WIKIPEDIA
The Free Encyclopedia

Large language model

A **large language model** (**LLM**) is a language model consisting of a neural network with many parameters (typically billions of weights or more), trained on large quantities of unlabeled text using self-supervised learning or semi-supervised learning.[1] LLMs emerged around 2018 and perform well at a wide variety of tasks. This has shifted the focus of natural language processing research away from the previous paradigm of training specialized supervised models for specific tasks.[2]

Though the term *large language model* has no formal definition, it often refers to deep learning models having a parameter count on the order of billions or more.[3] LLMs are general purpose models which excel at a wide range of tasks, as opposed to being trained for one specific task (such as sentiment analysis, named entity recognition, or mathematical reasoning).[2][4] The skill with which they accomplish tasks, and the range of tasks at which they are capable, seems to be a function of the amount of resources (data, parameter-size, computing power) devoted to them, in a way that is not dependent on additional breakthroughs in design.[5]

Though trained on simple tasks along the lines of predicting the next word in a sentence, neural language models with sufficient training and parameter counts are found to capture much of the syntax and semantics of human language. In addition, large language models demonstrate considerable general knowledge about the world, and are able to "memorize" a great quantity of facts during training.[2]

Properties

Pretraining datasets

LLMs are pre-trained on large textual datasets. Some commonly used textual datasets are Common Crawl, The Pile, MassiveText,[6] Wikipedia, and GitHub. The datasets run up to 10 trillion words in size.

The stock of high-quality language data is within 4.6-17 trillion words, which is within an order of magnitude for the largest textual datasets.[7]

Scaling laws

In general, a LLM can be characterized by 4 parameters: size of the model, size of the training dataset, cost of training, performance after training. Each of these four variables can be precisely defined into a real number, and they are empirically found to be related by simple statistical laws, called "scaling laws".

One particular scaling law ("Chinchilla scaling") for LLM autoregressively trained for one epoch, with a log-log learning rate schedule, states that:[8]

$$\begin{cases} C = C_0 ND \\ L = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + L_0 \end{cases}$$

where the variables are

- C is the cost of training the model, in FLOPs.

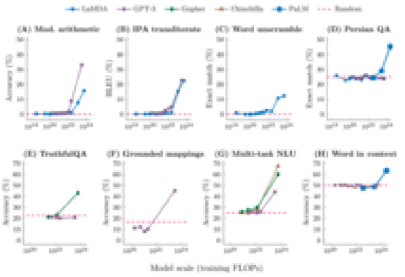
- N is the number of parameters in the model.
- D is the number of tokens in the training set.
- L is the average negative log-likelihood loss per token (nats/token), achieved by the trained LLM on the test dataset.

and the statistical parameters are

- $C_0 = 6$, meaning that it costs 6 FLOPs per parameter to train on one token.[9] Note that training cost is much higher than inference cost, where it costs 1 to 2 FLOPs per parameter to infer on one token.
- $\alpha = 0.34, \beta = 0.28, A = 406.4, B = 410.7, L_0 = 1.69$.

Emergent abilities

While it is generally the case that performance of large models on various tasks can be extrapolated based on the performance of similar smaller models, sometimes "breaks"[10] in downstream scaling laws occur such that larger models suddenly acquire substantial abilities at a different rate than in smaller models. These are often referred to as "emergent abilities", and have been the subject of substantial study. Researchers note that such abilities often "cannot be predicted simply by extrapolating the performance of smaller models".[4] These abilities are discovered rather than programmed-in or designed, in some cases only after the LLM has been publicly deployed.[5] Hundreds of emergent abilities have been described. Examples include multi-step arithmetic, taking college-level exams, identifying the intended meaning of a word,[4] chain-of-thought prompting,[4] decoding the International Phonetic Alphabet, unscrambling a word's letters, identifying offensive content in paragraphs of Hinglish (a combination of Hindi and English), and generating a similar English equivalent of Kiswahili proverbs.[11]



On a number of natural language benchmarks involving tasks such as question answering, models perform no better than random chance until they reach a certain scale (in this case, measured by training computation), at which point their performance sharply increases. These are examples of emergent abilities.

