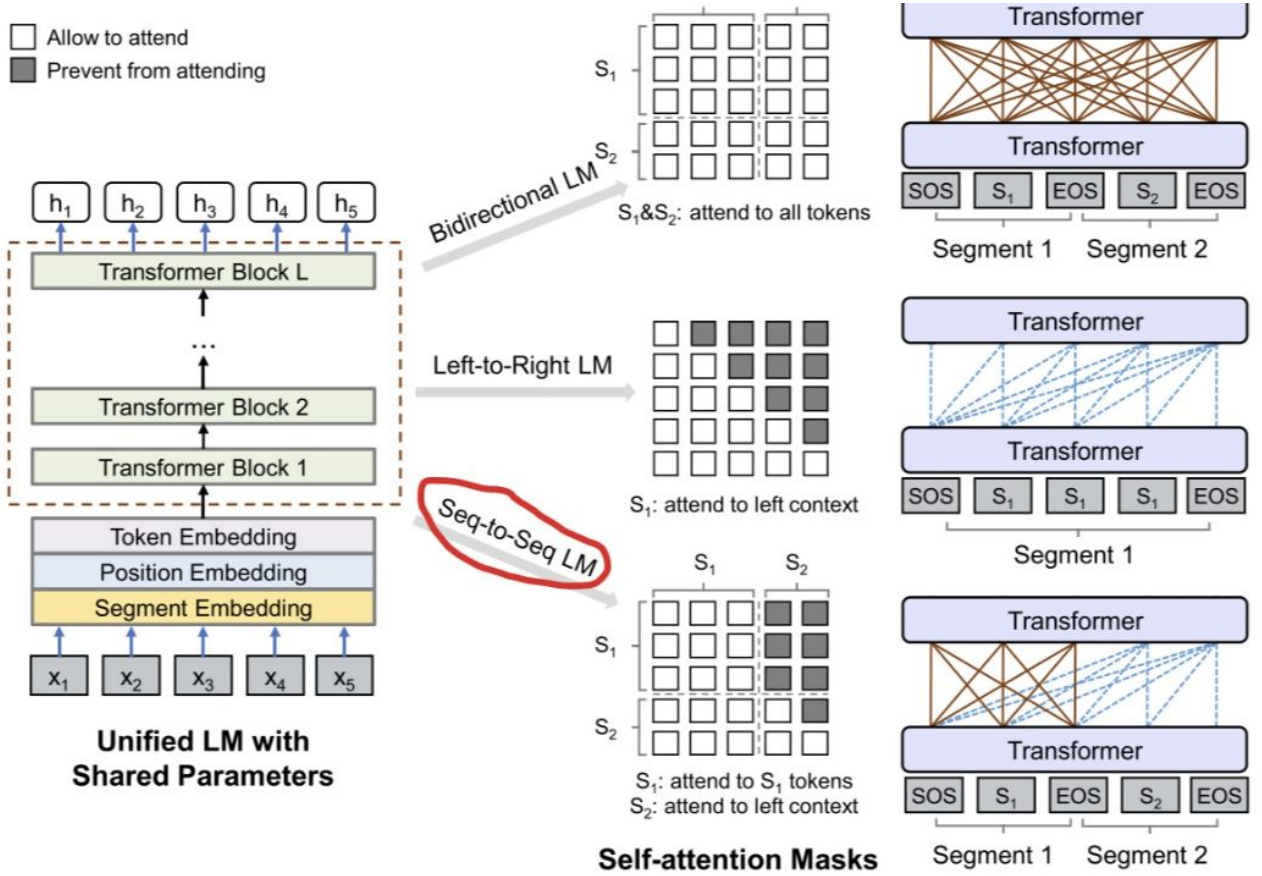
Triangular attention mask:

Token t can attend only to 1,2,...,t.Token t can attend only to 1,2,...,t.

This mask blocks all future positions:

- token 1 → sees only itself
- token 2 → sees token 1
- token 3 → sees tokens 1, 2
- …and so on.

Thus, the model **never cheats** by looking at the answer.



Why this enables language modeling

Language modeling requires predicting the next word:

$$p(w_t \mid w_1, \ldots, w_{t-1})$$

Decoder-only models naturally satisfy this constraint because:

- they process tokens sequentially,
- they never look ahead,
- they generate text one token at a time.

This allows them to be trained with a simple objective:

That is, predict the next token given previous ones.

Decoder-only models can perform language modeling because they use causal self-attention, which prevents tokens from seeing future words and allows them to predict the next token in a left-to-right manner.

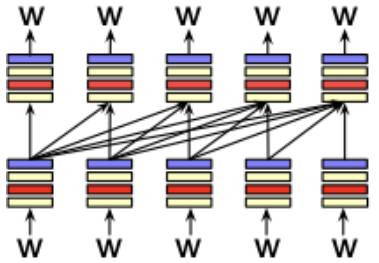
Decoders Model

Decoder-only architectures (e.g., GPT models) use **causal self-attention**, meaning:

- Each word can only attend to **previous** words.
- It cannot look at future tokens.
- The model processes the sequence *from left to right*.

This perfectly matches the structure of a language model.

Summary: Decoder-Only (e.g., GPT, Llama)



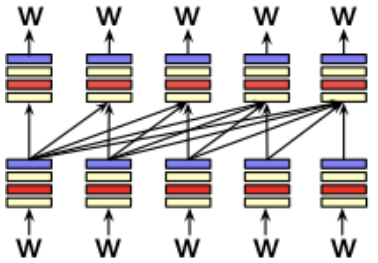
Architecture

- A stack of Transformer decoders only.
- Uses causal (left-to-right) attention: predicts the next word without seeing future words.

Pretraining Strategy

- Autoregressive Language Modeling (Next-Token Prediction)
  - Predict the next word in a sequence.
  - Trained on massive text corpora.

Summary: Decoder-Only (e.g., GPT, Llama)



Strengths

- Extremely strong generative capabilities:
  - Coherent long-form text
  - Dialogue
  - Code generation
  - Story continuation
- Scales very well; simple pretraining objective.

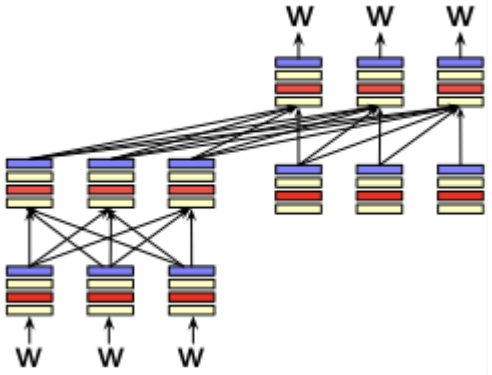
Limitations

- Does not use future context, so weaker on pure “understanding” tasks compared to BERT-like models unless fine-tuned.
- Requires larger scale for strong performance.

Examples

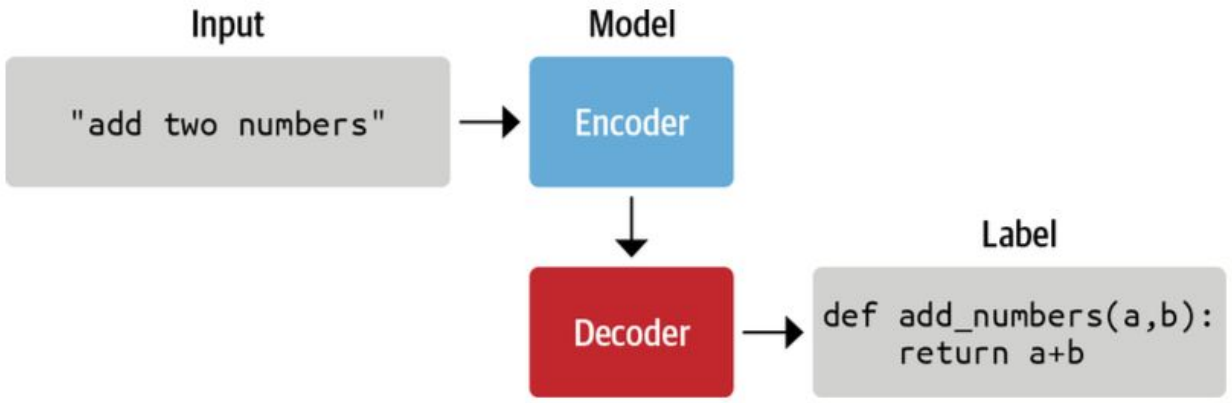
Decoder-Only Models
GPT (1-4, Neo, J)
LLaMA (1,2)
Mistral/Mixtral
PaLM (1,2)
Falcon
BLOOM
Phi-2
Command R+
Orca (2)
Yi
Zephyr
Gemma
Claude (1,2)
Grok
CodeGen
Codex
StarCoder
WizardLM
OpenChat

Encoder-Decoders

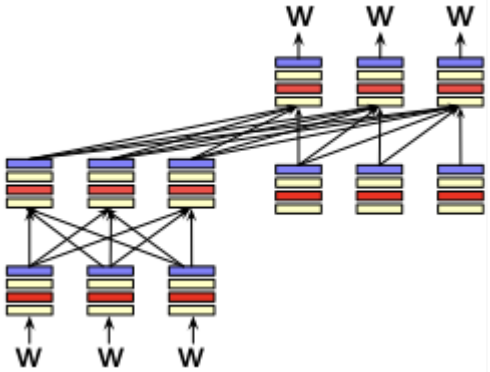


Trained to map from one sequence to another Very popular for:

i. machine translation (map from one language to another)
ii. speech recognition (map from acoustics to words)



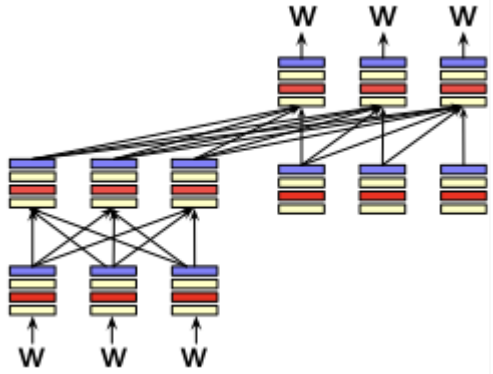
Encoder-Decoders



Architecture

- A Transformer encoder + a Transformer decoder.
- Encoder processes the full input bidirectionally.
- Decoder generates output autoregressively (left-to-right).

Encoder-Decoders



Strengths

- Combines the best of both worlds:
  - Encoder: strong comprehension
  - Decoder: strong generation
- Excellent for tasks requiring transformation of text:
  - Translation
  - Summarization
  - Paraphrasing
  - Question answering
  - Any seq-to-seq task

Limitations

- More expensive (two components).
- Pretraining objectives are more complex

Examples

Encoder-Decoder Models
T5
mT5
BART
mBART
PEGASUS
FLAN-T5
ByT5
ProphetNet
MarianMT
ViLT
GIT
OFA
Flamingo
PaLI
Kosmos
Pix2Seq
BLIP (2)
VideoBLIP

When to Use Encoder-Only, Decoder-Only, and Encoder–Decoder Models

Encoder-Only Models

Examples: BERT, RoBERTa, ELECTRA, DeBERTa, **Architecture:** Bidirectional transformer encoder only (no autoregressive decoder).

Best for:

Understanding tasks (not generation)

- Text classification (sentiment, hate speech, topic classification)
- Named entity recognition (NER)
- Part-of-speech tagging
- Semantic similarity / retrieval
- Question-answering when output is short/extractive
- Text embeddings for downstream models

Why?

- Full bidirectional context → strong representations
- Good at *encoding* meaning, but not designed to *generate* text
- Uses masked language modeling, not next-token prediction

Not good for: Long-form generation, Dialogue , Machine translation (modern SOTA), Open-ended tasks

Decoder-Only Models

Examples: GPT-2/3/4, LLaMA, Mistral, Gemma, ChatGPT , **Architecture:** Autoregressive transformer predicting next token.

Best for:

Text generation tasks

- Chatbots, dialogue systems
- Story generation
- Code generation
- Reasoning, chain-of-thought
- Instruction following (after tuning & RLHF)

General foundation models

- Can perform classification or analysis *via prompting*
- Extremely flexible zero-shot & few-shot learners

Why?

- Trained on next-token prediction → excels at long-form coherent output
- Prompting turns every task into a sequence-generation task
- Scaling laws favor decoder-only for massive models

Decoder-Decoder Models

Examples: T5, BART, mT5, original Transformer (Vaswani et al.) , **Architecture:** Encoder reads the input/Decoder generates the output with cross-attention to the encoder

Best for:

Sequence-to-sequence transformation tasks

- Machine translation
- Summarization
- Paraphrasing
- Text simplification
- Data-to-text generation
- Question-answering (generative style)

Why?

- Input has full bidirectional encoding
- Decoder attends to structured representation → strong mapping
- Very efficient when input ≠ output (different lengths)

Large Language Model

GPT

Generative Pre-trained Transformer — a family of large language models (LLMs) developed primarily by **OpenAI** and built on the Transformer

Component	Meaning
Generative	GPT generates new content such as text, code, etc.
Pre-trained	Initially trained on a large dataset in an unsupervised manner before task-specific fine-tuning.
Transformer	Based on the attention-based Transformer architecture.

