Scaling and Emergent Behavior for Large Language Models

Mateen Afshari, Preethi Chidambaram, Nitin Maddi

Agenda

- Training Compute-Optimal Large Language Models
- Scaling Data-Constrained Language Models
- Emergent Abilities of Large Language Models
- Are Emergent Abilities of Large Language Models a Mirage?

Training Compute-Optimal Large Language Models

Authors: Jordan Hoffmann et al.

Publication Date: Mar 2022

Purpose

What amount of training tokens and parameters are needed to make a computationally efficient model given a fixed compute budget?

- The compute and energy cost for training large language models is substantial
- Allocated training compute budget is often known in advance
- Only feasible to train these large models once

Related Work

Kaplan et al. (2020) showed that there is a power law relationship between the number of parameters in an autoregressive language model (LM) and its performance.

- The field has been training larger and larger models, expecting performance improvements
- Given a 10× increase computational budget, they suggests that the size of the model should increase 5.5× while the number of training tokens should only increase 1.8×.

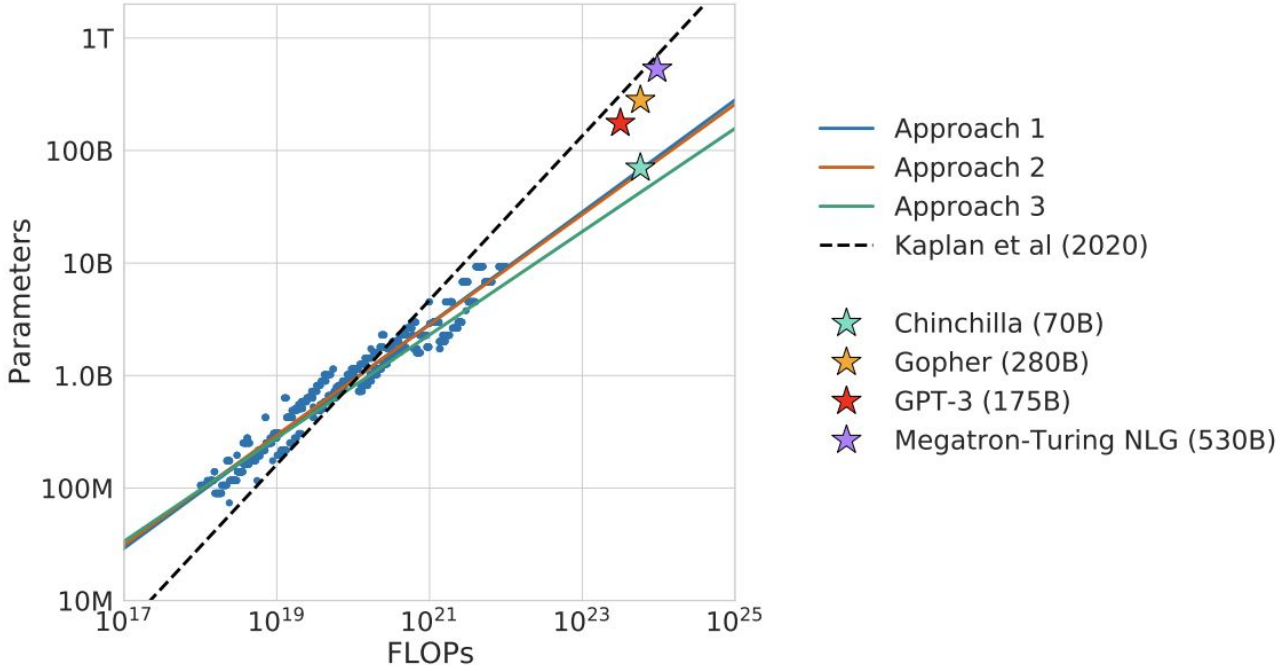
Current Models

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

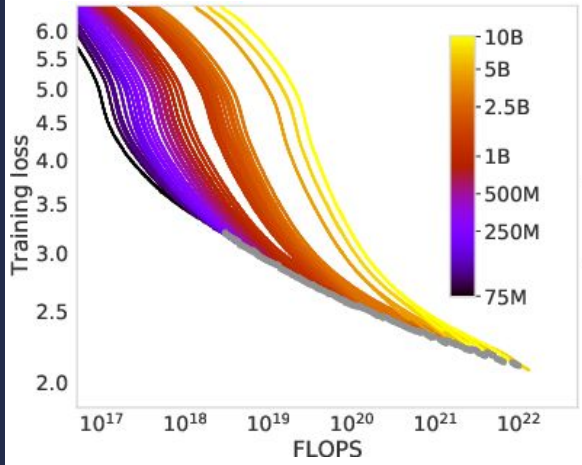
Methodology

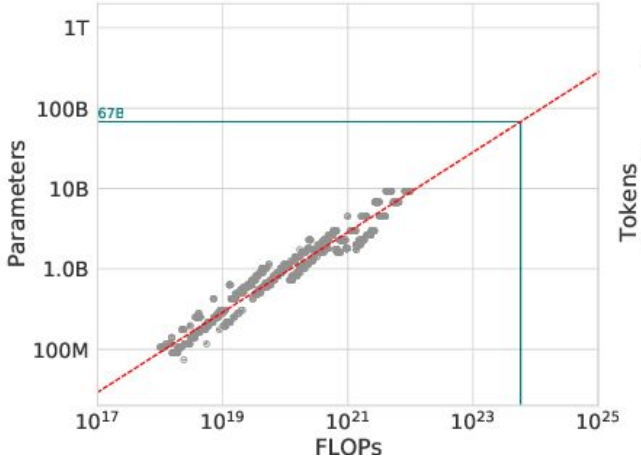
- Trained over 400 language models
- Model size ranged from 70 million to over 16 billion parameters
- Models trained on 5 to 500 billion tokens

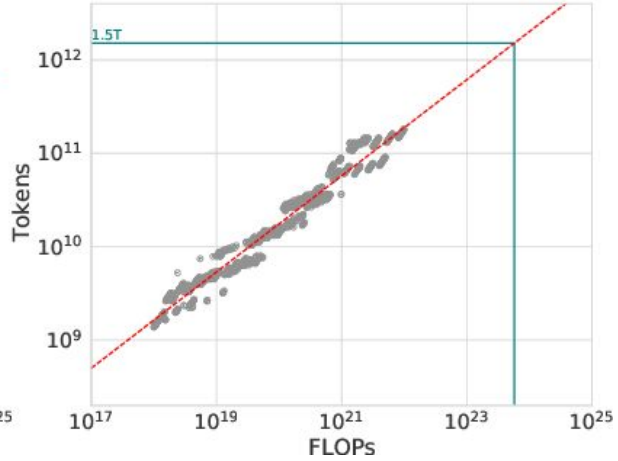
Methodology



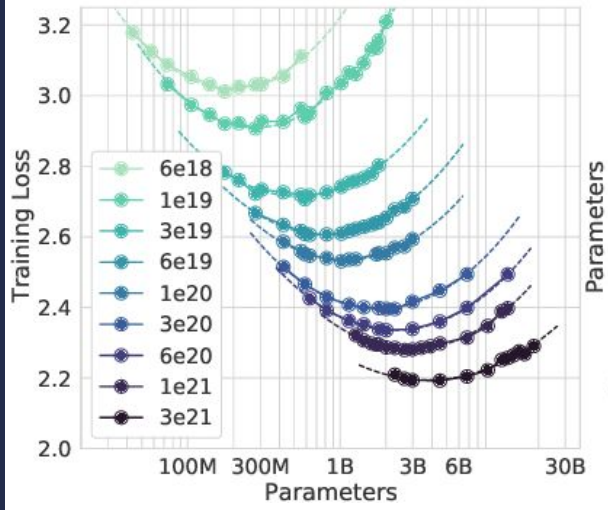
Approach 1:
Fix model sizes and vary number of training tokens