Hallucination

Generative LLMs have been observed to confidently assert claims of fact which do not seem to be justified by their training data, a phenomenon which has been termed "hallucination".[12]

Architecture

Large language models have most commonly used the transformer architecture, which, since 2018, has become the standard deep learning technique for sequential data (previously, recurrent architectures such as the LSTM were most common).[2]

Tokenization

LLMs are mathematical functions whose input and output are lists of numbers. Consequently, words must be converted to numbers.

In general, a LLM uses a separate tokenizer. A tokenizer maps between texts and lists of integers. The tokenizer is generally adapted to the entire training dataset first, then *frozen*, before the LLM is trained. A common choice is byte pair encoding.

Another function of tokenizers is text compression, which saves compute. Common words or phrases like "where is" can be encoded into one token, instead of 7 characters. The OpenAI GPT series uses a tokenizer where 1 token maps to around 4 characters, or around 0.75 words, in common English text.[13] Uncommon English text is less predictable, thus less compressible, thus requiring more tokens to encode.

Tokenizer cannot output arbitrary integers. They generally output only integers in the range $\{0, 1, 2, \ldots, V-1\}$, where V is called its vocabulary size.

Some tokenizers are capable of handling arbitrary text (generally by operating directly on Unicode), but some do not. When encountering un-encodable text, a tokenizer would output a special token (often 0) that represents "unknown text". This is often written as [UNK], such as in the BERT paper.

Another special token commonly used is [PAD] (often 1), for "padding". This is used because LLMs are generally used on batches of text at one time, and these texts do not encode to the same length. Since LLMs generally require input to be an array that is not jagged, the shorter encoded texts must be padded until they match the length of the longest one.

Output

The output of a LLM is a probability distribution over its vocabulary. This is usually implemented as follows:

- Upon receiving a text, the bulk of the LLM outputs a vector $y \in \mathbb{R}^V$ where V is its vocabulary size (defined above).
- The vector y is passed through a softmax function to obtain $softmax(y)$.

In the process, the vector y is usually called the unnormalized logit vector, and the vector $softmax(y)$ is called the probability vector. Since the vector $softmax(y)$ has V entries, all non-negative, and they sum to 1, we can interpret it as a probability distribution over $\{0, 1, 2, \ldots, V-1\}$—that is, it is a probability distribution over the LLM's vocabulary.

Note that the softmax function is defined mathematically with no parameters to vary. Consequently it is not trained.

Training

Most LLM are pre-trained such that given a training dataset of text tokens, the model predicts the tokens in the dataset. There are two general styles of such pretraining:[14]

- autoregressive (GPT-style, "predict the next word"): Given a segment of text like "I like to eat" the model predicts the *next* tokens, like "ice cream".
- masked ("BERT-style",[15] "cloze test"): Given a segment of text like "I like to [MASK] [MASK] cream" the model predicts the masked tokens, like "eat ice".

LLMs may be trained on auxiliary tasks which test their understanding of the data distribution, such as Next Sentence Prediction (NSP), in which pairs of sentences are presented and the model must predict whether they appear consecutively in the training corpus.[15]

Usually, LLMs are trained to minimize a specific loss function: the average negative log likelihood per token (also called cross-entropy loss). For example, if an autoregressive model, given "I like to eat", predicts a probability distribution $Pr(\cdot|\text{I like to eat})$ then the negative log likelihood loss on this token is $-\log Pr(\text{ice}|\text{I like to eat})$.

During training, regularization loss is also used to stabilize training. However regularization loss is usually not used during testing and evaluation. There are also many more evaluation criteria than just negative log likelihood. See the section below for details.

Training dataset size

The earliest LLMs were trained on corpora having on the order of billions of words.