Version	Parameters	Released By	Key Features
GPT-1	117M	OpenAI (2018)	Proof of concept for unsupervised language learning
GPT-2	1.5B	OpenAI (2019)	Capable of generating coherent paragraphs and performing multiple NLP tasks
GPT-3	175B	OpenAI (2020)	Achieved impressive zero-shot and few-shot learning
GPT-3.5	13B–175B?	OpenAI (2022)	Backbone of ChatGPT with improved dialogue capabilities
GPT-4	???	OpenAI (2023)	Multimodal inputs, improved reasoning and factual accuracy
GPT-4 Turbo	???	OpenAI (2023)	Faster and more efficient variant of GPT-4

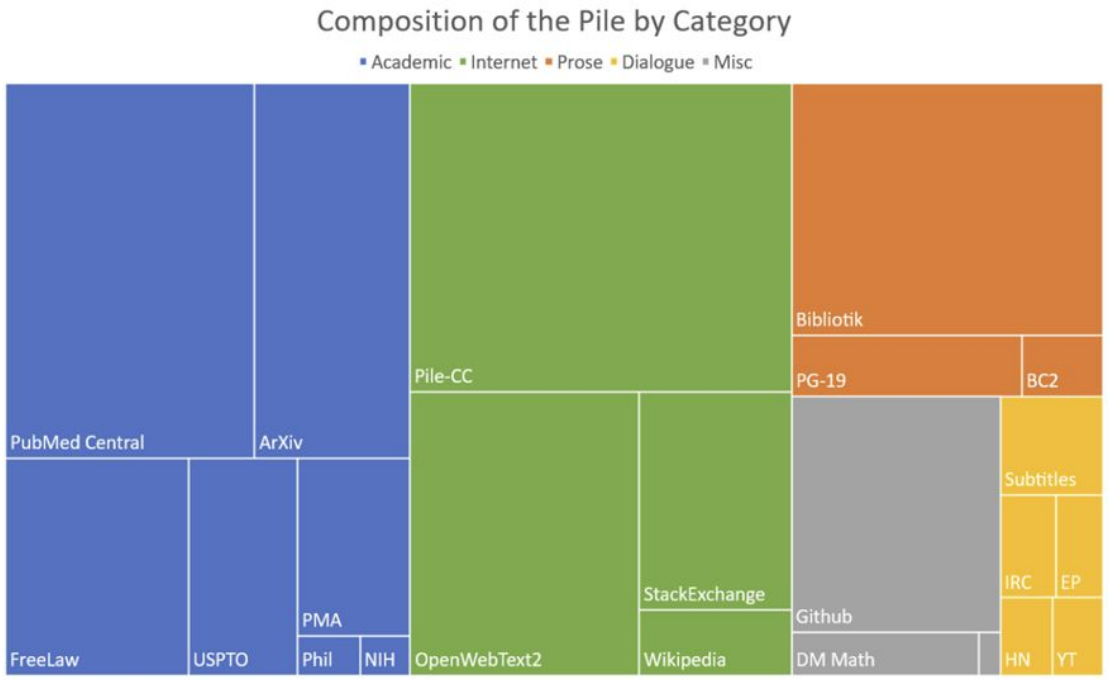
Generative Pretrained Transformer (GPT)

2018's GPT was a big success in pretraining a decoder!

- Transformer decoder with 12 layers, 117M parameters.
- 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers.
- Byte-pair encoding with 40,000 merges
- Trained on BooksCorpus: over 7000 unique books.
  - Contains long spans of contiguous text, for learning long-distance dependencies.

Pretraining can be massively diverse

- It's not just about the quantity, but also the incredible *diversity* of internet text data



[Gao+ 20]

Source	Doc Type	UTF-8 bytes *(GB)*	Documents *(millions)*	Unicode words *(billions)*	Llama tokens *(billions)*
Common Crawl	web pages	9,812	3,734	1,928	2,479
GitHub	code	1,043	210	260	411
Reddit	social media	339	377	72	89
Semantic Scholar	papers	268	38.8	50	70
Project Gutenberg	books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	encyclopedic	16.2	6.2	3.7	4.3
Total		**11,519**	**4,367**	**2,318**	**3,059**

[Soldani+ 24]

Pre-Training Datasets

Dataset	Size	Availability	Web	CC Processing
C4	~360GT	Public	100%	Rules + NSFW words blocklist
OSCAR-21.09	~370GT	Public	100%	Built at the line-level
OSCAR-22.01	~283GT	Public	100%	Line-level rules + optional rules & NSFW URL blocklist
GPT-3	300GT	Private	60%	Content filter trained on known high-quality sources
The Pile	~340GT	Public	18%	jusText for extraction, content filter trained on curated data
PaLM	780GT	Private	27%	Filter trained on HQ data
RefinedWeb	~5,000GT	Public (600GT)	100%	trafilatura for extraction, document and line-level rules, NSFW URL blocklist

Source	Disk Size	Documents	Tokens	Sampling Proportion
MassiveWeb	1.9 TB	604M	506B	48%
Books	2.1 TB	4M	560B	27%
C4	0.75 TB	361M	182B	10%
News	2.7 TB	1.1B	676B	10%
GitHub	3.1 TB	142M	422B	3%
Wikipedia	0.001 TB	6M	4B	2%

Source	Kind	Gzip files (GB)	Documents (millions)	Tokens (billions)
Common Crawl	web	4,197	4,600	2,415
C4	web	302	364	175
peS2o	academic	150	38.8	57
The Stack	code	675	236	430
Project Gutenberg	books	6.6	0.052	4.8
Wikipedia, Wikibooks	encyclopedic	5.8	6.1	3.6
Total		5,334	5,245	3,084

https://medium.com/@jelkhoury880/how-have-pre-training-datasets-for-large-language-models-evolved-13d74c01f8e8

Example: Llama1 Pre-training Data Mixture

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

1.4 Trillion Tokens!

How Large are 1T Tokens?

Physical Size (if printed)

- Average words per page: A typical page contains about 300-500 words.
- Words from 1 trillion tokens: Assuming 750 billion words, and an average of 400 words per page:
  - Total Pages: Approximately 1.875 billion pages.

Digital Storage

- Character Encoding: Assuming each character takes up 1 byte (in a simple encoding like ASCII), 1 trillion tokens (4 trillion characters) would require about 4 terabytes (TB) of storage.

Reading Time

- Reading Speed: The average reading speed is about 200-250 words per minute.
- Time to Read 750 Billion Words: At 200 words per minute, it would take about 3.75 billion minutes, or approximately 7,125 years of continuous reading.

Fair Use and Concern

Google swallows 11,000 novels to improve AI's conversation

As writers learn that tech giant has processed their work without permission, the Authors Guild condemns 'blatantly commercial use of expressive authorship'

'It doesn't harm the authors' ... Google's headquarters in Mountain View, California. Photograph: Marcio Jose Sanchez/AP

Arts and Humanities, Law, Regulation, and Policy, Machine Learning

Reexamining "Fair Use" in the Age of AI

Generative AI claims to produce new language and images, but when those ideas are based on copyrighted material, who gets the credit? A new paper from Stanford University looks for answers.

Jun 5, 2023 | Andrew Myers

Data is stolen everywhere

BUSINESS INSIDER AFRICA

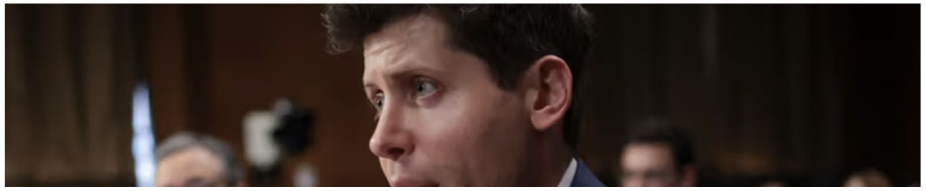
MARKETS LEADERS CAREERS LIFESTYLE

News

A lawsuit claims OpenAI stole 'massive amounts of personal data,' including medical records and information about children, to train ChatGPT

GRACE DEAN | 29 June 2023 01:42 PM

OpenAI stole "massive amounts of personal data" to train ChatGPT, a lawsuit alleges.



CNN Business Markets Tech Media Calculators Videos Wat

NASDAQ 17,299.29 0.14%

White House Correspondents Associ

Google hit with lawsuit alleging it stole data from millions of users to train its AI tools

By Catherine Thorbecke, CNN

4 minute read · Updated 8:48 AM EDT, Wed July 12, 2023

DETECTING PRETRAINING DATA FROM LARGE LANGUAGE MODELS

Weijia Shi[1] * **Anirudh Ajith**[2]* **Mengzhou Xia**[2] **Yangsibo Huang**[2]
Daogao Liu[1] **Terra Blevins**[1] **Danqi Chen**[2] **Luke Zettlemoyer**[1]
[1]University of Washington [2]Princeton University
swj0419.github.io/detect-pretrain.github.io

ABSTRACT

Although large language models (LLMs) are widely deployed, the data used to train them is rarely disclosed. Given the incredible scale of this data, up to trillions of tokens, it is all but certain that it includes potentially problematic text such as copyrighted materials, personally identifiable information, and test data for widely reported reference benchmarks. However, we currently have no way to know which data of these types is included or in what proportions. In this paper, we study the pretraining data detection problem: *given a piece of text and black-box access to an LLM without knowing the pretraining data, can we determine if the model was trained on the provided text?* To facilitate this study, we introduce a dynamic benchmark WIKIMIA that uses data created before and after model training to support gold truth detection. We also introduce a new detection method MIN-K% PROB based on a simple hypothesis: an unseen example is likely to contain a few outlier words with low probabilities under the LLM, while a seen example is less likely to have words with such low probabilities. MIN-K% PROB can be applied without any knowledge about the pretraining corpus or any additional training, departing from previous detection methods that require training a reference model on data that is similar to the pretraining data. Moreover, our experiments demonstrate that MIN-K% PROB achieves a 7.4% improvement on WIKIMIA over these previous methods. We apply MIN-K% PROB to three real-world scenarios, copyrighted book detection, contaminated downstream example detection and privacy auditing of machine unlearning, and find it a consistently effective solution.

Data, Data Everywhere: A Guide for Pretraining Dataset Construction

Jupinder Parmar*, **Shrimai Prabhumoye**, **Joseph Jennings**,
Bo Liu, **Aastha Jhunjhunwala**, **Zhilin Wang**, **Mostofa Patwary**,
Mohammad Shoeybi , **Bryan Catanzaro**
NVIDIA

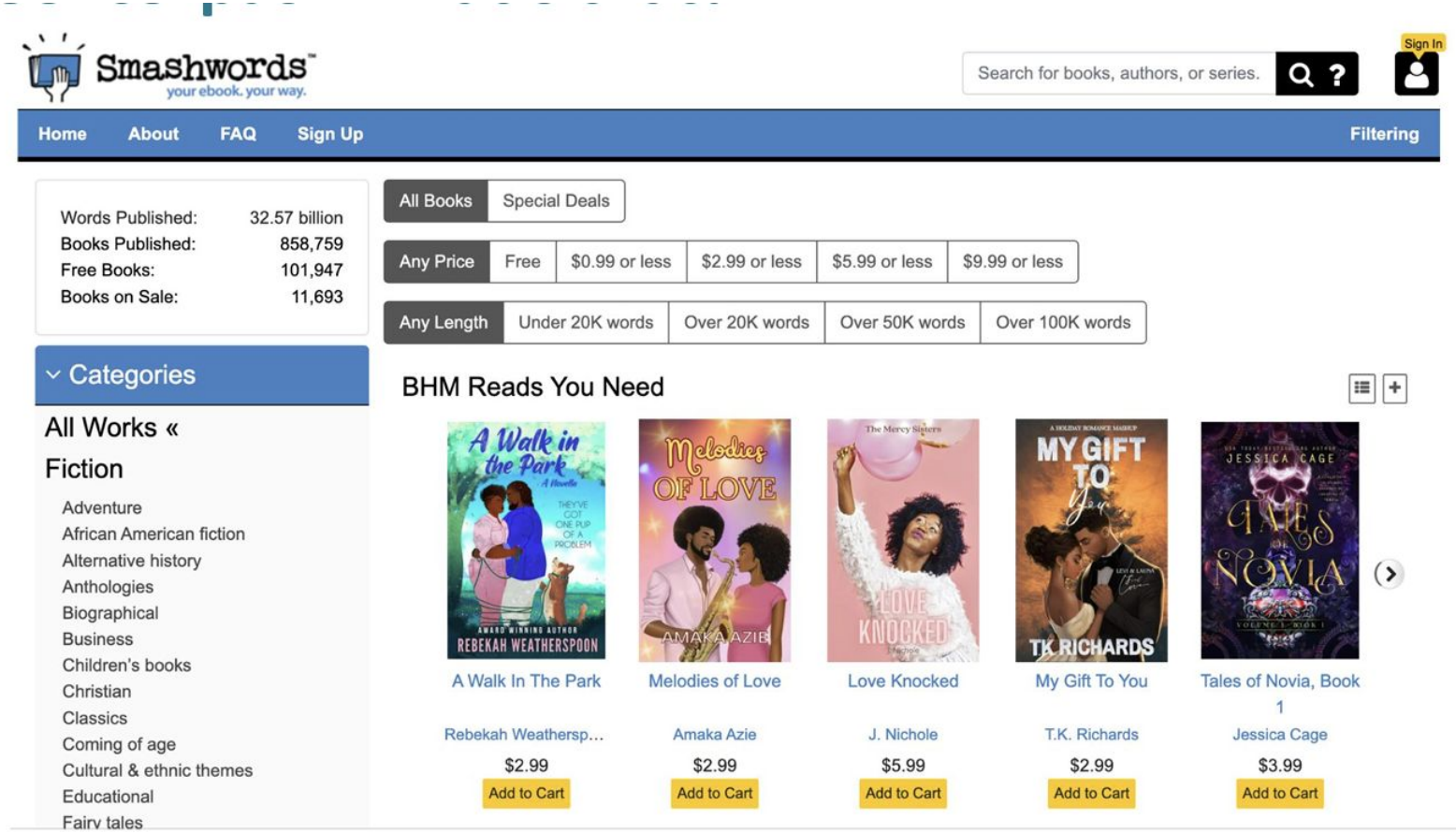
Abstract

The impressive capabilities of recent language models can be largely attributed to the multi-trillion token pretraining datasets that they are trained on. However, model developers fail to disclose their construction methodology which has lead to a lack of open information on how to develop effective pretraining sets. To address this issue, we perform the first systematic study across the entire pipeline of pretraining set construction. First, we run ablations on existing techniques for pretraining set development to identify which methods translate to the largest gains in model accuracy on downstream evaluations. Then, we categorize the most widely used data source, web crawl snapshots, across the attributes of toxicity, quality, type of speech, and domain. Finally, we show how such attribute information can be used to further refine and improve the quality of a pretraining set. These findings constitute an actionable set of steps that practitioners can use to develop high quality pretraining sets.