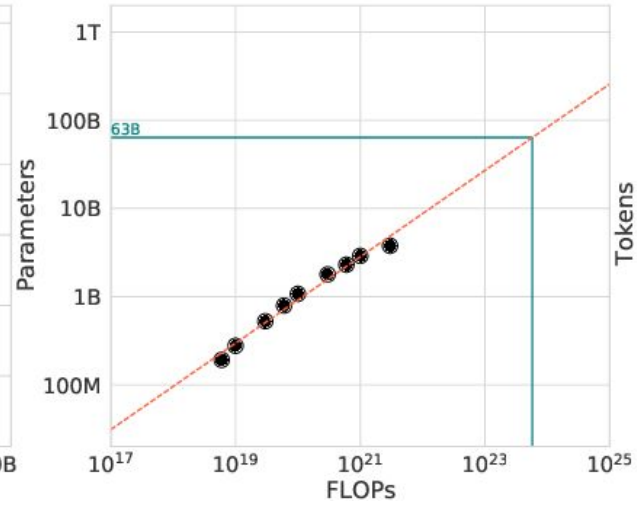


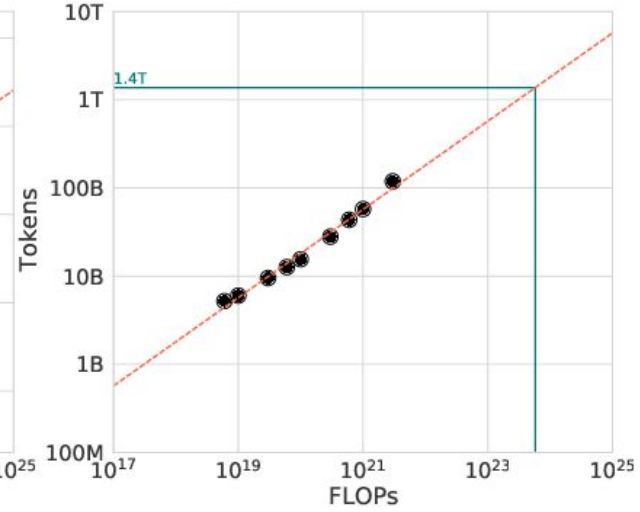




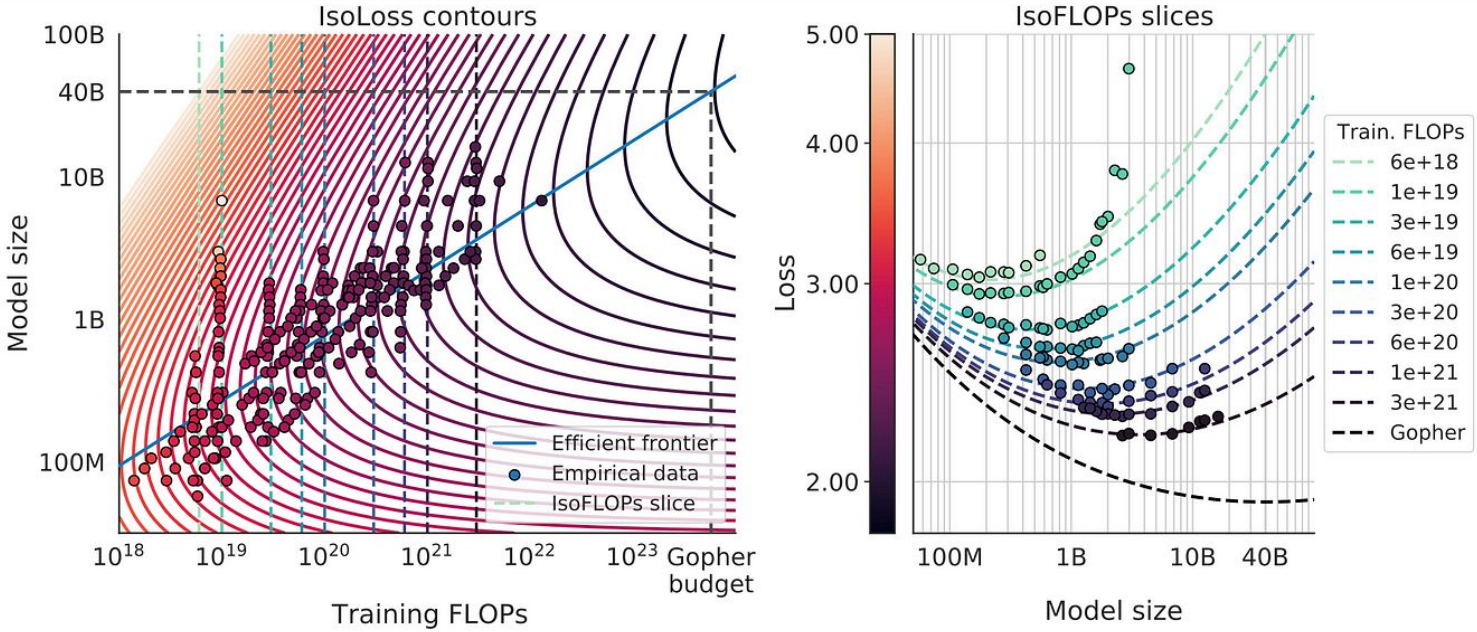
Approach 2: IsoFLOP profiles







Approach 3:
Fitting a parametric loss function



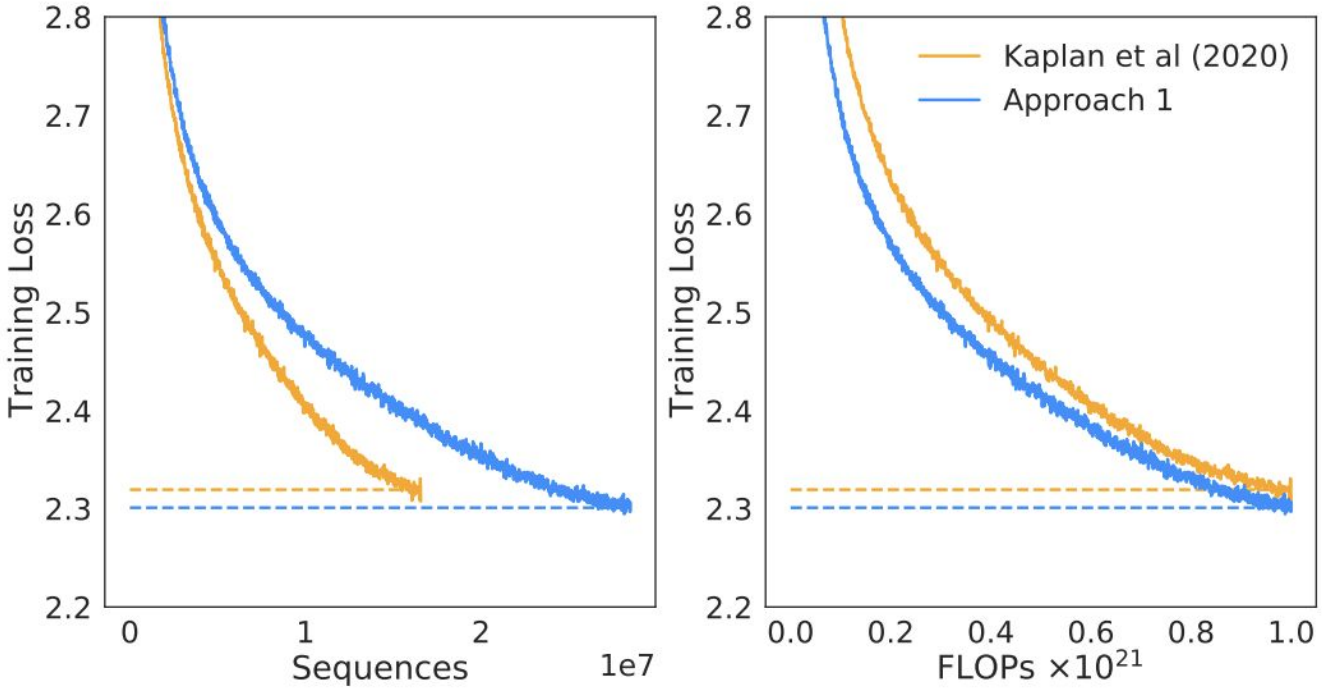
Optimal Model Scaling

Parameters	FLOPs	FLOPs (in *Gopher* unit)	Tokens
400 Million	1.92e+19	1/29,968	8.0 Billion
1 Billion	1.21e+20	1/4,761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