GPT-1, the first model in OpenAI's numbered series of generative pre-trained transformer models, was trained in 2018 on BookCorpus, consisting of 985 million words.[16] In the same year, BERT was trained on a combination of BookCorpus and English Wikipedia, totalling 3.3 billion words.[15] Since then, training corpora for LLMs have increased by orders of magnitude, reaching up to trillions of tokens.[15]

Training cost

LLMs are computationally expensive to train. A 2020 study estimated the cost of training a 1.5 billion parameter model (2 orders of magnitude smaller than the state of the art at the time) at $1.6 million.[17] Advances in software and hardware have brought the cost substantially down, with a 2023 paper reporting a cost of 72,300 A100-GPU-hours to train a 12 billion parameter model.[18]

For Transformer-based LLM, it costs 6 FLOPs per parameter to train on one token.[9] Note that training cost is much higher than inference cost, where it costs 1 to 2 FLOPs per parameter to infer on one token.

Application to downstream tasks

Between 2018 and 2020, the standard method for harnessing an LLM for a specific natural language processing (NLP) task was to fine tune the model with additional task-specific training. It has subsequently been found that more powerful LLMs such as GPT-3 can solve tasks without additional training via "prompting" techniques, in which the problem to be solved is presented to the model as a text prompt, possibly with some textual examples of similar problems and their solutions.[2]

Fine-tuning

Fine-tuning is the practice of modifying an existing pretrained language model by training it (in a supervised fashion) on a specific task (e.g. sentiment analysis, named-entity recognition, or part-of-speech tagging). It is a form of transfer learning. It generally involves the introduction of a new set of weights connecting the final layer of the language model to the output of the downstream task. The original weights of the language model may be "frozen", such that only the new layer of weights connecting them to the output are learned during training. Alternatively, the original weights may receive small updates (possibly with earlier layers frozen).[15]

Prompting

In the prompting paradigm, popularized by GPT-3,[4] the problem to be solved is formulated via a text prompt, which the model must solve by providing a completion (via inference). In "few-shot prompting", the prompt includes a small number of examples of similar (problem, solution) pairs.[2] For example, a sentiment analysis task of labelling the sentiment of a movie review could be prompted as follows:[4]

```
Review: This movie stinks.
Sentiment: negative
```

```
Review: This movie is fantastic!
Sentiment:
```

If the model outputs "positive", then it has correctly solved the task. In zero-shot prompting, no solved examples are provided.[17][19] An example of a zero-shot prompt for the same sentiment analysis task would be "The sentiment associated with the movie review 'This movie is fantastic!' is".[20]

Few-shot performance of LLMs has been shown to achieve competitive results on NLP tasks, sometimes surpassing prior state-of-the-art fine-tuning approaches. Examples of such NLP tasks are translation, question answering, cloze tasks, unscrambling words, and using a novel word in a sentence.[19] The creation and optimisation of such prompts is called prompt engineering.

Instruction tuning

Instruction tuning is a form of fine-tuning designed to facilitate more natural and accurate zero-shot prompting interactions. Given a text input, a pretrained language model will generate a completion which matches the distribution of text on which it was trained. A naive language model given the prompt "Write an essay about the main themes of *Hamlet*." might provide a completion such as "A late penalty of 10% per day will be applied to submissions received after March 17." In instruction tuning, the language model is trained on many examples of tasks formulated as natural language instructions, along with appropriate responses.

Various techniques for instruction tuning have been applied in practice. One example, "self-instruct", fine-tunes the language model on a training set of examples which are themselves generated by an LLM (bootstrapped from a small initial set of human-generated examples).[21]

Reinforcement learning

OpenAI's InstructGPT protocol involves supervised fine-tuning on a dataset of human-generated (prompt, response) pairs, followed by reinforcement learning from human feedback (RLHF), in which a reward model was supervised-learned on a dataset of human preferences, then this reward model was used to train the LLM itself by proximal policy optimization.[22]

Evaluation

Perplexity

The most commonly used measure of a language model's performance is its perplexity on a given text corpus. Perplexity is a measure of how well a model is able to predict the contents of a dataset; the higher the likelihood the model assigns to the dataset, the lower the perplexity. Mathematically, perplexity is defined as the exponential of the average negative log likelihood per token:

$$\log(\text{Perplexity}) = -\frac{1}{N}\sum_{i=1}^{N}\log(Pr(\text{token}_i|\text{context for token}_i))$$

here N is the number of tokens in the text corpus, and "context for token i" depends on the specific type of LLM used. If the LLM is autoregressive, then "context for token i" is the segment of text appearing before token i. If the LLM is masked, then "context for token i" is the segment of text surrounding token i.