Most leading models (OpenAI, 2024; Team, 2024b; Anthropic, 2024; Jiang et al., 2023) do not divulge what methods were used to go from raw data sources to a final pre-training set. Other models document only small sections of their process (Touvron et al., 2023b; Parmar et al., 2024; Bai et al., 2023; Team et al., 2024) and lack information on why or how the chosen decisions were made. The scarcity of open knowledge in this area hinders the general community from contributing to the advancement of model capabilities (Rogers, 2021).

The steps in pretraining set construction are shown in Figure 1: the pipeline starts with a collection of text data sources, removes ill-formed and duplicate documents during data curation, further filters out low-quality documents via data selection, and finally assigns sampling weights to determine the prevalence of each data source during training. Recent works (Longpre et al., 2023; Penedo et al., 2023; Soldaini et al., 2024; Penedo et al., 2024) have started to elucidate strategies for effective pre-

BookCorpus



Scraped ebooks from the internet – highly controversial

Correcting FLORES Evaluation Dataset for Four African Languages

Idris Abdulmumin[1*+], **Sthembiso Mkhwanazi**[2], **Mahlatse S. Mbooi**[2], **Shamsuddeen Hassan Muhammad**[3*+], **Ibrahim Said Ahmad**[4*+], **Neo Putini**[5], **Miehleketo Mathebula**[1], **Matimba Shingange**[1], **Tajuddeen Gwadabe**[*+], **Vukosi Marivate**[1,6]

[1]Data Science for Social Impact, University of Pretoria, [2]Council for Scientific and Industrial Research, South Africa, [3]Imperial College, London, [4]Northeastern University, [5]University of KwaZulu-Natal, [6]Lelapa AI, [*]HausaNLP, [+]MasakhaneNLP

correspondence: idris.abdulmumin@up.ac.za

Abstract

This paper describes the corrections made to the FLORES evaluation (dev and devtest) dataset for four African languages, namely Hausa, Northern Sotho (Sepedi), Xitsonga, and isiZulu. The original dataset, though groundbreaking in its coverage of low-resource languages, exhibited various inconsistencies and inaccuracies in the reviewed languages that could potentially hinder the integrity of the evaluation of downstream tasks in natural language processing (NLP), especially machine translation. Through a meticulous review process by native speakers, several corrections were identified and implemented, improving the overall quality and reliability of the dataset. For each language, we provide a concise summary of the errors encountered and corrected and also present some statistical analysis that measures the difference between the existing and corrected datasets. We believe that our corrections improve the linguistic accuracy and reliability of the data and, thereby, contribute to a more effective evaluation of NLP tasks involving the four African languages. Finally, we recommend that future translation efforts, particularly in low-resource languages, prioritize the active involvement of native speakers at every stage of the process to ensure linguistic accuracy and cultural relevance.

the University of Pretoria's Data Science for Social Impact (DSFSI) Research Group, and other individual initiatives (Abdulmumin et al., 2022; Parida et al., 2023). For machine translation evaluation, the FLORES dataset (Goyal et al., 2021; NLLB Team et al., 2022) is widely accepted as a benchmark for evaluation, especially because it was the first of its kind for many languages and enables many-to-many evaluation, making it easier to evaluate say a Hausa to Sepedi translation system without pivoting through a high resource language, e.g., English. Recently, the MAFAND dataset (Adelani et al., 2022) was created, but it only allows bilingual evaluation and is limited to the news domain.

While all these resources are being developed, it is imperative to review them for validation to ensure that they meet the expected standard of accuracy and representation. A revealing work by Kreutzer et al. (2022), albeit on mostly web-crawled datasets, found that many of the datasets that are being relied upon for low-resource languages are littered with significant errors such as misalignments, incorrect translations, and other issues. The significance of evaluation datasets make them even more deserving of such reviews especially by literate native speakers that know how these languages are written and spoken. This pa-

Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets

Julia Kreutzer[a,b], **Isaac Caswell**[a], **Lisa Wang**[a], **Ahsan Wahab**[c], **Daan van Esch**[a], **Nasanbayar Ulzii-Orshikh**[d], **Allahsera Tapo**[b,e], **Nishant Subramani**[b,δ], **Artem Sokolov**[a], **Claytone Sikasote**[b,g], **Monang Setyawan**[h], **Supheakmungkol Sarin**[h], **Sokhar Samb**[b,i], **Benoît Sagot**[j], **Clara Rivera**[a], **Annette Rios**[k], **Isabel Papadimitriou**[l], **Salomey Osei**[b,m], **Pedro Ortiz Suarez**[j,n], **Iroro Orife**[b,o], **Kelechi Ogueji**[b,p], **Andre Niyongabo Rubungo**[b,q], **Toan Q. Nguyen**[r], **Mathias Müller**[k], **André Müller**[k], **Shamsuddeen Hassan Muhammad**[b,s], **Nanda Muhammad**[h], **Ayanda Mnyakeni**[h], **Jamshidbek Mirzakhalov**[c,t], **Tapiwanashe Matangira**[h], **Colin Leong**[b], **Nze Lawson**[h], **Sneha Kudugunta**[a], **Yacine Jernite**[b,u], **Mathias Jenny**[k], **Orhan Firat**[a,c], **Bonaventure F. P. Dossou**[b,v], **Sakhile Dlamini**[h], **Nisansa de Silva**[w], **Sakine Çabuk Ballı**[k], **Stella Biderman**[x], **Alessia Battisti**[k], **Ahmed Baruwa**[b,y], **Ankur Bapna**[a], **Pallavi Baljekar**[a], **Israel Abebe Azime**[b,i], **Ayodele Awokoya**[b,z], **Duygu Ataman**[c,k], **Orevaoghene Ahia**[b,α], **Oghenefego Ahia**[h], **Sweta Agrawal**[β], **Mofetoluwa Adeyemi**[b,γ],

[a]Google Research, [b]Masakhane NLP, [c]Turkic Interlingua, [d]Haverford College, [e]RobotsMali, [f]Intel Labs, [g]University of Zambia, [h]Google, [i]AIMS-AMMI, [j]Inria, [k]University of Zurich, [l]Stanford University, [m]Kwame Nkrumah University of Science and Technology, [n]Sorbonne Université, [o]Niger-Volta LTI, [p]University of Waterloo [q]University of Electronic Science and Technology of China, [r]University of Notre Dame, [s]Bayero University Kano, [t]University of South Florida, [u]Hugging Face, [v]Jacobs University Bremen, [w]University of Moratuwa, [x]EleutherAI, [y]Obafemi Awolowo University, [z]University of Ibadan, [α]Instadeep, [β]University of Maryland, [γ]Defence Space Administration Abuja, [δ]Allen Institute for Artificial Intelligence

Abstract

With the success of large-scale pre-training and multilingual modeling in Natural Language Processing (NLP), recent years have seen a proliferation of large, web-mined text datasets covering hundreds of languages. We manually audit the quality of 205 language-specific corpora released with five major public datasets (CCAligned, ParaCrawl, WikiMatrix, OSCAR, mC4). Lower-resource corpora have systematic issues: At least 15 corpora have no usable text, and a significant fraction contains less than 50% sentences of acceptable quality. In addition, many are mislabeled or use non-standard/ambiguous language codes. We demonstrate that these issues are easy to detect even for non-proficient speakers, and supplement the human audit with automatic analyses. Finally, we recommend techniques to evaluate and improve multilingual corpora and discuss potential risks that come with low-quality data releases.

1 Introduction

Access to multilingual datasets for NLP research has vastly improved over the past years. A variety of web-derived collections for hundreds of languages is available for anyone to download, such as ParaCrawl (Esplà et al., 2019; Bañón et al., 2020), WikiMatrix (Schwenk et al., 2021) CCAligned (El-Kishky et al., 2020), OSCAR (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020), and several others. These have in turn enabled a variety of highly multilingual models, like mT5 (Xue et al., 2021), M2M-100 (Fan et al., 2020), M4 (Arivazhagan et al., 2019).

Curating such datasets relies on the websites giving clues about the language of their contents (e.g. a language identifier in the URL) and on automatic language classification (LangID). It is commonly known that these automatically crawled and filtered datasets tend to have overall lower quality than hand-curated collec-

Pre-training data sample

Bizarro Wonder Woman is a bizarro version of Wonder Woman.\n\nWhen Bizarro III found himself infused with radiation from a blue sun, he developed the ability to replicate himself as well as create other \"Bizarro\" lifeforms based upon likenesses of people from Earth. He used this power to populate a cube-shaped planetoid dubbed Bizarro World within the blue sun star system. One of the many duplicates that he created was a Bizarro version of Wonder Woman. Bizarro Wonder Woman, working alongside her Bizarro confederates Batman, Flash, Green Lantern and Hawkgirl, sought to save Bizarro from Bizarro Doomsday by dropping their hyperbolic headquarters on top of him.\n\nAs opposed to her counterpart, Bizarro Wonder Woman uses a lasso that causes those ensnared to tell lies.\ ...

Generative Pretrained Transformer (GPT)

We mentioned how pretrained decoders can be used **in their capacities as language models.**

GPT-2, a larger version (1.5B) of GPT trained on more data, was shown to produce relatively convincing samples of natural language.

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

GPT-2

GPT-2 is identical to GPT-1, but:

- Has Layer normalization in between each sub-block
- Vocab extended to 50,257 tokens and context size increased from 512 to 1024
- Data: 8 million docs from the web (Common Crawl), minus Wikipedia

---

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

Pre-Training Cost (withGCP/AWS)

- **BERT:** Base $500, Large $7000
- **GPT-2** (as reported in other work): $25,000
- This is for a single pre-training run…developing new pre-training techniques may require many runs.
- Fine-tuning these models can typically be done with a single GPU (but may take 1-3 days for medium-sized datasets).

GPT-3, In-context learning, and very large models

So far, we've interacted with pretrained models in two ways:

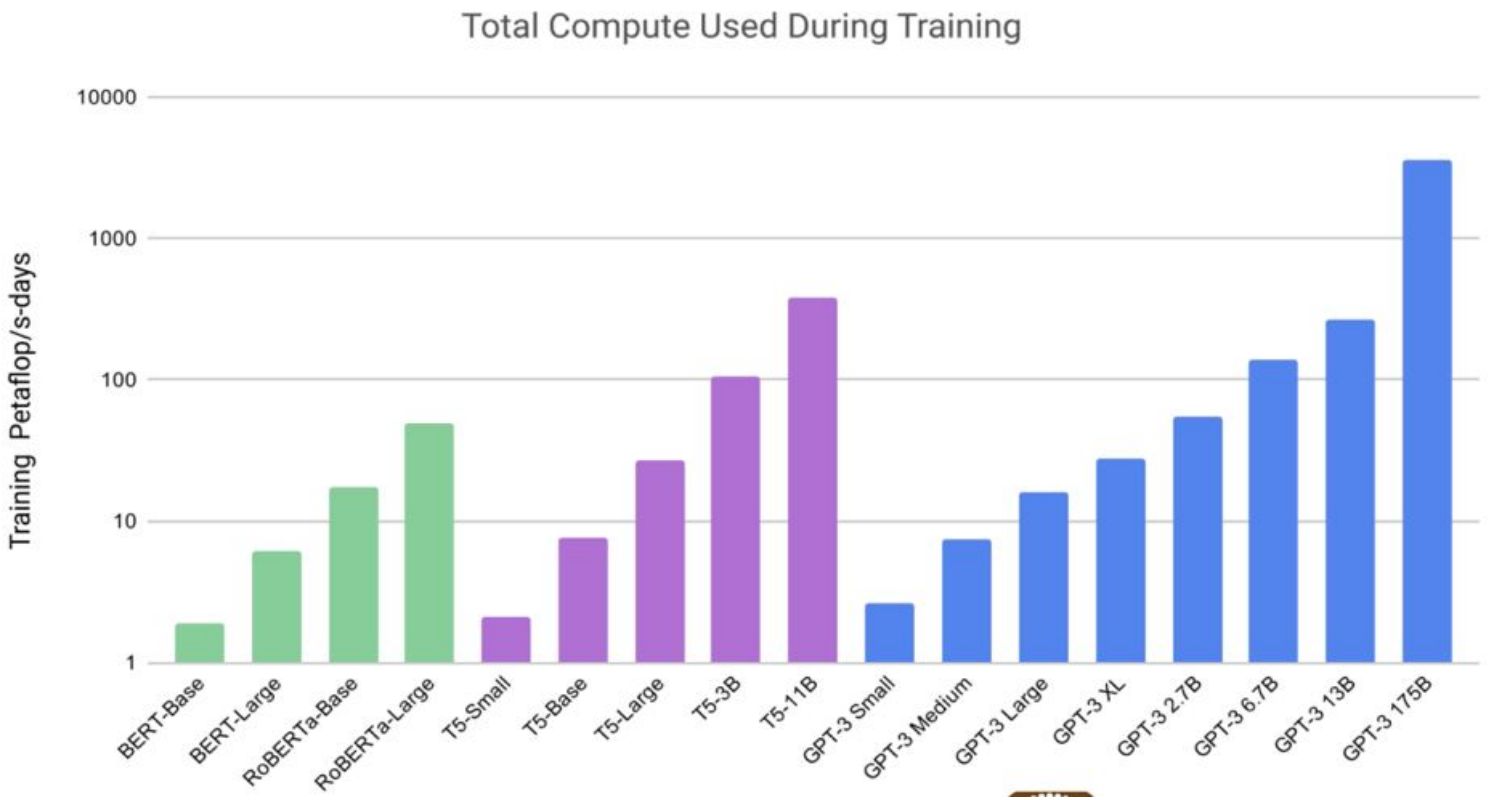
- Sample from the distributions they define (maybe providing a prompt)
- Fine-tune them on a task we care about, and take their predictions.