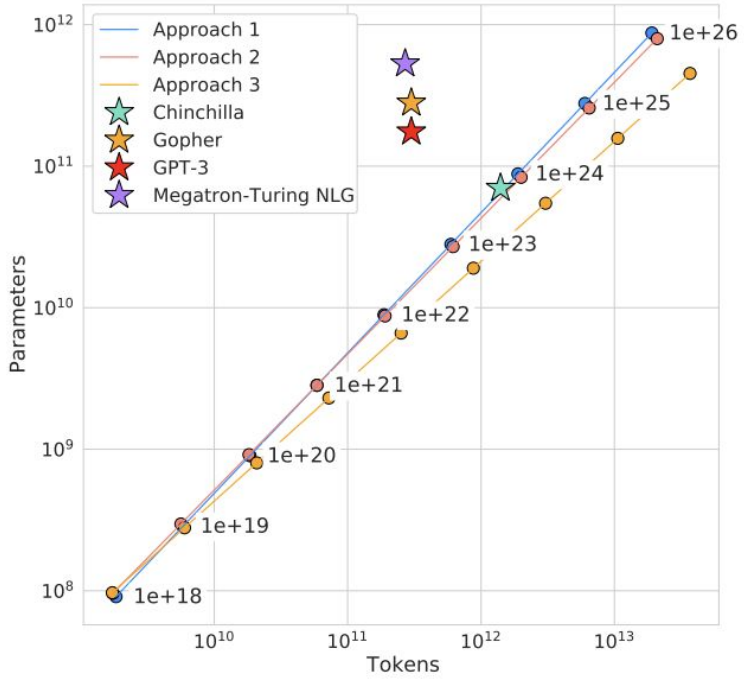
Optimal Model Scaling

	Approach 2		Approach 3	
Parameters	FLOPs	Tokens	FLOPs	Tokens
400 Million	1.84e+19	7.7 Billion	2.21e+19	9.2 Billion
1 Billion	1.20e+20	20.0 Billion	1.62e+20	27.1 Billion
10 Billion	1.32e+22	219.5 Billion	2.46e+22	410.1 Billion
67 Billion	6.88e+23	1.7 Trillion	1.71e+24	4.1 Trillion
175 Billion	4.54e+24	4.3 Trillion	1.26e+24	12.0 Trillion
280 Billion	1.18e+25	7.1 Trillion	3.52e+25	20.1 Trillion
520 Billion	4.19e+25	13.4 Trillion	1.36e+26	43.5 Trillion
1 Trillion	1.59e+26	26.5 Trillion	5.65e+26	94.1 Trillion
10 Trillion	1.75e+28	292.0 Trillion	8.55e+28	1425.5 Trillion

Optimal Model Scaling



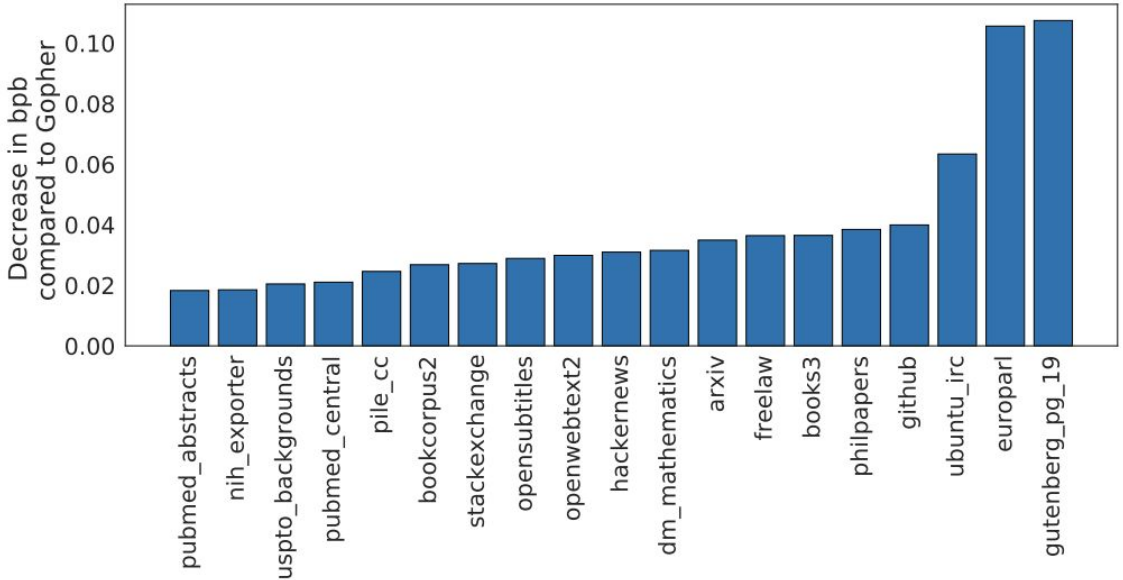
Chinchilla


Approach 1
Approach 2
Approach 3
Chinchilla
Gopher
GPT-3
Megatron-Turing NLG
1e+18
1e+19
1e+20
1e+21
1e+22
1e+23
1e+24
1e+25
1e+26
Parameters
Tokens
10^{8}
10^{9}
10^{10}
10^{11}
10^{12}
10^{10}
10^{11}
10^{12}
10^{13}

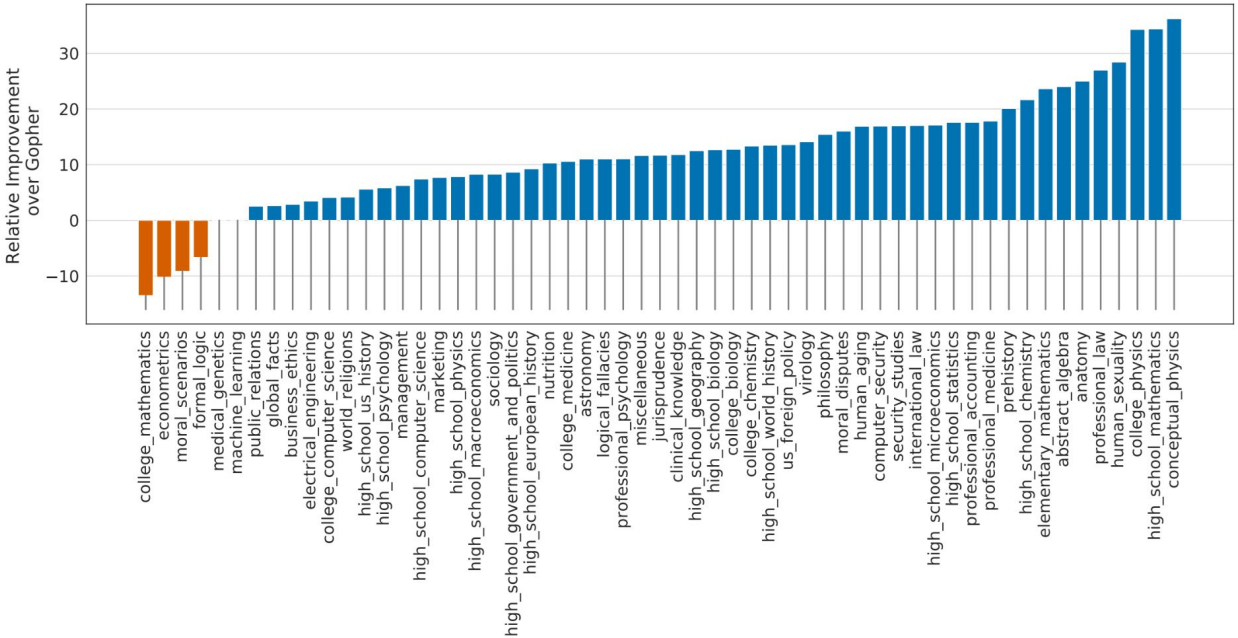
Results

	# Tasks	Examples
Language Modelling	20	WikiText-103, The Pile: PG-19, arXiv, FreeLaw, ...
Reading Comprehension	3	RACE-m, RACE-h, LAMBADA
Question Answering	3	Natural Questions, TriviaQA, TruthfulQA
Common Sense	5	HellaSwag, Winogrande, PIQA, SIQA, BoolQ
MMLU	57	High School Chemistry, Astronomy, Clinical Knowledge, ...
BIG-bench	62	Causal Judgement, Epistemic Reasoning, Temporal Sequences, ...

Results


Decrease in bpb compared to Gopher
0.00
0.02
0.04
0.06
0.08
0.10
pubmed_abstracts
nih_exporter
uspto_backgrounds
pubmed_central
pile_cc
bookcorpus2
stackexchange
opensubtitles
openwebtext2
hackernews
dm_mathematics
arxiv
freelaw
books3
philpapers
github
ubuntu_irc
europarl
gutenberg_pg_19

Results


Relative Improvement over Gopher
30
20
10
0
−10
college_mathematics
econometrics
moral_scenarios
formal_logic
medical_genetics
machine_learning
public_relations
global_facts
business_ethics
electrical_engineering
college_computer_science
world_religions
high_school_us_history
high_school_psychology
management
high_school_computer_science
marketing
high_school_physics
high_school_macroeconomics
sociology
high_school_government_and_politics
high_school_european_history
nutrition
college_medicine
astronomy
logical_fallacies
professional_psychology
miscellaneous
jurisprudence
clinical_knowledge
high_school_geography
high_school_biology
college_biology
college_chemistry
high_school_world_history
us_foreign_policy
virology
philosophy
moral_disputes
human_aging
computer_security
security_studies
international_law
high_school_microeconomics
high_school_statistics
professional_accounting
professional_medicine
prehistory
high_school_chemistry
elementary_mathematics
abstract_algebra
anatomy
professional_law
human_sexuality
college_physics
high_school_mathematics
conceptual_physics

Key Takeaways

- Emphasizes the importance of optimizing compute resources for training large language models, balancing model size and training data.
- Showcases how the Chinchilla model outperforms other large models in various tasks, highlighting the effectiveness of the compute-optimal approach.
- Presents a critical view of the prevailing trend in scaling up model size without proportionately increasing training data.