Because language models may overfit to their training data, models are usually evaluated by their perplexity on a test set of unseen data.[15] This presents particular challenges for the evaluation of large language models. As they are trained on increasingly large corpora of text largely scraped from the web, it becomes increasingly likely that models' training data inadvertently includes portions of any given test set.[19]

Task-specific datasets and benchmarks

A large number of testing datasets and benchmarks have also been developed to evaluate the capabilities of language models on more specific downstream tasks. Tests may be designed to evaluate a variety of capabilities, including general knowledge, commonsense reasoning, and mathematical problem-solving.

One broad category of evaluation dataset is question answering datasets, consisting of pairs of questions and correct answers, for example, ("Have the San Jose Sharks won the Stanley Cup?", "No").[23] A question answering task is considered "open book" if the model's prompt includes text from which the expected answer can be derived (for example, the previous question could be adjoined with some text which includes the sentence "The Sharks have advanced to the Stanley Cup finals once, losing to the Pittsburgh Penguins in 2016."[23]). Otherwise, the task is considered "closed book", and the model must draw on knowledge retained during training.[24] Some examples of commonly used question answering datasets include TruthfulQA, Web Questions, TriviaQA, and SQuAD.[24]

Evaluation datasets may also take the form of text completion, having the model select the most likely word or sentence to complete a prompt, for example: "Alice was friends with Bob. Alice went to visit her friend, ____".[19]

Some composite benchmarks have also been developed which combine a diversity of different evaluation datasets and tasks. Examples include GLUE, SuperGLUE, MMLU, BIG-bench, and HELM.[25][24]

It was previously standard to report results on a heldout portion of an evaluation dataset after doing supervised fine-tuning on the remainder. It is now more common to evaluate a pre-trained model directly through prompting techniques, though researchers vary in the details of how they formulate prompts for particular tasks, particularly with respect to how many examples of solved tasks are adjoined to the prompt (i.e. the value of *n* in *n*-shot prompting).

Adversarially constructed evaluations

Because of the rapid pace of improvement of large language models, evaluation benchmarks have suffered from short lifespans, with state of the art models quickly "saturating" existing benchmarks, exceeding the performance of human annotators, leading to efforts to replace or augment the benchmark with more challenging tasks.[26]

Some datasets have been constructed adversarially, focusing on particular problems on which extant language models seem to have unusually poor performance compared to humans. One example is the TruthfulQA dataset, a question answering dataset consisting of 817 questions which language models are susceptible to answering incorrectly by mimicking falsehoods to which they were repeatedly exposed during training. For example, an LLM may answer "No" to the question "Can you teach an old dog new tricks?" because of its exposure to the English idiom *you can't teach an old dog new tricks*, even though this is not literally true.[27]

Another example of an adversarial evaluation dataset is Swag and its successor, HellaSwag, collections of problems in which one of multiple options must be selected to complete a text passage. The incorrect completions were generated by sampling from a language model and filtering with a set of classifiers. The resulting problems are trivial for humans but at the time the datasets were created state of the art language models had poor accuracy on them. For example:

> We see a fitness center sign. We then see a man talking to the camera and sitting and laying on a exercise ball. The man...
> a) demonstrates how to increase efficient exercise work by running up and down balls.

b) moves all his arms and legs and builds up a lot of muscle.
c) then plays the ball and we see a graphics and hedge trimming demonstration.

d) performs sits ups while on the ball and talking.[28]

BERT selects b) as the most likely completion, though the correct answer is d).[28]