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.

GPT-3 is the canonical example of this. The largest T5 model had 11 billion parameters.

GPT-3 has 175 billion parameters.

GPT3



- **175B parameter model**
  - 96 layers, 96 heads, 12k-dim vectors
- Trained on Microsoft Azure, estimated to cost roughly **$10M**

GPT-3, In-context learning, and very large models

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.

The in-context examples seem to specify the task to be performed, and the conditional distribution mocks performing the task to a certain extent.

Input (prefix within a single Transformer decoder context):

“ thanks -> merci

hello -> bonjour

mint -> menthe

otter -> ”

Output (conditional generations):

loutre…”

GPT-4

GPT-4

Model	Usage
davinci-002	$0.0020 / 1K tokens

Model	Input	Output
gpt-4	$0.03 / 1K tokens	$0.06 / 1K tokens

- Transformer-based
  - The rest is …. mystery!
  - If we're going based on costs, GPT-4 is ~15-30 times costlier than GPT3. That should give you an idea how its likely size!

- Note, these language models involve more than just pre-training.
  - Pre-training provides the foundation based on which we build the model.
  - We will discuss the later stages next week.

Llama: A Family of Open-Source LLMs from Meta AI

- **Llama-1 + Llama-2**

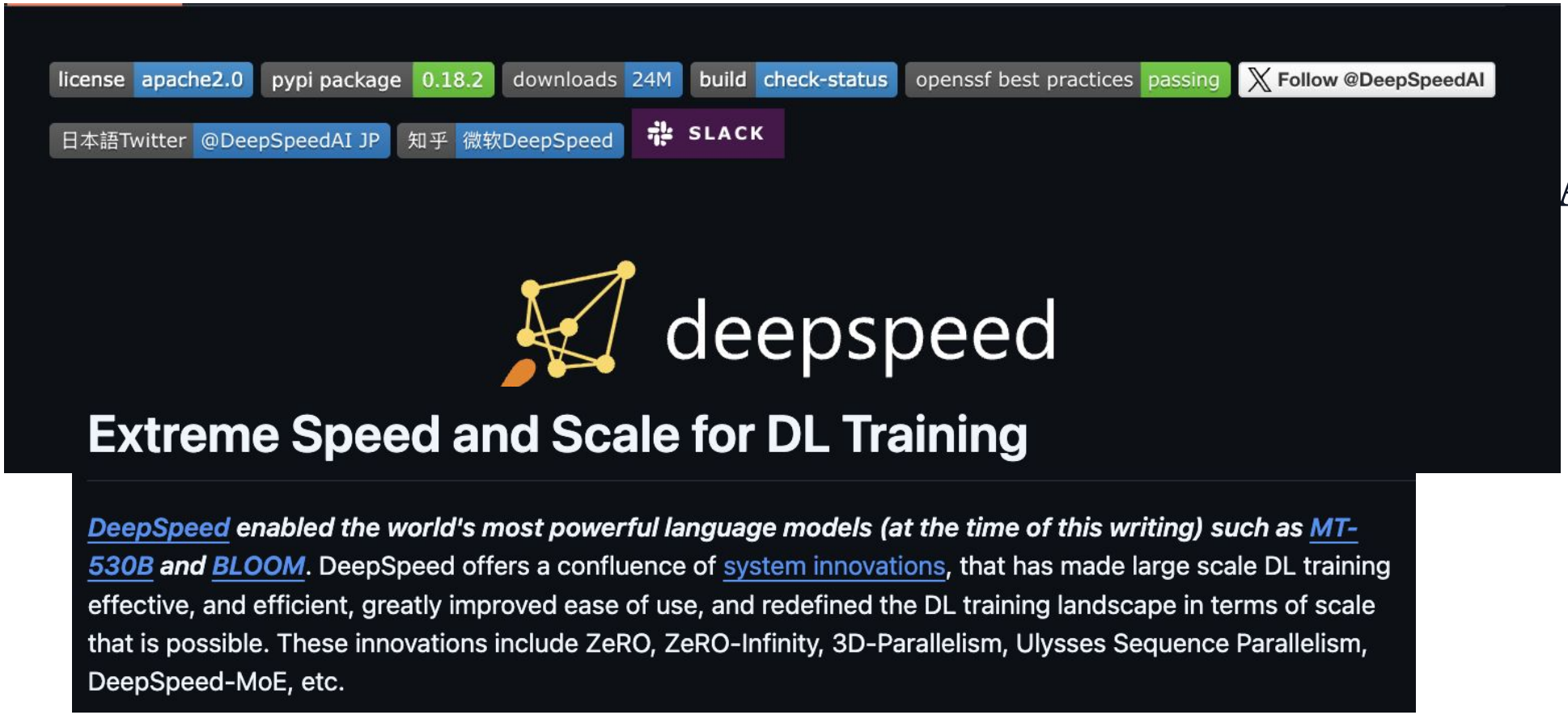
params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

- Models have mostly gotten smaller since GPT-3, but haven't changed much.
- **Tokenizer: Byte-Pair Encoding (BPE)** [Recall: we have already discussed this algorithm in lecture on 'Tokenization Strategies']
- **Rotary positional encodings**, a few other small architecture changes
- **Optimized mix of pre-training data**: Common Crawl, GitHub, Wikipedia, Books, etc.

How to train the model with a GPU cluster or multiple GPU nodes?

Training Library

- DeepSpeed is a deep learning optimization library that makes distributed training easy, efficient, and effective. It has been integrated into the Huggingface library.
- 

Training Library

- Megatron-LM is a large, powerful transformer model framework developed by the Applied Deep Learning Research team at NVIDIA.

Scaling Laws

Scaling Laws

"If you make the model bigger, give it more data and more compute, its performance will improve in a predictable way."

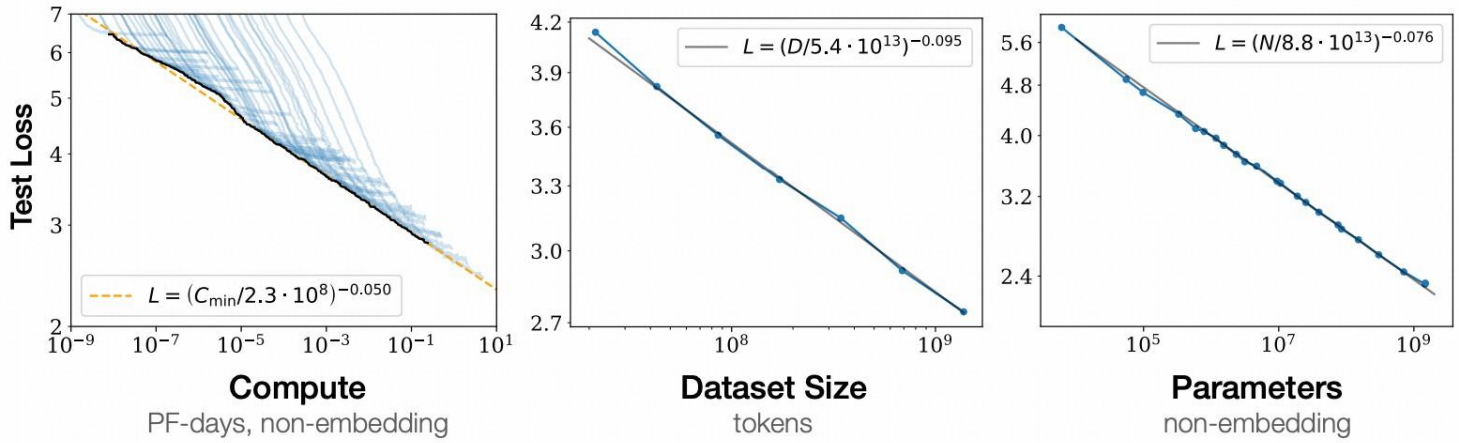


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

https://arxiv.org/pdf/2001.08361

Given constrained compute budget, what would be the optimal combination of model size and training data size (measured in number of tokens) that yields the lowest loss?

Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

Sam McCandlish*
OpenAI
sam@openai.com

Tom Henighan
OpenAI
henighan@openai.com

Tom B. Brown
OpenAI
tom@openai.com

Benjamin Chess
OpenAI
bchess@openai.com

Rewon Child
OpenAI
rewon@openai.com

Scott Gray
OpenAI
scott@openai.com

Alec Radford
OpenAI
alec@openai.com

Jeffrey Wu
OpenAI
jeffwu@openai.com

Dario Amodei
OpenAI
damodei@openai.com

Abstract

We study empirical scaling laws for language model performance on the cross-entropy loss. The loss scales as a power-law with model size, dataset size, and the amount of compute used for training, with some trends spanning more than seven orders of magnitude. Other architectural details such as network width or depth have minimal effects within a wide range. Simple equations govern the dependence of overfitting on model/dataset size and the dependence of training speed on model size. These relationships allow us to determine the optimal allocation of a fixed compute budget. Larger models are significantly more sample-efficient, such that optimally compute-efficient training involves training very large models on a relatively modest amount of data and stopping significantly before convergence.

Scaling Laws

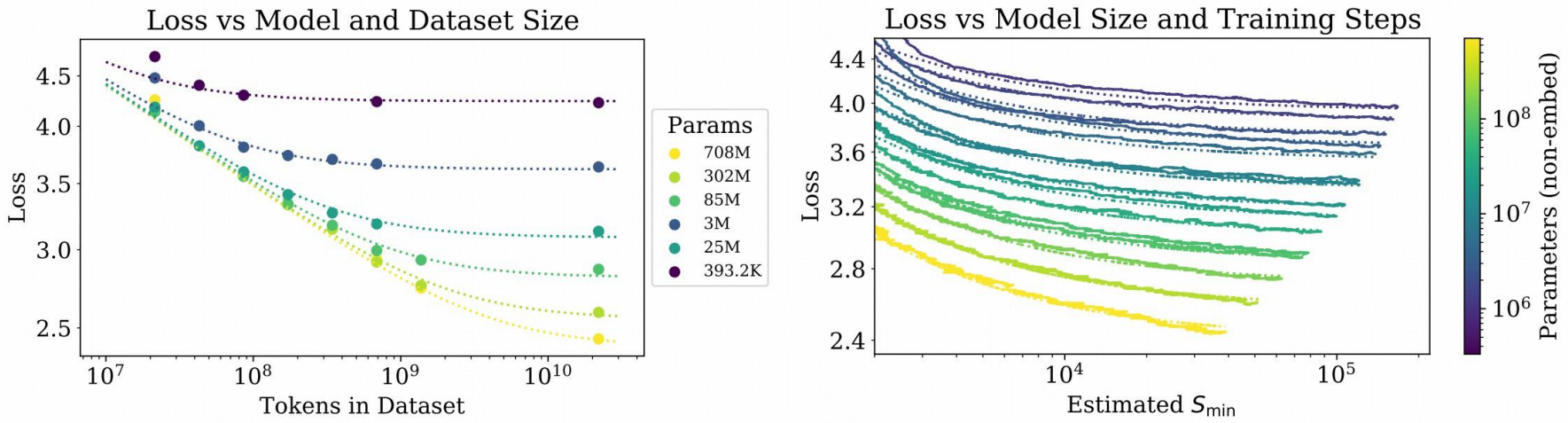


Figure 4 **Left**: The early-stopped test loss $L(N, D)$ varies predictably with the dataset size D and model size N according to Equation (1.5). **Right**: After an initial transient period, learning curves for all model sizes N can be fit with Equation (1.6), which is parameterized in terms of S_{min}, the number of steps when training at large batch size (details in Section 5.1).

Scaling Efficiency: how do we best use our compute

GPT-3 was **175B parameters** and trained on **300B** tokens of text.

Roughly, the cost of training a large transformer scales as **parameters*tokens**

Did OpenAI strike the right parameter-token data to get the best model? No.

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

This 70B parameter model is better than the much larger other models!

Open vs. Closed Access

Licenses and Permissiveness

- **Public domain, CC-0:** old copyrighted works and products of US government workers
- **MIT, BSD:** very few restrictions
- **Apache, CC-BY:** must acknowledge owner
- **GPL, CC-BY-SA:** must acknowledge and use same license for derivative works
- **CC-NC:** cannot use for commercial purposes
- **LLaMa, OPEN-RAIL:** various other restrictions
- **No License:** all rights reserved, but can use under fair use

Fair Use

- US **fair use** doctrine — can use copyrighted material in some cases
- A gross simplification:
  - **Quoting** a small amount of material → likely OK **Doesn't**
  - **diminish** commercial value → possibly OK Use for
  - **non-commercial** purposes → possibly OK

- Most data on the internet is copyrighted, so model training is currently done assuming fair use
- But there are lawsuits!

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

Why Restrict Model Access?

- **Commercial Concerns:** Want to make money from the models
- **Safety:** Limited release prevents possible misuse
- **Legal Liability:** Training models on copyrighted data is a legal/ethical gray area

Open/Closed Access

(e.g. Liang et al. 2022)

- **Weights:** open? described? closed?
- **Inference Code:** open? described? closed?
- **Training Code:** open? described? closed?
- **Data:** open? described? closed?