Limitations

- **Limited Large Scale Data:** Due to the cost of training large models, only two large scale models were compared (Chinchilla and Gopher)
- **May be overestimating the optimal size of large model:** Concavity observed at higher compute budgets
- Large datasets scraped from the web will contain toxic language, biases, and private information

Scaling Data Constrained Language Models

Authors: Niklas Muennighoff et al.

Publication Date: October 2023

Motivation

"Extrapolating this trend suggests that training dataset size may soon be limited by the amount of text data available on the internet"

- Current trend - increasing parameter count and training dataset size
- Data repetition
- Two fundamental questions
  - Allocation: What is the optimal balance of resources?
  - Return: What is the expected value of additional resources?

Background

- Computational power
  - Measured in FLOPs
- Effectiveness of training
  - Measured by loss
- Scaling law for allocation and return
  - Loss scales as a power law
  - Increase model size and amount of data equally

Related Work

This paper references the work in the previous paper (*Training Compute-Optimal Large Language Model*) to corroborate their claims on scaling data constrained models.

- Chinchilla model outperformed Gopher model
- 3 methods for making scaling predictions
  - Fixed parameters
  - Fixed FLOPs
  - Parametric fit
- Conclusion: Model size and training data should be increased proportionally

$$L(N, D) = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E$$

Methodology

- Primary method: repeating data
- Split data and parameters
  - Data divided into unique and repeated tokens
  - Parameters divided into base params and repetition factor
- Similar experimental methods as Chinchilla model
- Loss function defined as $L(N, D) = \frac{A}{N'^{\alpha}} + \frac{B}{D'^{\beta}} + E$

Methodology

Researchers propose that repeated data and model size gradually become less useful in training.

Effective Data

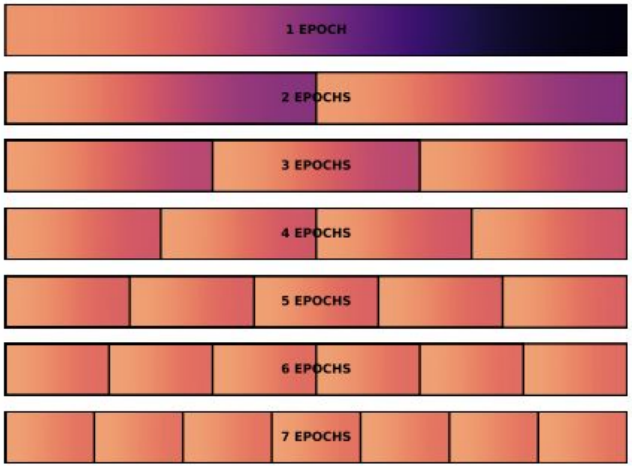
$$D' = U_D + U_D R_D^* (1 - e^{\frac{-R_D}{R_D^*}})$$

Effective Model Parameters

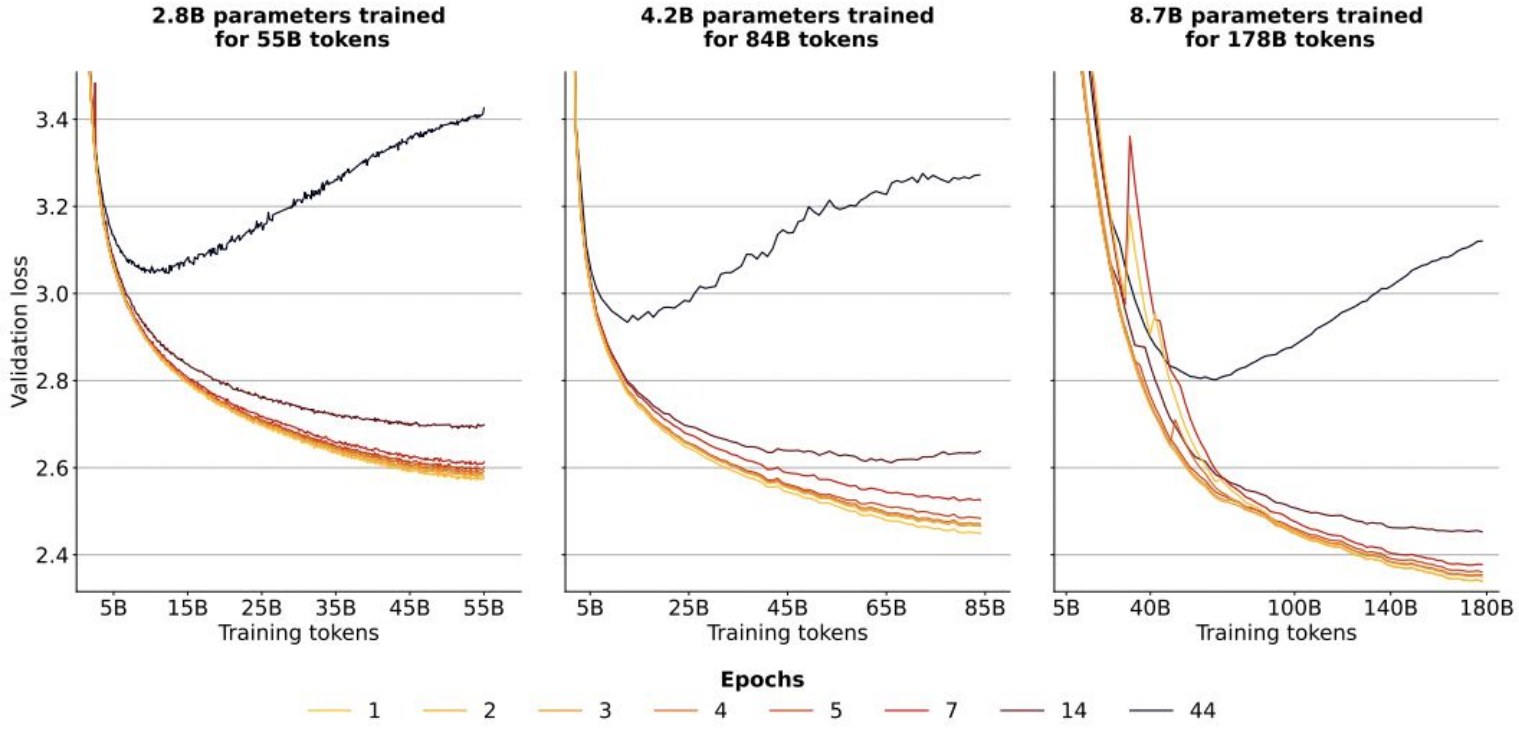
$$N' = U_N + U_N R_N^* (1 - e^{\frac{-R_N}{R_N^*}})$$

Experimental Setup

- Transformer language models with GPT-2 architecture
- Epochs repeat entire set of available data
  - Shuffled after each epoch
- Not much exploration into the extent of overfitting

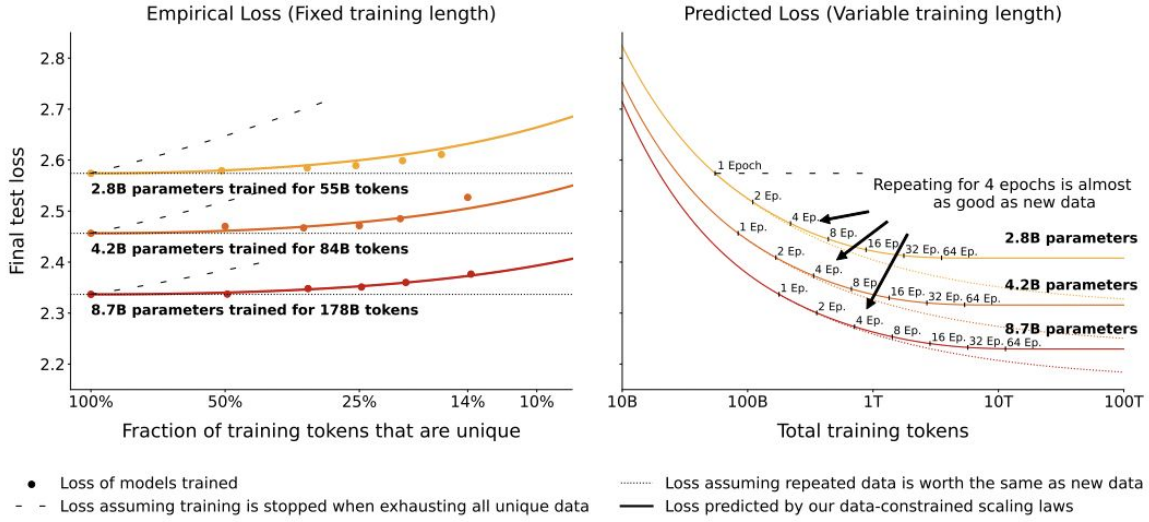


Results


2.8B parameters trained for 55B tokens
4.2B parameters trained for 84B tokens
8.7B parameters trained for 178B tokens
Validation loss
3.4
3.2
3.0
2.8
2.6
2.4
5B
15B
25B
35B
45B
55B
5B
25B
45B
65B
85B
5B
40B
100B
140B
180B
Training tokens
Training tokens
Training tokens
Epochs
1
2
3
4
5
7
14
44

Results

Allocation is optimized by using compute for more epochs rather than more parameters.