List of large language models

List of large language models

Name	Release date[a]	Developer	Number of parameters[b]	Corpus size	License[c]	Notes
BERT	2018	Google	340 million[29]	3.3 billion words[29]	Apache 2.0[30]	An early and influential language model,[2] but encoder-only and thus not built to be prompted or generative[31]
XLNet	2019	Google	~340 million[32]	33 billion words		An alternative to BERT; designed as encoder-only[33][34]
GPT-2	2019	OpenAI	1.5 billion[35]	40GB[36] (~10 billion tokens)[37]	MIT[38]	general-purpose model based on transformer architecture
GPT-3	2020	OpenAI	175 billion[17]	300 billion tokens[37]	public web API	A fine-tuned variant of GPT-3, termed GPT-3.5, was made available to the public through a web interface called ChatGPT in 2022.[39]
GPT-Neo	March 2021	EleutherAI	2.7 billion[40]	825 GiB[41]	MIT[42]	The first of a series of free GPT-3 alternatives released by EleutherAI. GPT-Neo outperformed an equivalent-size GPT-3 model on some benchmarks, but was significantly worse than the largest GPT-3.[42]
GPT-J	June 2021	EleutherAI	6 billion[43]	825 GiB[41]	Apache 2.0	GPT-3-style language model
Megatron-Turing NLG	October 2021[44]	Microsoft and Nvidia	530 billion[45]	338.6 billion tokens[45]	Restricted web access	Standard architecture but trained on a supercomputing cluster.
Ernie 3.0 Titan	December 2021	Baidu	260 billion[46]	4 Tb	Proprietary	Chinese-language LLM. Ernie Bot is based on this model.
Claude[47]	December 2021	Anthropic	52 billion[48]	400 billion tokens[48]	Closed beta	Fine-tuned for desirable

						behavior in conversations.[49]
GLaM (Generalist Language Model)	December 2021	Google	1.2 trillion[50]	1.6 trillion tokens[50]	Proprietary	Sparse mixture-of-experts model, making it more expensive to train but cheaper to run inference compared to GPT-3.
Gopher	December 2021	DeepMind	280 billion[51]	300 billion tokens[52]	Proprietary	
LaMDA (Language Models for Dialog Applications)	January 2022	Google	137 billion[53]	1.56T words,[53] 168 billion tokens[52]	Proprietary	Specialized for response generation in conversations.
GPT-NeoX	February 2022	EleutherAI	20 billion[54]	825 GiB[41]	Apache 2.0	based on the Megatron architecture
Chinchilla	March 2022	DeepMind	70 billion[55]	1.4 trillion tokens[55][52]	Proprietary	Reduced-parameter model trained on more data. Used in the Sparrow bot.
PaLM (Pathways Language Model)	April 2022	Google	540 billion[56]	768 billion tokens[55]	Proprietary	aimed to reach the practical limits of model scale
OPT (Open Pretrained Transformer)	May 2022	Meta	175 billion[57]	180 billion tokens[58]	Non-commercial research[d]	GPT-3 architecture with some adaptations from Megatron
YaLM 100B	June 2022	Yandex	100 billion[59]	1.7TB[59]	Apache 2.0	English-Russian model based on Microsoft's Megatron-LM.
Minerva	June 2022	Google	540 billion[60]	38.5B tokens from webpages filtered for mathematical content and from papers submitted to the arXiv preprint server[60]	Proprietary	LLM trained for solving "mathematical and scientific questions using step-by-step reasoning".[61] Minerva is based on PaLM model, further trained on mathematical and scientific data.
BLOOM	July 2022	Large collaboration led by Hugging Face	175 billion[62]	350 billion tokens (1.6TB)[63]	Responsible AI	Essentially GPT-3 but trained on a multi-lingual corpus (30% English excluding programming languages)
Galactica	November 2022	Meta	120 billion	106 billion tokens[64]	CC-BY-NC-4.0	Trained on scientific text and modalities.
AlexaTM (Teacher Models)	November 2022	Amazon	20 billion[65]	1.3 trillion[66]	public web API[67]	bidirectional sequence-to-