English-Centric Open Models

Birds-eye View

Open source/reproducible:

- **Pythia:** Fully open, many sizes/checkpoints
- **OLMo:** Possibly strongest reproducible model

Open weights:

- **LLaMa 1/2/3/3.1:** Most popular, heavily safety tuned
- **Mistral/Mixtral:** Strong and fast model, several European languages
- **Qwen:** Strong, more multilingual - particularly en/zh

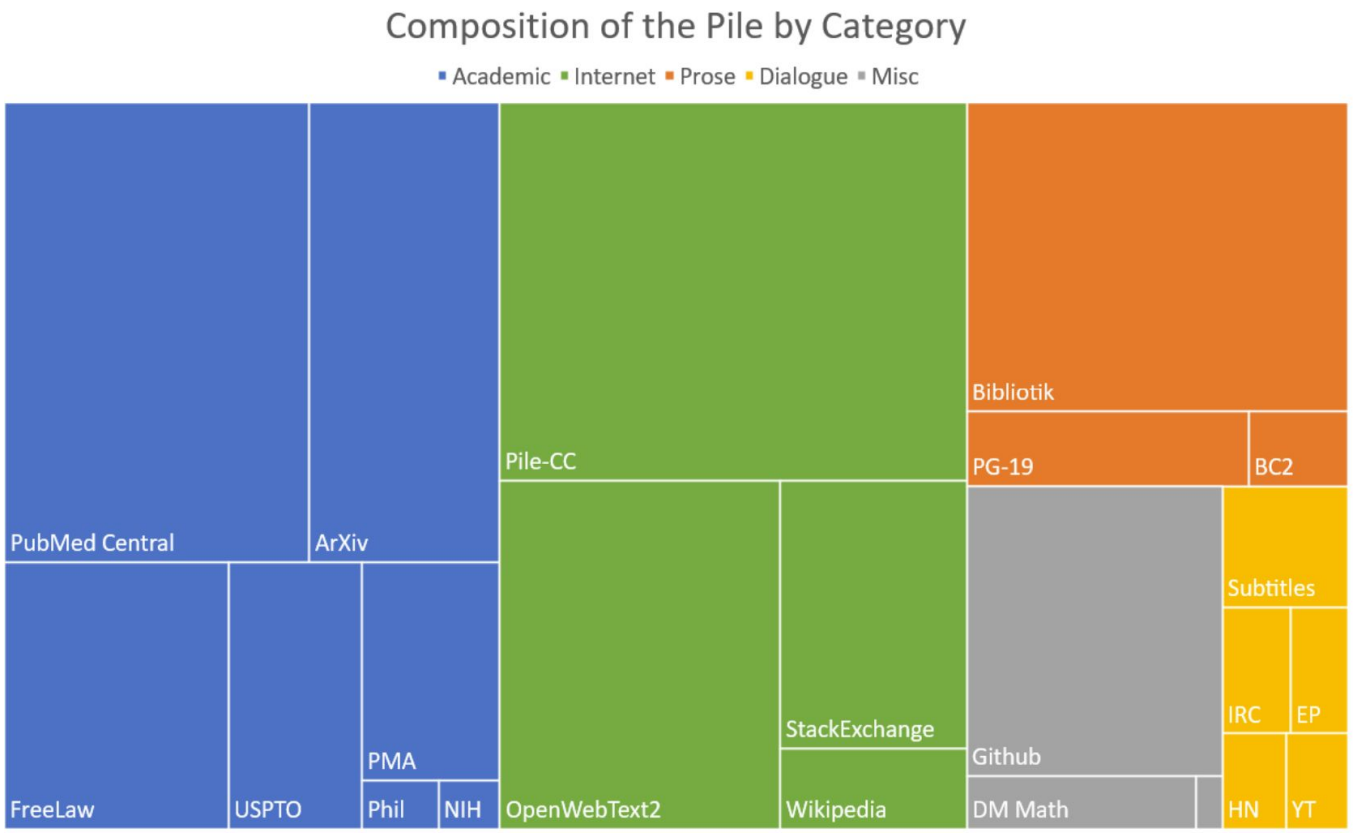
Pythia - Overview

- **Creator:** Eleuther AI
- **Goal:** Joint understanding of model training dynamics and scaling
- **Unique features:** 8 model sizes 70M-12B, 154 checkpoints for each

Arch	Transformer+RoPE+SwiGLU, context 2k (cf LLaMa 4k), parametric LN
Data	Trained on 300B tokens of The Pile (next slide), or deduped 207B
Train	LR scaled inversely to model size (7B=1.2e-4), batch size 2M tokens

The Pile

A now-standard 800GB dataset of lots of text/code



The Pile: An 800GB Dataset of Diverse Text for Language Modeling

Leo Gao **Stella Biderman** **Sid Black** **Laurence Golding**

Travis Hoppe **Charles Foster** **Jason Phang** **Horace He**

Anish Thite **Noa Nabeshima** **Shawn Presser** **Connor Leahy**

EleutherAI

contact@eleuther.ai

Abstract

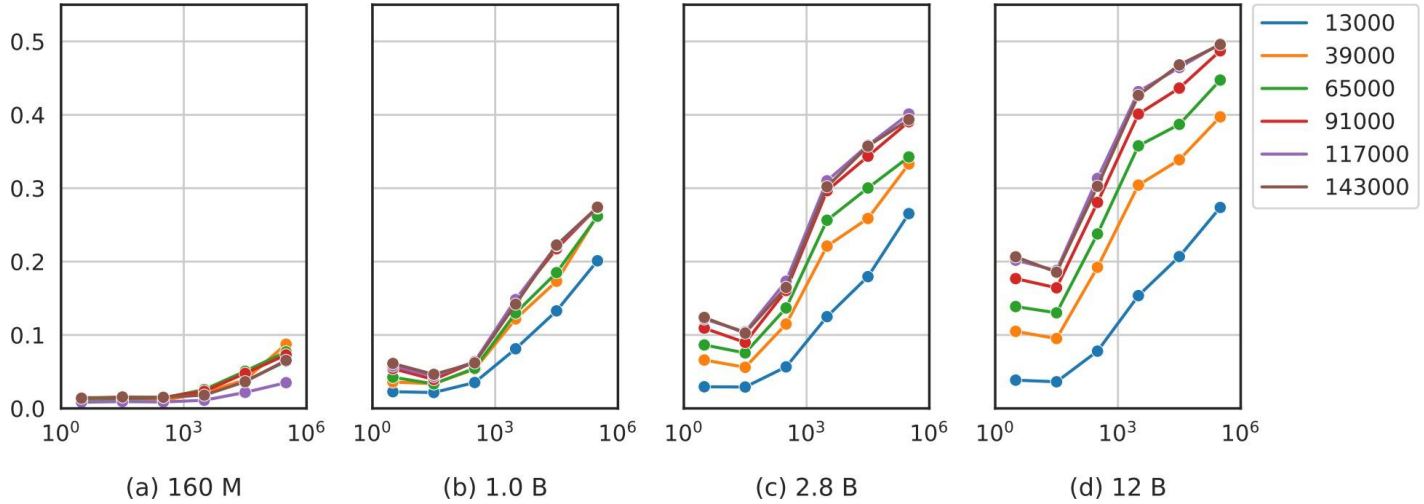
Recent work has demonstrated that increased training dataset diversity improves general cross-domain knowledge and downstream generalization capability for large-scale language models. With this in mind, we present *the Pile*: an 825 GiB English text corpus targeted at training large-scale language models. The Pile is constructed from 22 diverse high-quality subsets—both existing and newly constructed—many of which derive from academic or professional sources. Our evaluation of the untuned performance of GPT-2 and GPT-3 on the Pile shows that these models struggle on many of its components, such as academic writing. Conversely, models trained on the Pile improve significantly over both Raw CC and CC-100 on all components of the Pile, while improving performance on downstream evaluations. Through an in-depth exploratory analysis, we document potentially concerning aspects of the data for prospective users. We make publicly available the code used in its construction.[1]

versity leads to better downstream generalization capability (Rosset, 2019). Additionally, large-scale language models have been shown to effectively acquire knowledge in a novel domain with only relatively small amounts of training data from that domain (Rosset, 2019; Brown et al., 2020; Carlini et al., 2020). These results suggest that by mixing together a large number of smaller, high quality, diverse datasets, we can improve the general cross-domain knowledge and downstream generalization capabilities of the model compared to models trained on only a handful of data sources.

To address this need, we introduce the Pile: a 825.18 GiB English text dataset designed for training large scale language models. The Pile is composed of 22 diverse and high-quality datasets, including both established natural language processing datasets and several newly introduced ones. In addition to its utility in training large language models, the Pile can also serve as a broad-coverage benchmark for cross-domain knowledge and generalization ability of language models.

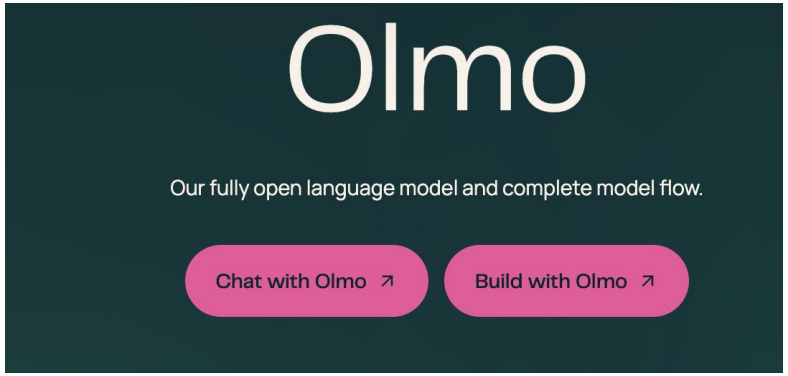
Pythia - Findings

- Some insights into training dynamics, e.g. larger models memorize facts more quickly (x axis: fact frequency, legend: training step)



- It is possible to intervene on data to reduce gender bias

OLMo - Overview



- **Creator:** Ai2 Allen Institute for AI
- **Goal:** Better science of state-of-the-art LMs
- **Unique features:** Top performance of fully documented model, instruction tuned etc.

Arch	Transformer+RoPE+SwiGLU, context 4k, non-parametric LN
Data	Trained on 2.46T tokens of Dolma corpus (next slide)
Train	LR scaled inversely to model size (7B=3e-4), batch size 4M tokens

OLMo

OLMo: Accelerating the Science of Language Models

Dirk Groeneveld[α] Iz Beltagy[α]
Pete Walsh[α] Akshita Bhagia[α] Rodney Kinney[α] Oyvind Tafjord[α]
Ananya Harsh Jha[α] Hamish Ivison[αβ] Ian Magnusson[α] Yizhong Wang[αβ]
Shane Arora[α] David Atkinson[α] Russell Authur[α] Khyathi Raghavi Chandu[α]
Arman Cohan[γα] Jennifer Dumas[α] Yanai Elazar[αβ] Yuling Gu[α]
Jack Hessel[α] Tushar Khot[α] William Merrill[δ] Jacob Morrison[α]
Niklas Muennighoff Aakanksha Naik[α] Crystal Nam[α] Matthew E. Peters[α]
Valentina Pyatkin[αβ] Abhilasha Ravichander[α] Dustin Schwenk[α] Saurabh Shah[α]
Will Smith[α] Emma Strubell[αμ] Nishant Subramani[α] Mitchell Wortsman[β]
Pradeep Dasigi[α] Nathan Lambert[α] Kyle Richardson[α]
Luke Zettlemoyer[β] Jesse Dodge[α] Kyle Lo[α] Luca Soldaini[α]
Noah A. Smith[αβ] Hannaneh Hajishirzi[αβ]

[α]Allen Institute for Artificial Intelligence
[β]University of Washington [γ]Yale University
[δ]New York University [μ]Carnegie Mellon University