Key Takeaways & Limitations

- **Data Repetition:** Training LLMs for multiple epochs with repeated data is beneficial
- **Scaling Laws:** Proposed extension to Chinchilla scaling that accounts for diminishing returns of repeated data
- **Complementary Approaches**: Code augmentation and data filtering
- **Limitation on Repetition**: Need for efficient use of data

Emergent Abilities of Large Language Models

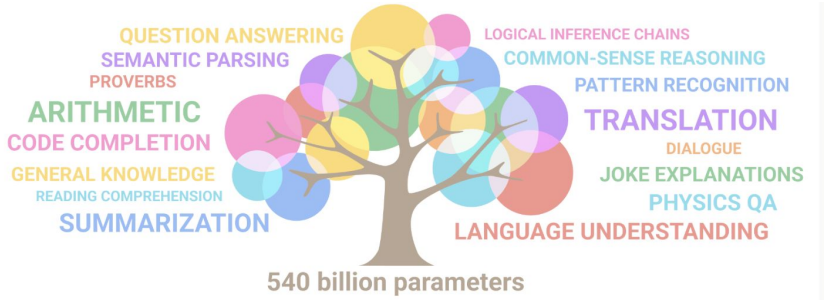
Authors: Jason Wei et al.

Publication Date: Aug 2022

What is Emergence and Why is it Important?

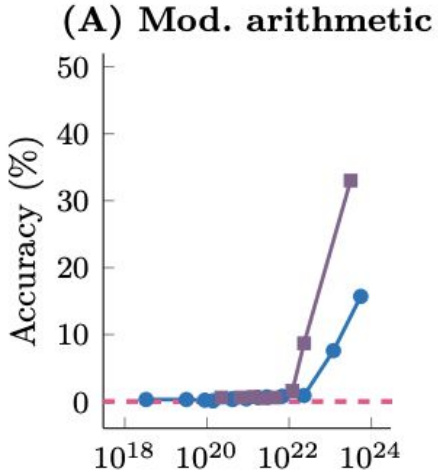
"Emergent abilities of large language models are abilities that are not present in smaller-scale models but are present in large scale models"

- Impossible to predict by extrapolating the performance of smaller scale models
- More scaling may result in new emergent abilities



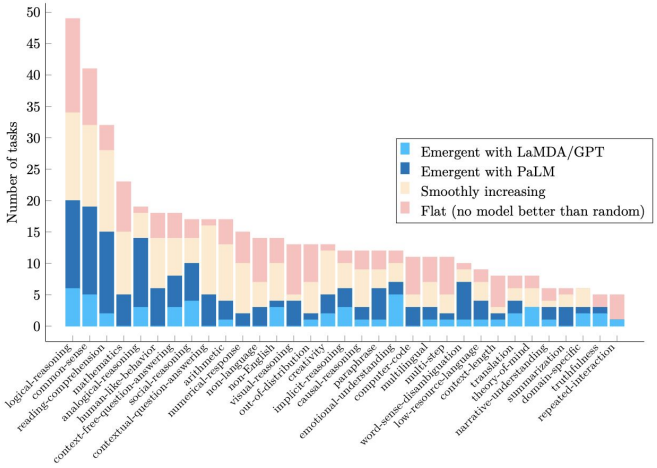
Methodology

- Ran tests on models of different scale in various LLM tasks
- Scale measured in training FLOPs (Floating Point Operations)
  - Number of parameters
  - Size of the training dataset & number of epochs
- Model architecture not significant



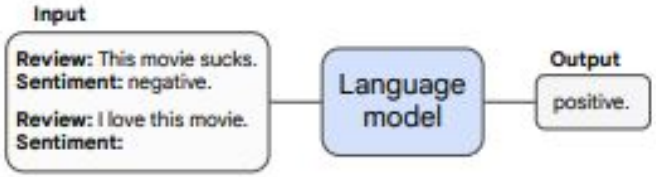
Benchmarks

- **BIG-Bench**
  - 200+ benchmarks for language model evaluation
- **TruthfulQA**
  - Measuring ability to answer questions truthfully
  - Adversarially created against GPT-3 models
- **Massive Multi-task Language Understanding (MMLU)**
  - Wide range of tests requiring deep understanding
  - Small models do not perform better than random

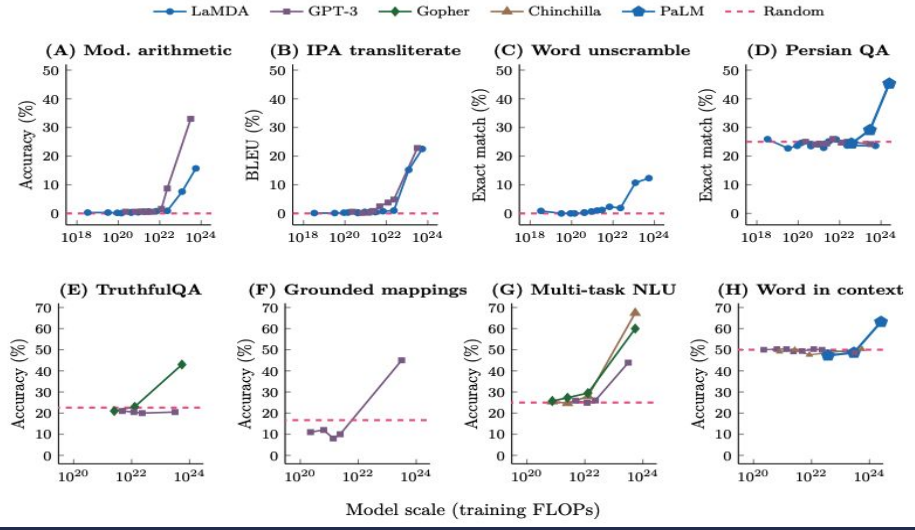


Few Shot Prompted Tasks

Model is given a prompt with a few input-out examples and asked to complete the task without any gradient updates



Example of the Prompting paradigm

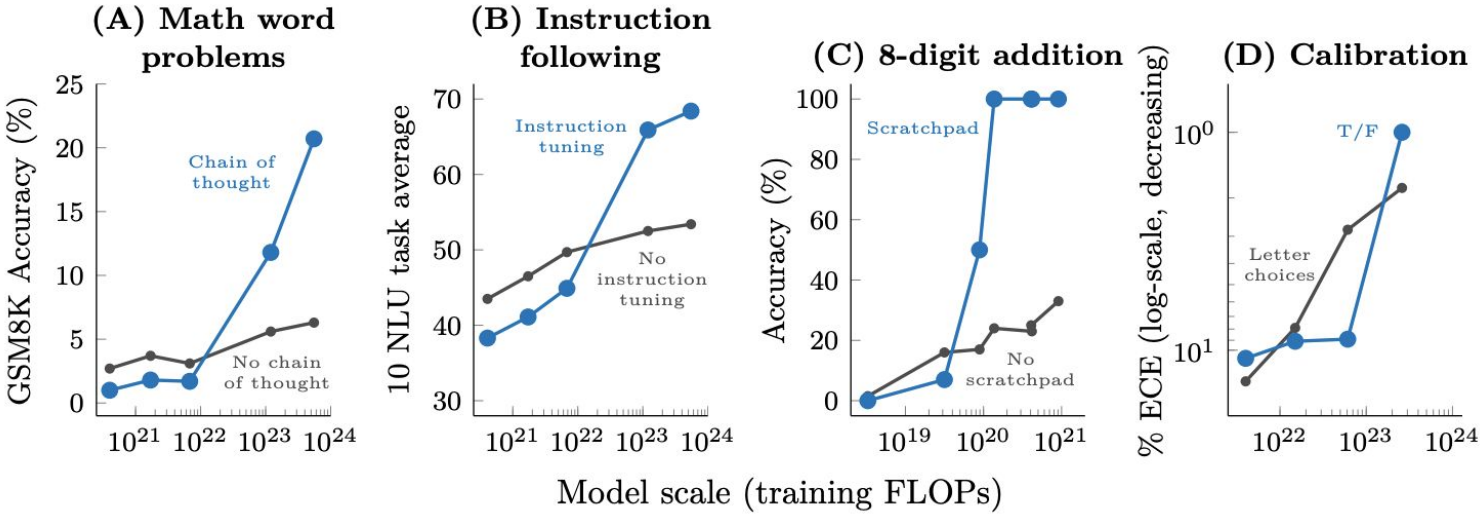


Augmented Prompting Strategies

Prompting/fine-tuning strategies to further improve the abilities of LLMs.

- **Multistep Reasoning**: Chain of thought prompting by guiding LLM to produce a sequence of intermediate events.
- **Instruction Following**: Perform new tasks by reading instructions describing the task.
- **Program Execution or Addition**: Provide a “scratchpad” or a way for the LLM to store intermediate outputs.
- **Model Calibration**: Measure if the model is able to predict which questions it can answer accurately.

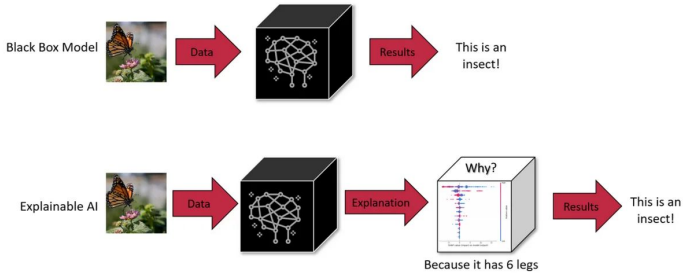
Augmented Prompting Emergence



Black Box Nature of LLMs

Impossible to tell exactly why the model is acting in the way that it is due to the massive scale of LLMs

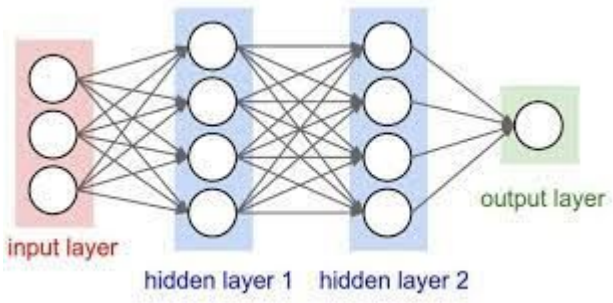
- Difficult to reason emergent abilities
- Emergent Risk may also appear by making a model bigger (TruthfulQA)
  - Untruthfulness, bias, and toxicity can seep into the model
  - Vulnerability and harmful content synthesis



Possible Causes of Emergence

It is very difficult to tell what is really causing these emergent behaviors due to complex interactions.

- Multi-step reasoning may require at least L layers for tasks requiring L steps.
- More parameters/compute allow for better memorization of world knowledge
- Metric chosen may induce emergent abilities

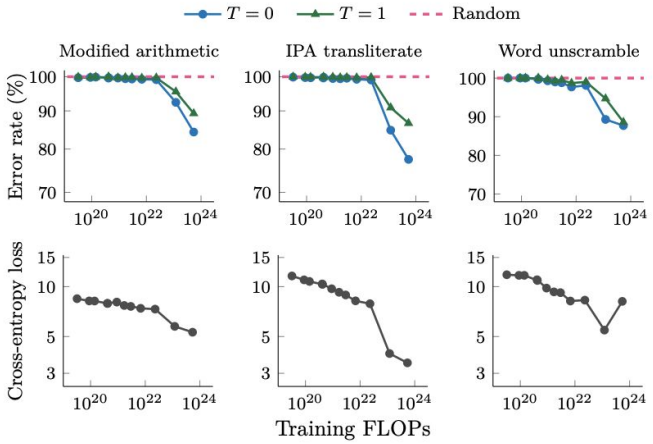


Emergence and Loss

Even as accuracy for an emergent task stays near random, cross entropy loss is steadily decreasing

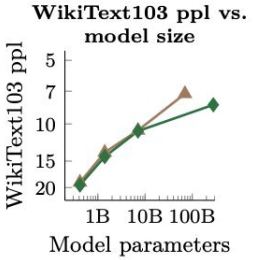
- Loss is different from Exact Match(EM) or accuracy, because it captures improvements in accuracy.
  - One of two wrong answers will have lower loss
- Large jump in loss occurs when emergent ability is noticed

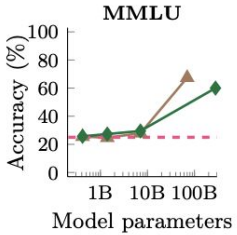
$$L(N, D) = \underbrace{\frac{406.4}{N^{0.34}}}_{\text{finite model}} + \underbrace{\frac{410.7}{D^{0.28}}}_{\text{finite data}} + \underbrace{1.69}_{\text{irreductible}}$$

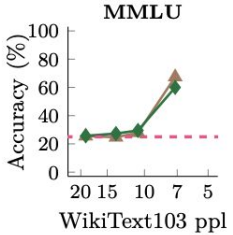


Beyond Scale

- New, smaller models achieve emergent abilities sooner, by using better resources/architecture
- Perplexity of WikiText103 as a indicator of emergent abilities
- Scale may not be the full picture and emergence may arise from complex interactions

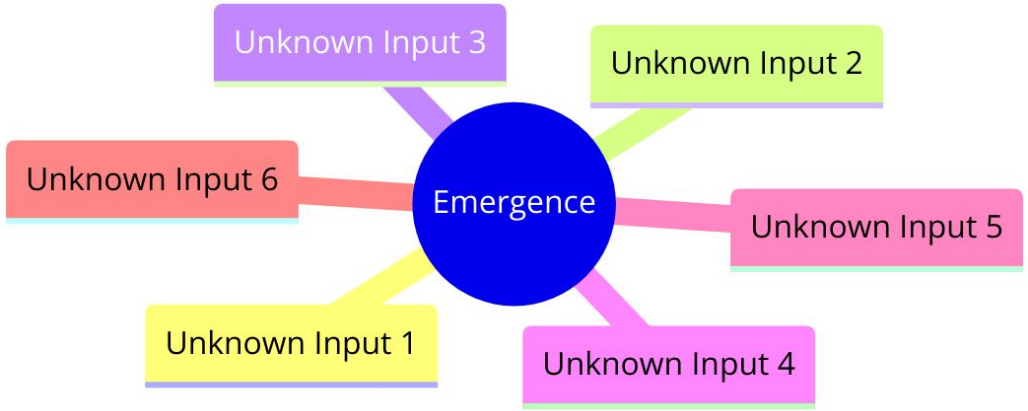






Key Takeaways and Limitations

- Emergence is unpredictable and increasing scale may lead to new emergent abilities
- The real reason emergence occurs is unknown and is likely to be a culmination of different inputs
- Only a small number metrics were tested
- Analysis of loss was not discussed enough



Are Emergent Abilities of LLMs a Mirage?

Authors: Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo

Publication Date: May 2023

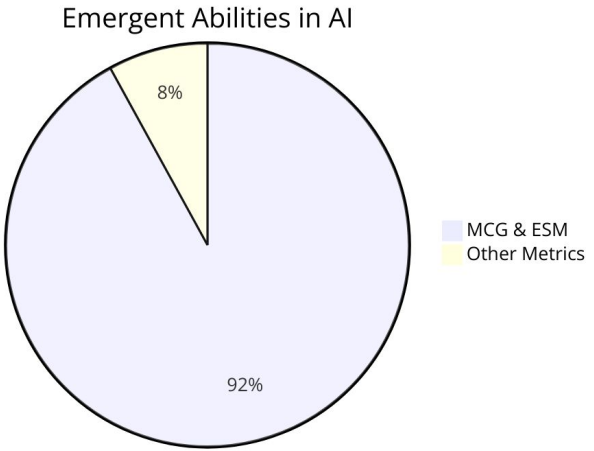
Do Emergent Abilities Really Exist?