olmo@allenai.org

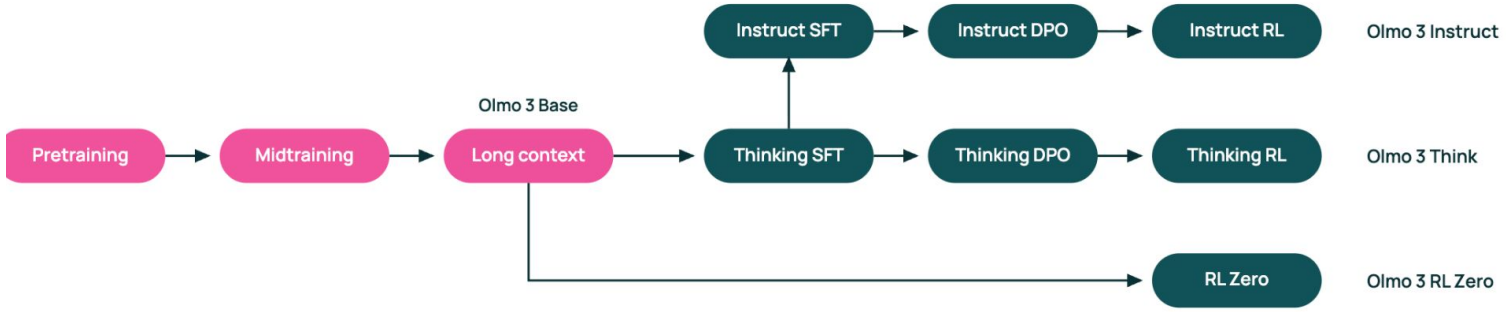
Abstract

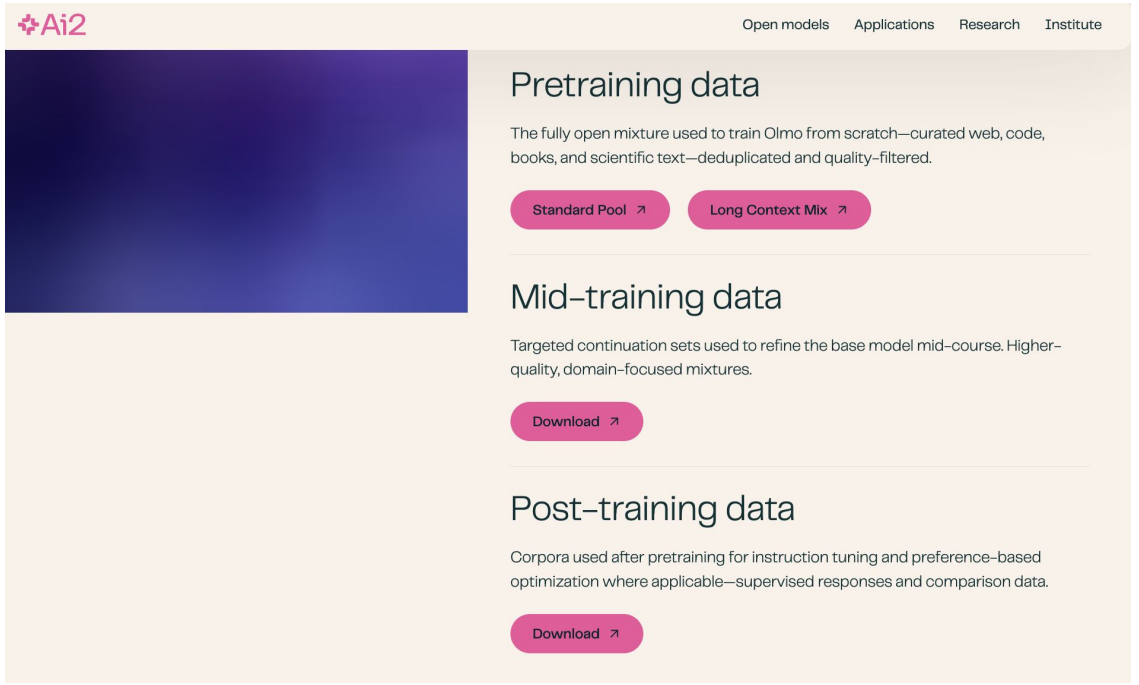
Language models (LMs) have become ubiquitous in both NLP research and in commercial product offerings. As their commercial importance has surged, the most powerful models have become closed off, gated behind proprietary interfaces, with important details of their training data, architectures, and development undisclosed. Given the importance of these details in scientifically studying these models, including their biases and potential risks, we believe it is essential for the research community to have access to powerful, truly open LMs. To this end, we have built OLMo, a competitive, truly **O**pen **L**anguage **Mo**del, to enable the scientific study of language models. Unlike most prior efforts that have only released model weights and inference code, we release OLMo alongside open training data and training and evaluation code. We hope this release will empower the open research community and inspire a new wave of innovation.

gio et al., 2003; Mikolov et al., 2013; Peters et al., 2018; Brown et al., 2020). Recently, due to large-scale pretraining and human annotation for alignment, they have become commercially valuable (OpenAI, 2023). However, as their commercial value has increased, the largest models have become gated behind proprietary interfaces, with important details left undisclosed.

We believe that full access to open language models for the research community is critical to the scientific study of these models, their strengths and weaknesses, and their biases and risks. Accordingly, we introduce **OLMo**, a powerful, truly open language model alongside open training data, training and evaluation code, intermediate model checkpoints, and training logs.

Recent LM releases have varied in their degree of openness. For example, Mixtral 8x7B provided model weights and a brief report (Jiang et al., 2024), while LLaMA came with in-depth adap-





Dolma

3T token corpus created and released by AI2 for LM training

a pipeline of (1) language filtering, (2) quality filtering, (3) content filtering, (4) deduplication, (5) multi-source mixing, and (6) tokenization

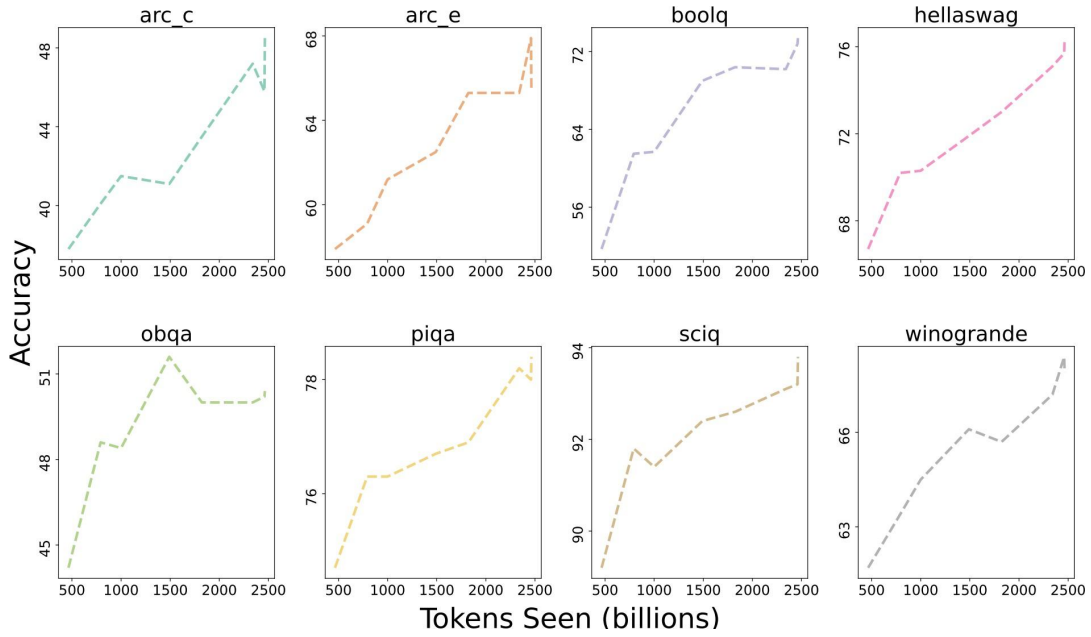
Source	Doc Type	UTF-8 bytes *(GB)*	Documents *(millions)*	Unicode words *(billions)*	Llama tokens *(billions)*
Common Crawl	web pages	9,022	3,370	1,775	2,281
The Stack	code	1,043	210	260	411
C4	web pages	790	364	153	198
Reddit	social media	339	377	72	89
PeS2o	STEM papers	268	38.8	50	70
Project Gutenberg	books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	encyclopedic	16.2	6.2	3.7	4.3
Total		**11,519**	**4,367**	**2,318**	**3,059**

OLMo - Findings – Competitive average performance

7B Models	arc challenge	arc easy	boolq	hella-swag	open bookqa	piqa	sciq	wino-grande	avg.
Falcon	47.5	70.4	74.6	75.9	53.0	78.5	93.9	68.9	70.3
LLaMA	44.5	67.9	75.4	76.2	51.2	77.2	93.9	70.5	69.6
Llama 2	48.5	69.5	80.2	76.8	48.4	76.7	94.5	69.4	70.5
MPT	46.5	70.5	74.2	77.6	48.6	77.3	93.7	69.9	69.8
Pythia	44.1	61.9	61.1	63.8	45.0	75.1	91.1	62.0	63.0
RPJ-INCITE	42.8	68.4	68.6	70.3	49.4	76.0	92.9	64.7	66.6
OLMo-7B	48.5	65.4	73.4	76.4	50.4	78.4	93.8	67.9	69.3

Table 6: Zero-shot evaluation of OLMo-7B and 6 other publicly available comparable model checkpoints on 8 core tasks from the downstream evaluation suite described in Section 2.4. For OLMo-7B, we report results for the 2.46T token checkpoint.

Performance increases constantly w/ training



LLaMa2 - Overview

- **Creator:** | Meta
- **Goal:** Strong and safe open LM w/ base+chat versions
- **Unique features:** Open model with strong safeguards and chat tuning, good performance

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm
Data	Trained on “public sources, up-sampling the most factual sources”, LLaMa 1 has more info (next page), total 2T tokens
Train	7B=3e-4, batch size 4M tokens

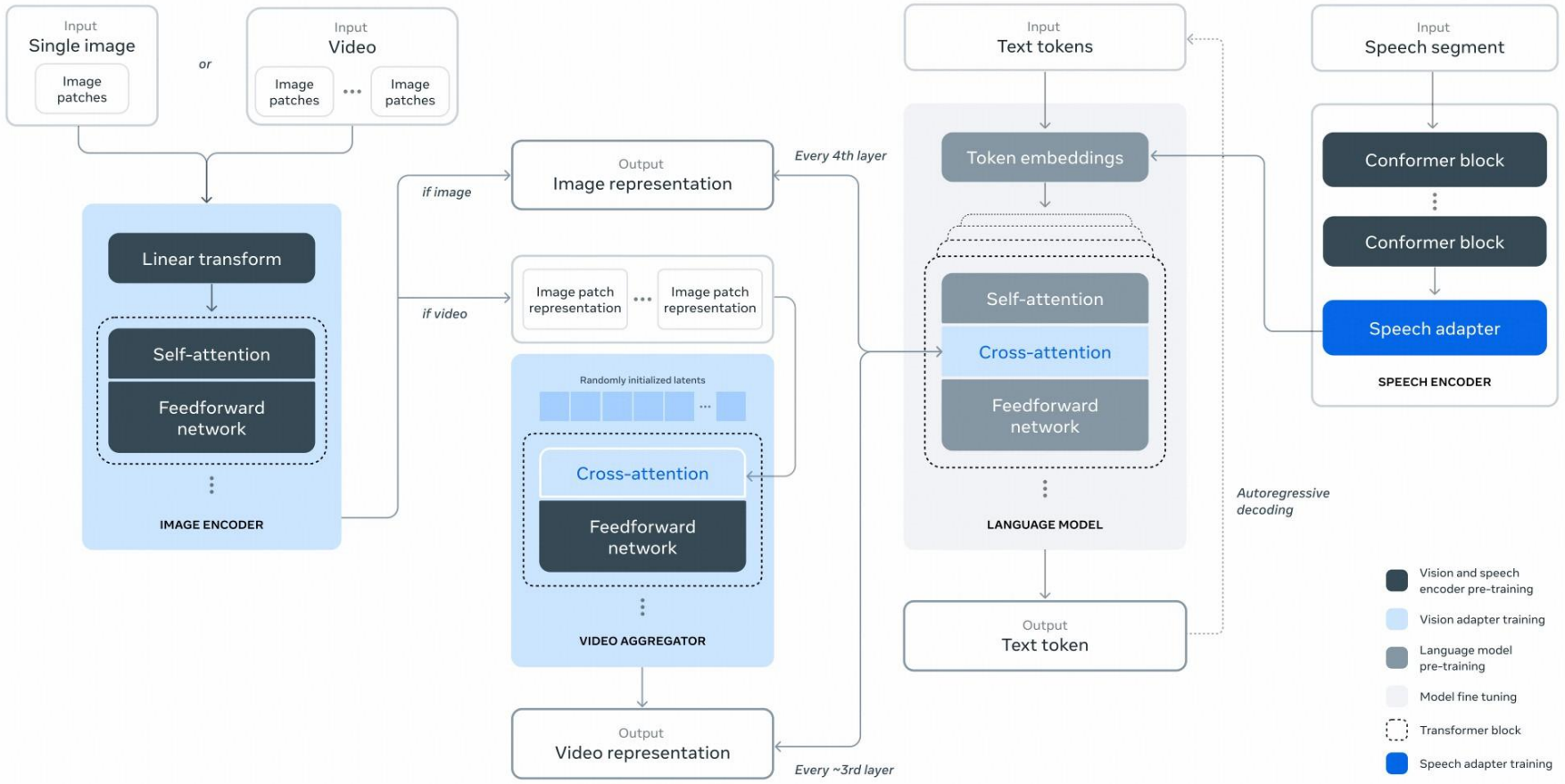
LLaMa 3.1 - Overview

- **Creator:** Meta
- **Goal:** A herd of language models that natively support multilinguality, coding, reasoning, and tool usage
- **Compared with Llama 2:** Larger Data scale (15T multilingual tokens vs 1.8T tokens). More Training FLOPs (3.8 × 10^25 FLOPs, almost 50× more than the largest version of Llama 2)