The researcher's choice of metric is what creates the mirage that an emergent ability has arised rather than a fundamental change

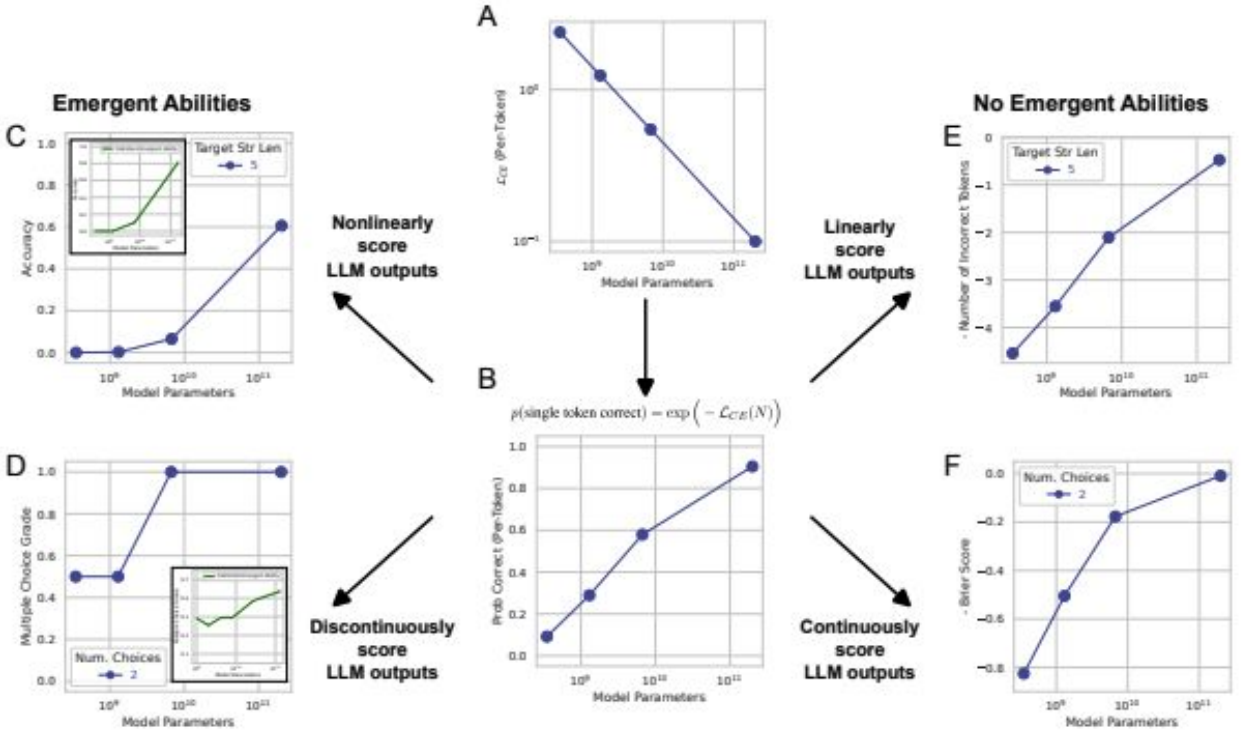
- Nonlinear and discontinuous metrics produce apparent emergent behaviors
- Linear/continuous metrics for the same task create predictable changes in performance
- Emergent abilities go away when we change the metric in use

Metrics

- Exact String Match: Each token in string is exactly correct
- Multiple Choice Grade: Highest probability mass on correct answer
- Non-linear/discontinuous metrics!



Hypothesis


A
B
C
D
E
F
Emergent Abilities
No Emergent Abilities
Nonlinearly score LLM outputs
Linearly score LLM outputs
Discontinuously score LLM outputs
Continuously score LLM outputs
Model Parameters
Accuracy
Target Str Len
Num. Choices
Multiple Choice Grade
Prob Correct (Per-Token)
- Number of Incorrect Tokens
- Brier Score
$p(\text{single token correct}) = \exp\left(-\mathcal{L}_{CE}(N)\right)$

Non linearity of Exact Match

Cross Entropy Loss with Power Law Scaling

$$\mathcal{L}_{CE}(N) = \left(\frac{N}{c}\right)^{\alpha}$$

Per Token Cross Entropy

$$\mathcal{L}_{CE}(N) \overset{\text{def}}{=} -\sum_{v \in V} p(v) \log \hat{p}_N(v)$$

Single token case

$$\mathcal{L}_{CE}(N) = -\log \hat{p}_N(v^*)$$

$$p(\text{single token correct}) = \exp\Big(-\mathcal{L}_{CE}(N)\Big) = \exp\Big(-(N/c)^{\alpha}\Big)$$

Non linearity of Exact Match cont.

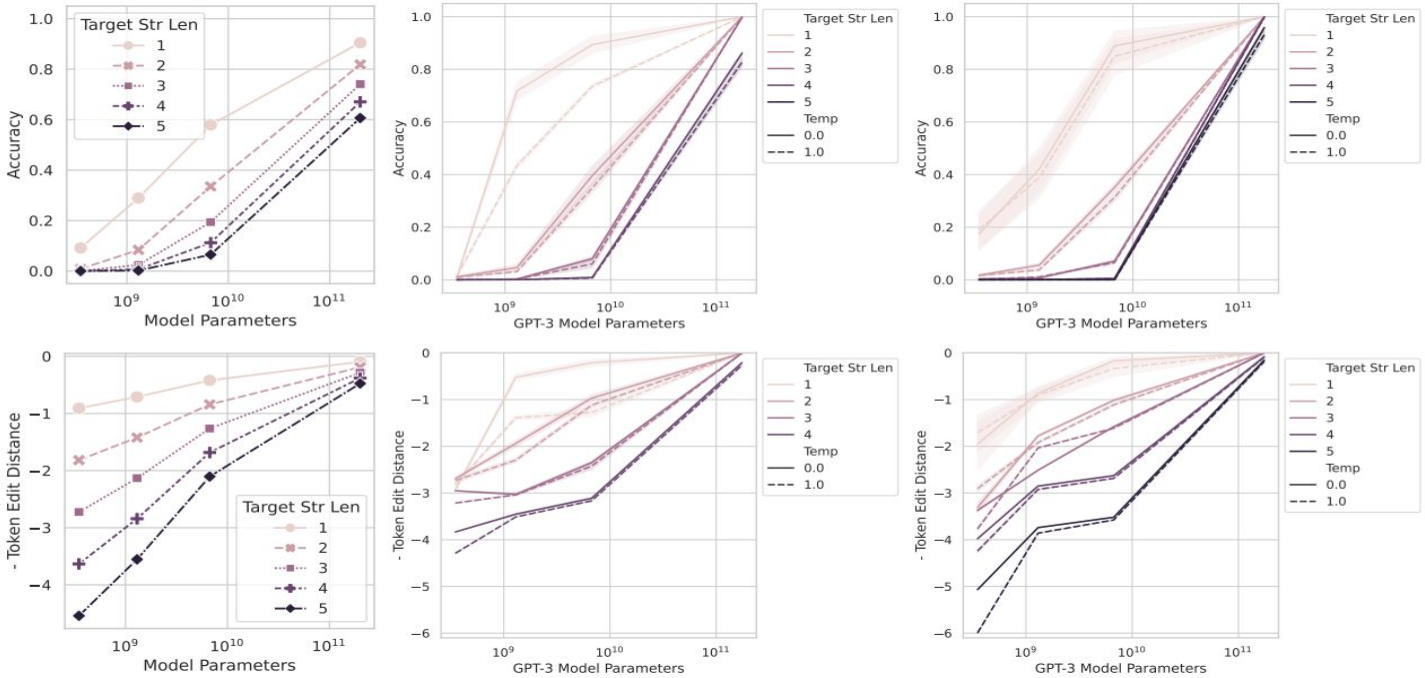
$$\text{Accuracy}(N) \approx p_N(\text{single token correct})^{\text{num. of tokens}} = \exp\left(-(N/c)^{\alpha}\right)^{L}$$

Geometric increase with increasing token length

$$\text{Token Edit Distance}(N) \approx L\left(1 - p_N(\text{single token correct})\right) = L\left(1 - \exp\left(-(N/c)^{\alpha}\right)\right)$$

Linear metric for smooth performance increase

Exact Match vs Token Edit Distance

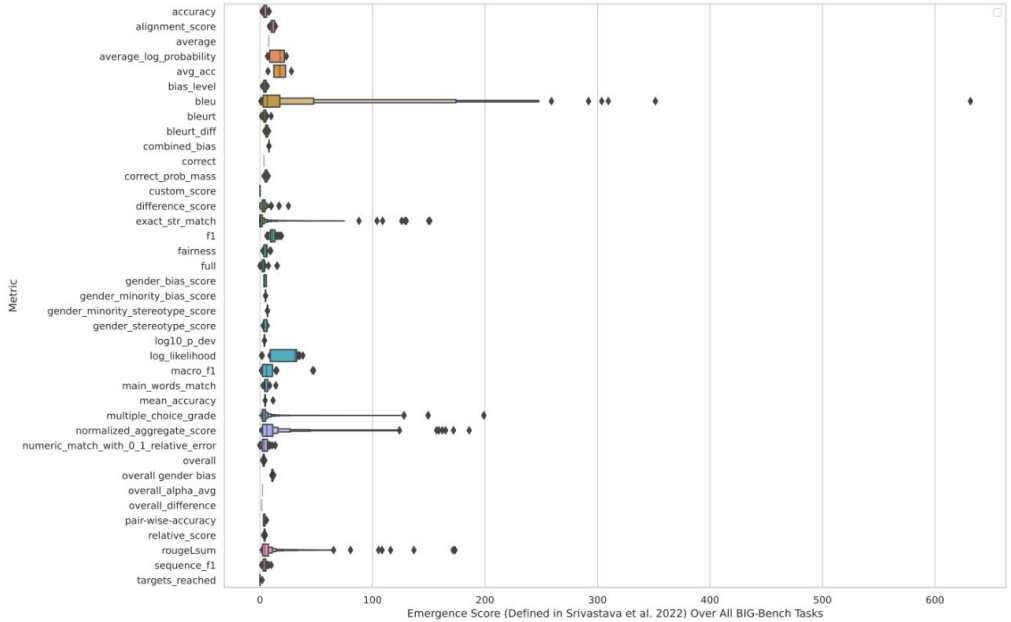


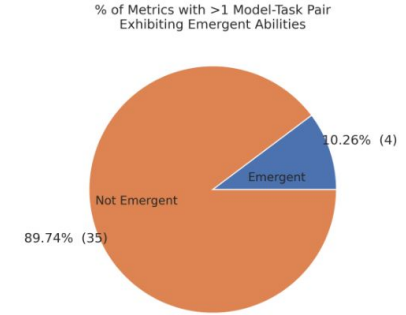
Task-Metric-Model vs Task-Model

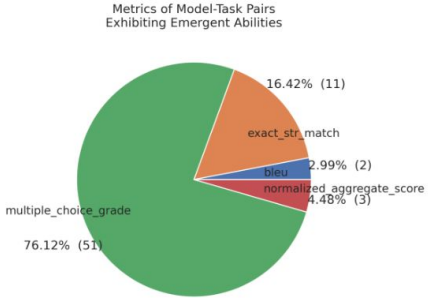
- Task-Metric-Model Triplets should create “emergent behavior”
- Emergent Task-Model pairs are based almost entirely around certain metrics
- If emergent abilities are real, we would expect them to show up for all reasonable metrics

Task-Metric-Model = Addition - Exact Match - GPT-3
Task Model = Addition - GPT3

Overall Metrics







Inducing Emergent Abilities

Researchers focused on inducing emergent abilities on computer vision tasks because emergent capabilities have not been observed in vision models

Emergent Reconstruction by Autoencoders

- New metric resulted in sharp, unpredictable change in performance

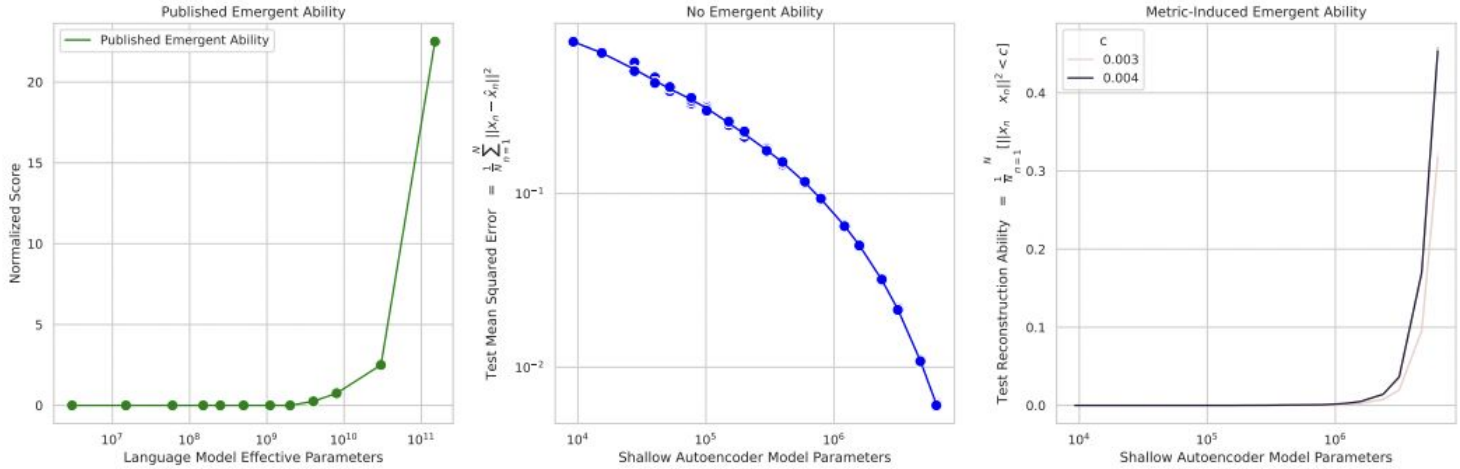
$$\text{Reconstruction}_c\left(\{x_n\}_{n=1}^N\right) \overset{\text{def}}{=} \frac{1}{N}\sum_n \mathbb{I}\left[||x_n - \hat{x}_n||^2 < c\right]$$

Emergent Classification by Transformers

- Increasing accuracy with increase scale
- Metric focused on correct classification of all characters

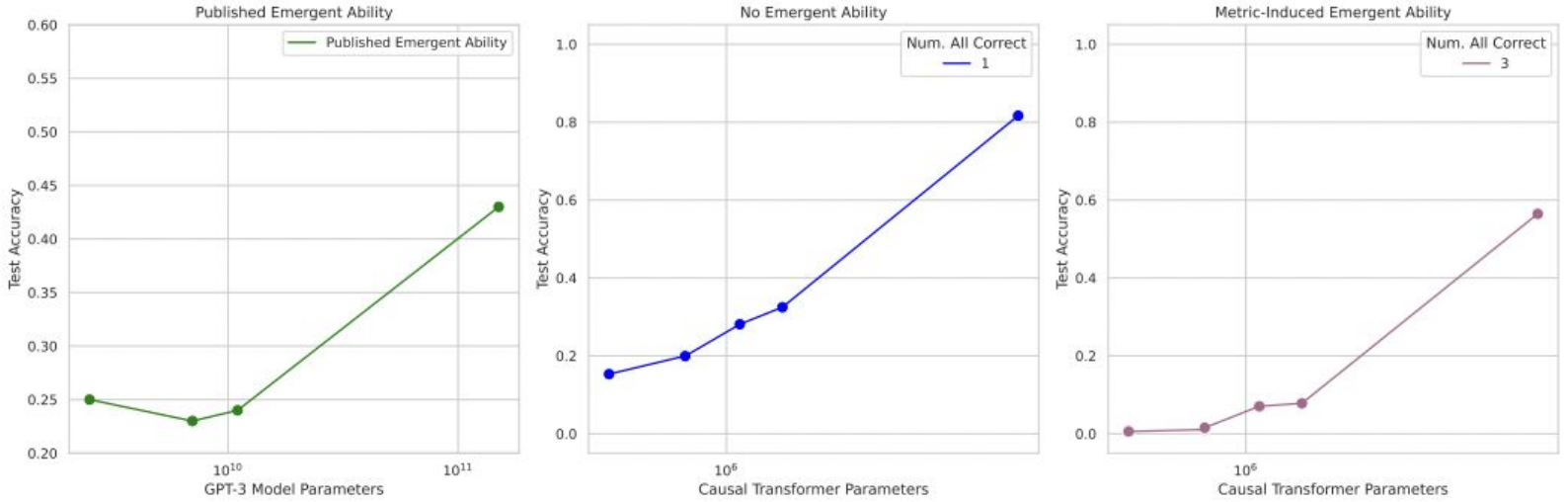
Inducing Emergent Abilities

Reconstruction of natural images by nonlinear autoencoders



Inducing Emergent Abilities

Classification ability in autoregressive transformers



Key Takeaways & Limitations

What are often considered emergent abilities in LLMs may actually be created by the choice of the metrics chosen by researchers

- Challenges the notion of emergent abilities as intrinsic properties of AI models
- Task and metric selection can induce emergent abilities
- Proper controls are must be included to make claims on LLMs
- Necessity of publicly available dataset and models for further testing

Questions?