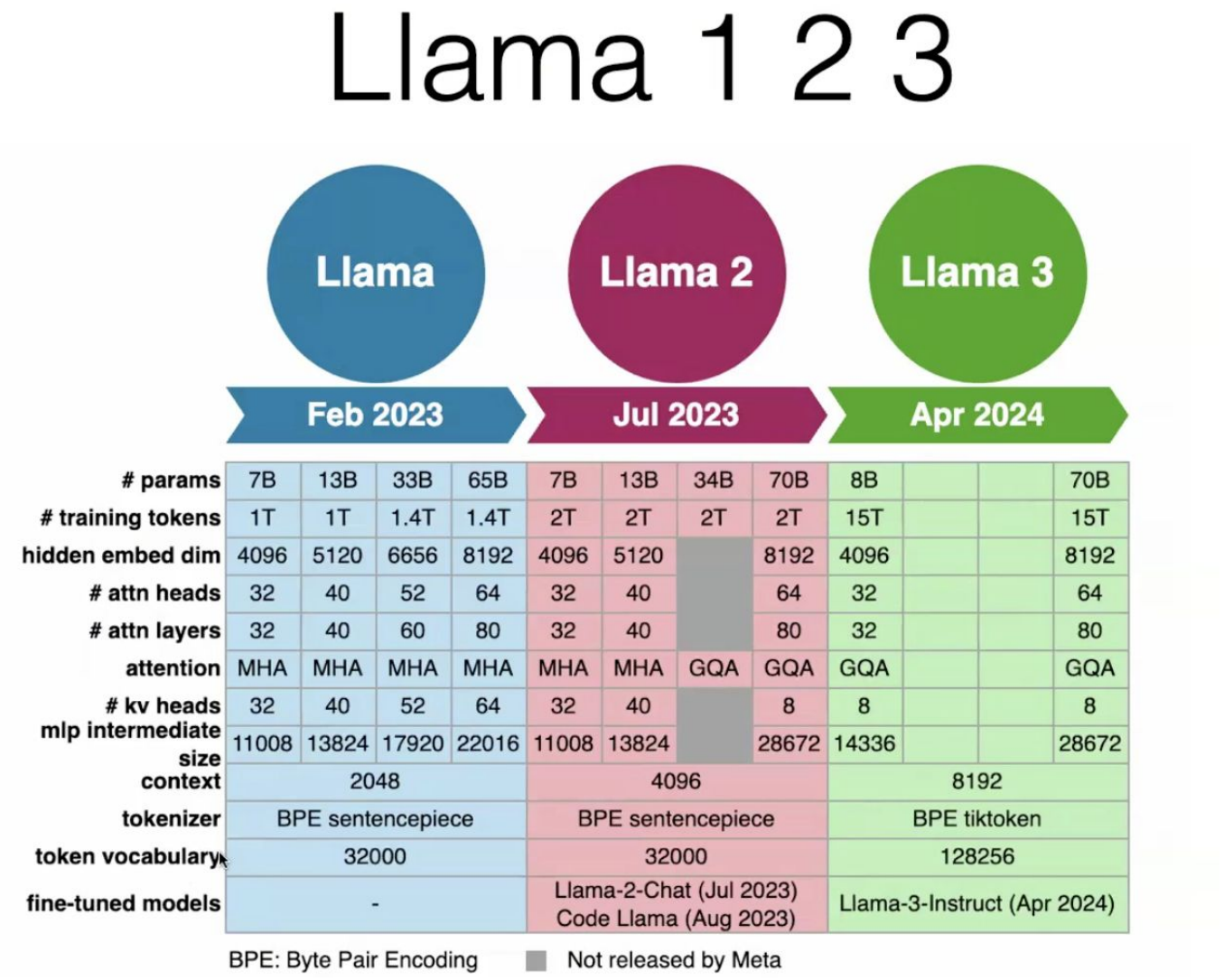
GPUs	TP	CP	PP	DP	Seq. Len.	Batch size/DP	Tokens/Batch	TFLOPs/GPU	BF16 MFU
8,192	8	1	16	64	8,192	32	16M	430	43%
16,384	8	1	16	128	8,192	16	16M	400	41%
16,384	8	16	16	4	131,072	16	16M	380	38%

Table 4 Scaling configurations and MFU for each stage of Llama 3 405B pre-training. See text and Figure 5 for descriptions of each type of parallelism.

LLaMa 3.1 - Multimodality



LLaMa Family



# params	7B	13B	33B	65B	7B	13B	34B	70B	8B			70B
# training tokens	1T	1T	1.4T	1.4T	2T	2T	2T	2T	15T			15T
hidden embed dim	4096	5120	6656	8192	4096	5120		8192	4096			8192
# attn heads	32	40	52	64	32	40		64	32			64
# attn layers	32	40	60	80	32	40		80	32			80
attention	MHA	MHA	MHA	MHA	MHA	MHA	GQA	GQA	GQA			GQA
# kv heads	32	40	52	64	32	40		8	8			8
mlp intermediate size	11008	13824	17920	22016	11008	13824		28672	14336			28672
context	2048				4096				8192			
tokenizer	BPE sentencepiece				BPE sentencepiece				BPE tiktoken			
token vocabulary	32000				32000				128256			
fine-tuned models	-				Llama-2-Chat (Jul 2023) Code Llama (Aug 2023)				Llama-3-Instruct (Apr 2024)			

BPE: Byte Pair Encoding Not released by Meta

Mistral/Mixtral - Overview

- **Creator:** 

- **Goal:** Strong and somewhat multilingual open LM
- **Unique features:** Speed optimizations, including GQA and Mixture of Experts

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm, sliding window attention. Mixtral has 8x experts in feed-forward layer
Data	Not disclosed? But includes English and European languages
Train	Not disclosed?

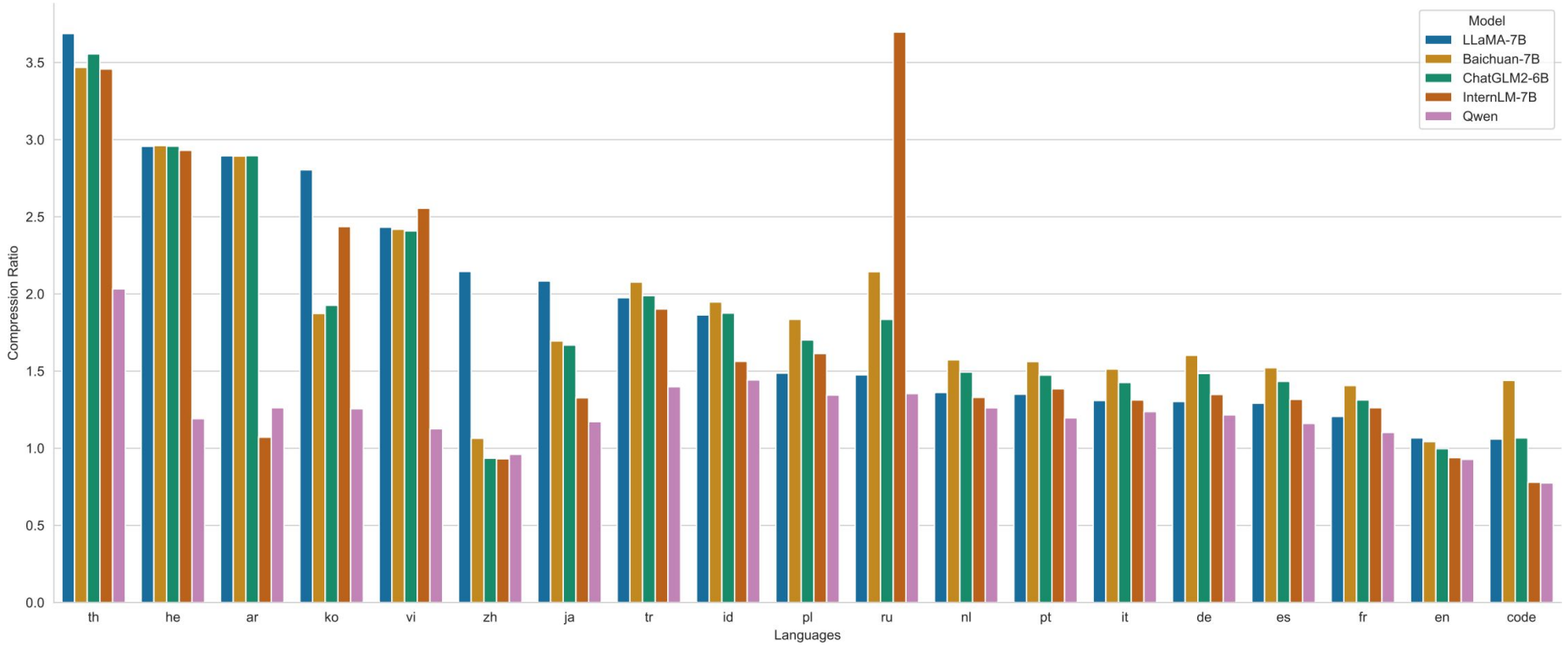
Qwen - Overview

- **Creator:** Alibaba
- **Goal:** Strong multilingual (esp. English and Chinese) LM
- **Unique features:** Large vocabulary for multilingual support, strong performance

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm, bias in attention layer
Data	Trained on multilingual data + instruction data at pre-training time, 2-3T tokens
Train	3e-4, batch size 4M tokens

Qwen - Multilinguality

- Token compression ratio re: XLM-R (lower is better)



SmolLM - Overview

- **Creator:** 

- **Goal:** Small scale (135M, 360M, and 1.7B parameters) but strong performance
- **Unique features:** Fully Open-sourced with a high-quality pre-training corpus.

- **Cosmopedia v2**: A collection of synthetic textbooks and stories generated by Mixtral (28B tokens)
- **Python-Edu**: educational Python samples from The Stack (4B tokens)
- **FineWeb-Edu (deduplicated)**: educational web samples from FineWeb (220B tokens)

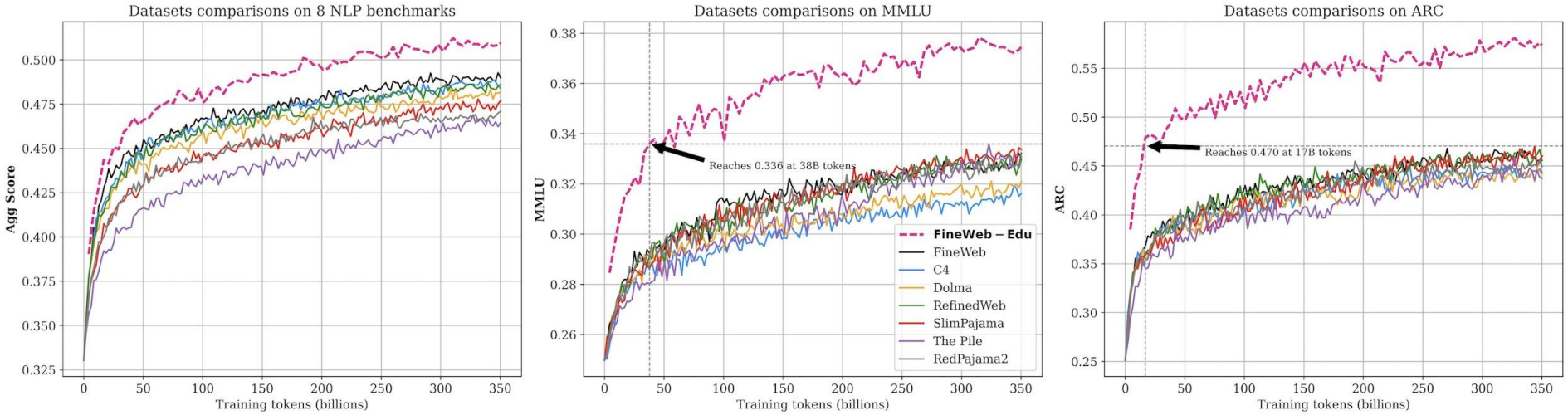
https://huggingface.co/blog/smollm

FineWeb - (Edu)

FineWeb dataset consists of more than 15T tokens of cleaned and deduplicated english web data from CommonCrawl.

Url Filtering -> Trafilatura text extraction from HTML ->FastText LanguageFilter -> Quality filtering -> MinHash deduplication -> PII Formatting

“To enhance FineWeb's quality, we developed an educational quality classifier using annotations generated by LLama3-70B-Instruct. We then used this classifier to retain only the most educational web pages.”



Other Models

Code Models

- **StarCoder 2** — by Big Science (leads: Hugging Face + Service Now), fully open model
- **CodeLlama** — by Meta, code adaptation of LLaMa
- **DeepSeek Coder** — by DeepSeek, strong performance across many tasks
- **Yi Coder -** by 01.AI, smaller scales (9B/1.5B) but strong performance.

Math Models

- **LLEMA** — by EleutherAI and others, model for math theorem proving trained on proof pile
- **DeepSeek Math** — by DeepSeek, finds math-related pages on the web
- More in code and math class!

Science Model: Galactica

- Model for science trained by Meta
- Diverse set of interesting training data

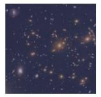
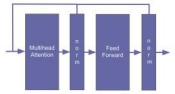
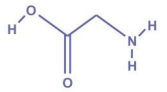
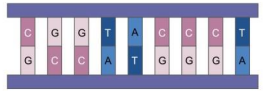
Modality	Entity	Sequence	
Text	Abell 370	`Abell 370 is a cluster...`	
LaTeX	Schwarzschild radius	`r_{s} = \frac{2GM}{c^2}`	$r_s = \frac{2GM}{c^2}$
Code	Transformer	`class Transformer(nn.Module)`	
SMILES	Glycine	`C(C(=O)O)N`	
AA Sequence	Collagen α-1(II) chain	`MIRLGAPQTL..`	
DNA Sequence	Human genome	`CGGTACCCTC..`	

Table 1: Tokenizing Nature. Galactica trains on text sequences that represent scientific phenomena.

Closed Models

GPT-4o - Overview

- **Creator:** 

- De-facto standard “strong” language model
- Tuned to be good as a chat-based assistant
- Supports calling external tools through “function calling” interface
- Accepts image inputs
- Fast and cheaper inference compared with earlier GPT-4 versions

Gemini

- **Creator:** 

- Performance competitive with corresponding GPT models (Gemini Pro 1.0 ~ gpt-3.5, Gemini Ultra 1.0 ~ gpt-4)
- Pro 1.5 supports very long inputs, 1-10M tokens
- Supports image and video inputs
- Can generate images natively

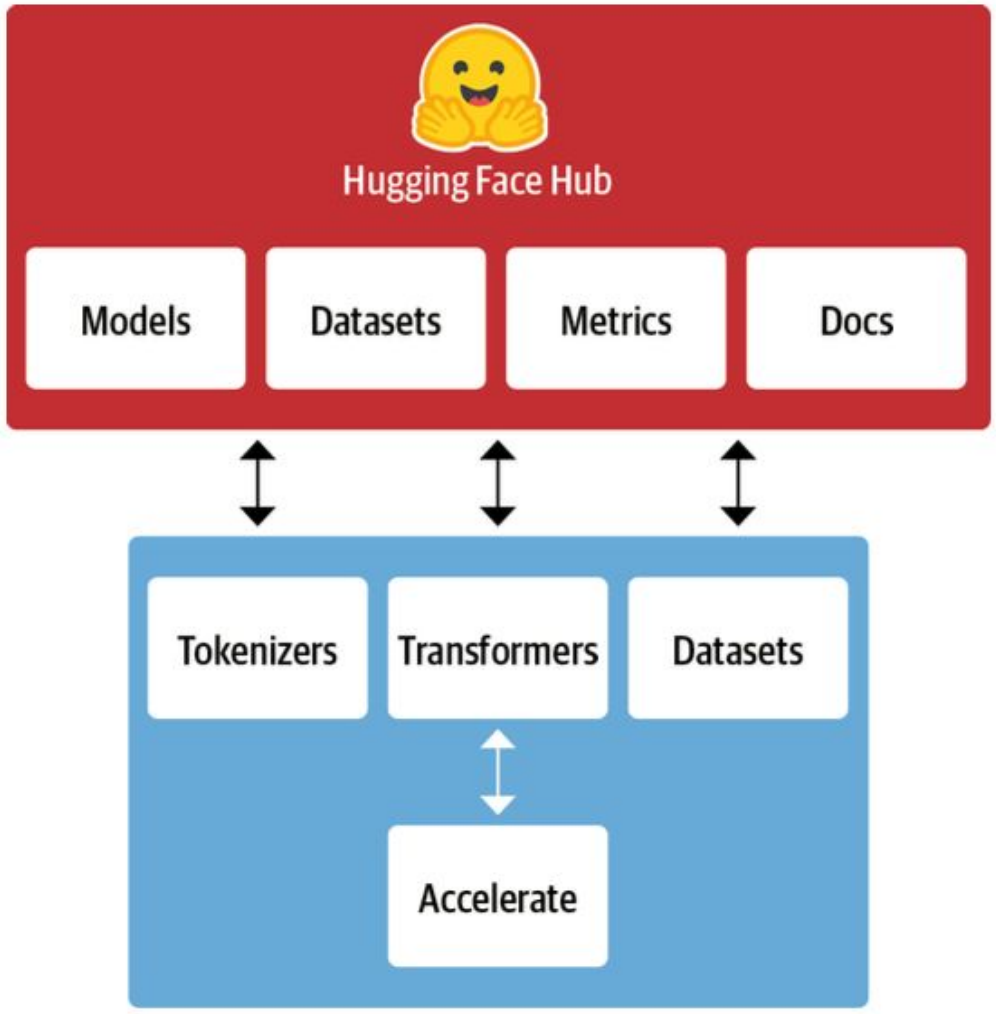
Claude 3 - Overview

- **Creator:** ANTHROP\C
- Context window up to 200k
- Allows for processing images
- Overall strong results competitive with GPT-4

Models Models Everywhere

Transformers, a library that catalyzed the explosion of research into **transformers,** making it easy to integrate these models into many real-life applications today

The Hugging Face Ecosystem



The Hugging Face ecosystem consists mainly of two parts: a family of libraries and the Hub.

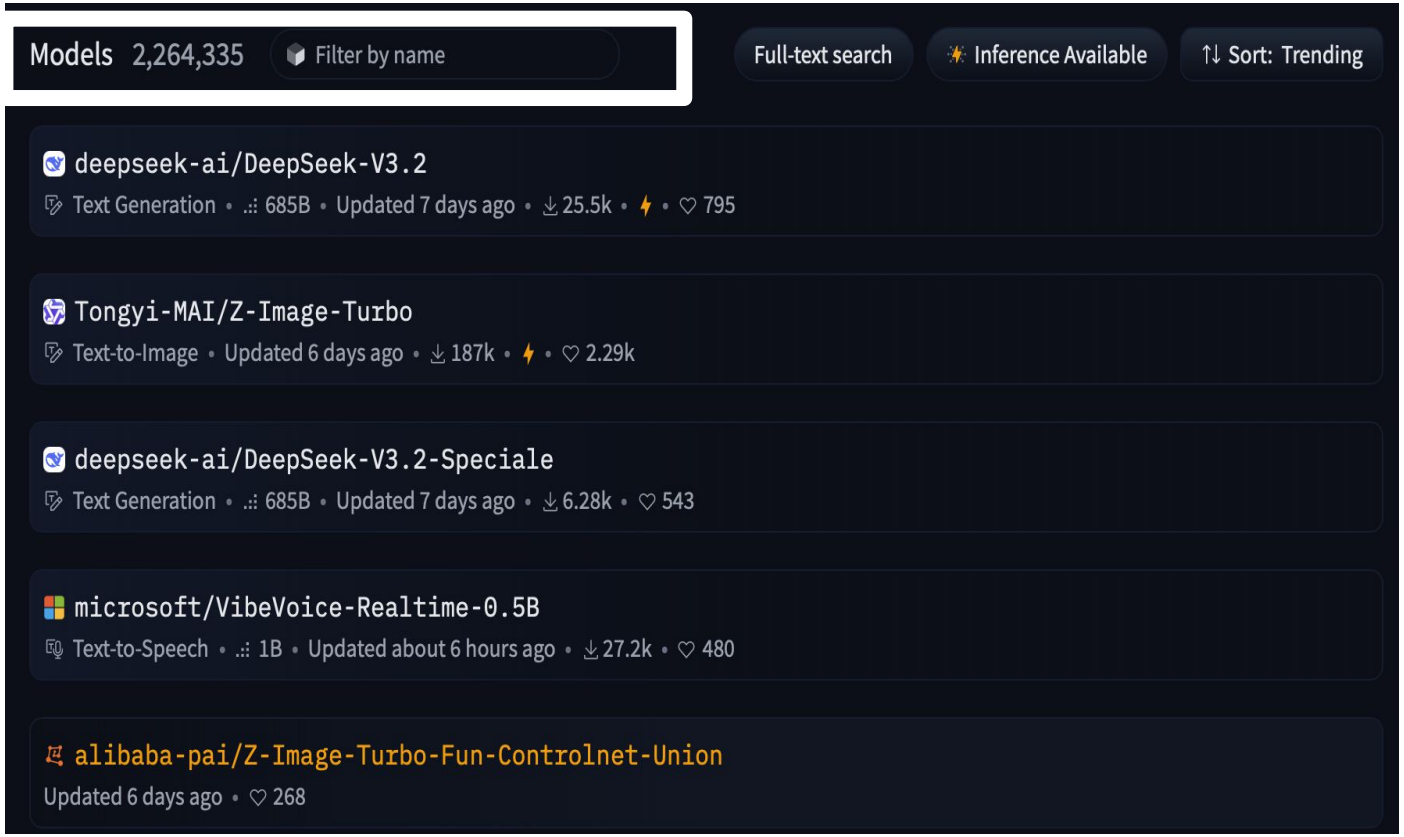
The libraries **provide the code** while the ***Hub provides the pretrained model weights, datasets, scripts for the evaluation metrics, and more.***

The Hugging Face Ecosystem

Different research labs release their models in incompatible frame-works (PyTorch or TensorFlow), it wasn't always easy for NLP practitioners to port these models to their own applications.

With the release of **Transformers**, a unified API across many architectures was progressively built. **Transformers** catalyzed the explosion of research into **transformers and quickly trickled down to NLP practi-tioners**, making it easy to integrate these models into many real-life applications today. Let's have a look!

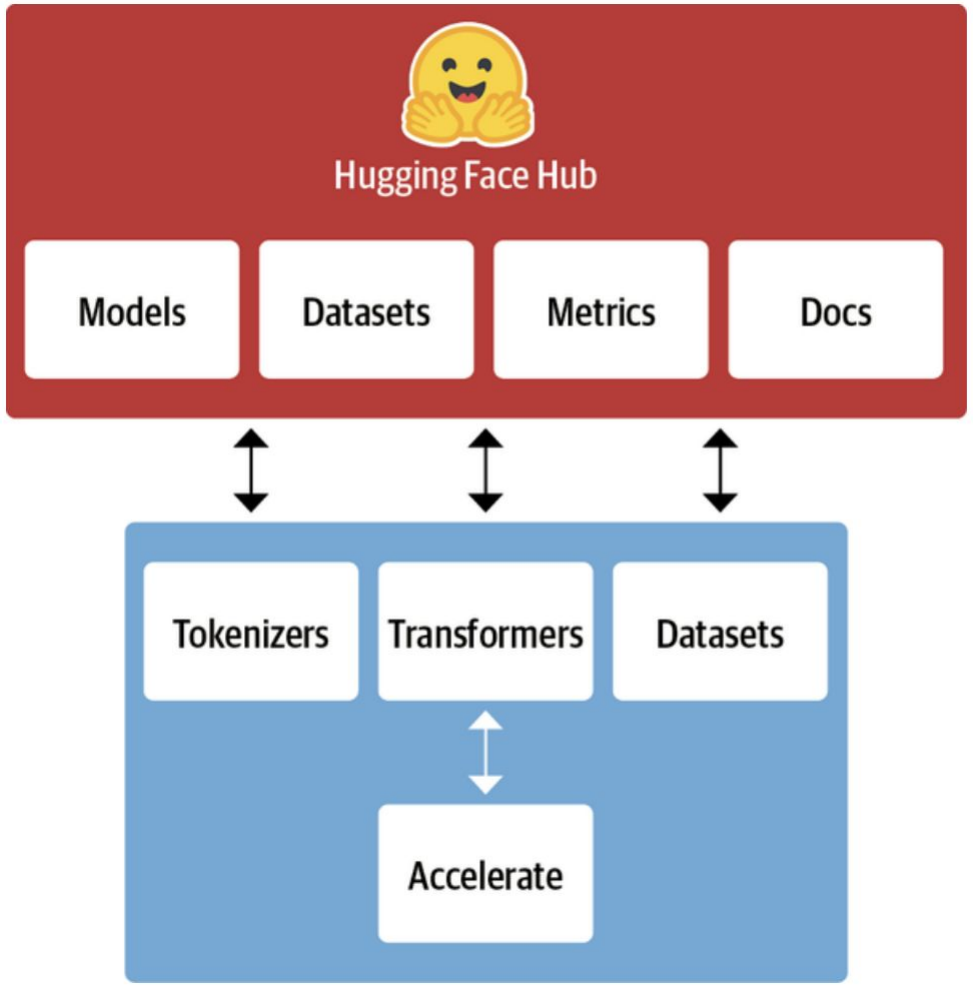
The Hugging Face Ecosystem



As outlined earlier, transfer learning is one of the key factors driving the success of transformers because it makes it possible to reuse pretrained models for new tasks. Consequently, it is crucial to be able to load pretrained models quickly and run experiments with them.

The Hugging Face Hub hosts over 2 million freely available models

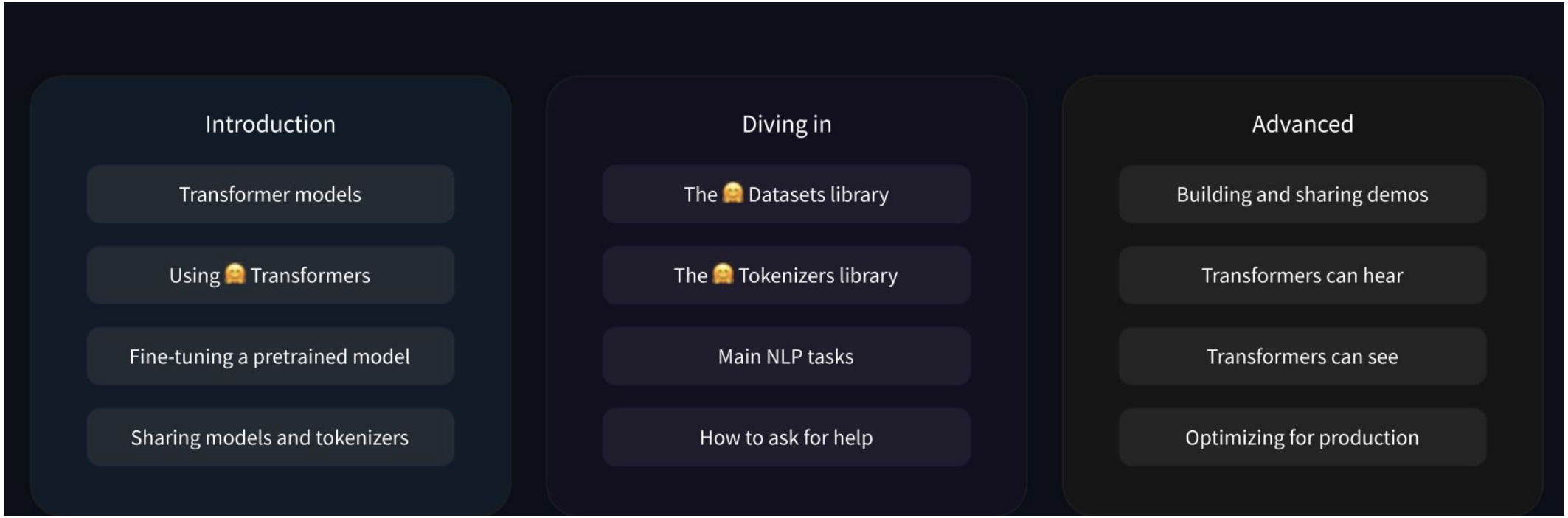
HuggingFace Hub



HuggingFace NLP Course

https://huggingface.co/learn/nlp-course/en/chapter1/1

HuggingFace LLM Course



This course will teach you about large language models (LLMs) and natural language processing (NLP) using libraries from the Hugging Face ecosystem — 🤗 Transformers, 🤗 Datasets, 🤗 Tokenizers, and 🤗 Accelerate — as well as the Hugging Face Hub.
https://huggingface.co/learn/llm-course/en/chapter1/1

IMPERIAL

Q and A

IMPERIAL

Next Lecture: Fine-tuning and Instruction Tuning