						sequence architecture
LLaMA (Large Language Model Meta AI)	February 2023	Meta	65 billion[68]	1.4 trillion[68]	Non-commercial research[e]	Trained on a large 20-language corpus to aim for better performance with fewer parameters.[68] Researchers from Stanford University trained a fine-tuned model based on LLaMA weights, called Alpaca.[69]
GPT-4	March 2023	OpenAI	Exact number unknown, approximately 1 trillion [f]	Unknown	public web API	Available for ChatGPT Plus users and used in several products.
Cerebras-GPT	March 2023	Cerebras	13 billion[71]		Apache 2.0	Trained with Chinchilla formula.
Falcon	March 2023	Technology Innovation Institute	40 billion[72]	1 Trillion tokens (1TB)[72]	Apache 2.0[73]	The model is claimed to use only 75% of GPT-3's training compute, 40% of Chinchilla's, and 80% of PaLM-62B's.
BloombergGPT	March 2023	Bloomberg L.P.	50 billion	363 billion token dataset based on Bloomberg's data sources, plus 345 billion tokens from general purpose datasets[74]	Proprietary	LLM trained on financial data from proprietary sources, that "outperforms existing models on financial tasks by significant margins without sacrificing performance on general LLM benchmarks"
PanGu-Σ	March 2023	Huawei	1.085 trillion	329 billion tokens[75]	Proprietary	
OpenAssistant[76]	March 2023	LAION	17 billion	1.5 trillion tokens	Apache 2.0	Trained on crowdsourced open data
PaLM 2 (Pathways Language Model 2)	May 2023	Google	340 billion[77]	3.6 trillion tokens[77]	Proprietary	Used in Bard chatbot.[78]

See also

- Foundation models

Notes

a. This is the date that documentation describing the model's architecture was first released.

b. In many cases, researchers release or report on multiple versions of a model having different sizes. In these cases, the size of the largest model is listed here.

c. This is the license of the pre-trained model weights. In almost all cases the training code itself is open-source or can be easily replicated.

d. The smaller models including 66B are publicly available, while the 175B model is available on request.

e. Facebook's license and distribution scheme restricted access to approved researchers, but the model weights were leaked and became widely available.

f. As stated in Technical report: "Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method ..."[70] Approximate number in the comparison chart that compares the relative storage, from the same report.

References

1. Goled, Shraddha (May 7, 2021). "Self-Supervised Learning Vs Semi-Supervised Learning: How They Differ" (https://analyticsindiamag.com/self-supervised-learning-vs-semi-supervised-learning-how-they-differ/). *Analytics India Magazine*.
2. Manning, Christopher D. (2022). "Human Language Understanding & Reasoning" (https://www.amacad.org/publication/human-language-understanding-reasoning). *Daedalus*. **151** (2): 127–138. doi:10.1162/daed_a_01905 (https://doi.org/10.1162%2Fdaed_a_01905). S2CID 248377870 (https://api.semanticscholar.org/CorpusID:248377870).
3. Carlini, Nicholas; Tramer, Florian; Wallace, Eric; Jagielski, Matthew; Herbert-Voss, Ariel; Lee, Katherine; Roberts, Adam; Brown, Tom B; Song, Dawn; Erlingsson, Ulfar (2021). *Extracting Training Data from Large Language Models* (https://www.usenix.org/system/files/sec21-carlini-extracting.pdf) (PDF). USENIX Security Symposium. Vol. 6.
4. Wei, Jason; Tay, Yi; Bommasani, Rishi; Raffel, Colin; Zoph, Barret; Borgeaud, Sebastian; Yogatama, Dani; Bosma, Maarten; Zhou, Denny; Metzler, Donald; Chi, Ed H.; Hashimoto, Tatsunori; Vinyals, Oriol; Liang, Percy; Dean, Jeff; Fedus, William (31 August 2022). "Emergent Abilities of Large Language Models" (https://openreview.net/forum?id=yzkSU5zdwD). *Transactions on Machine Learning Research*. ISSN 2835-8856 (https://www.worldcat.org/issn/2835-8856).
5. Bowman, Samuel R. (2023). "Eight Things to Know about Large Language Models" (https://cims.nyu.edu/~sbowman/eightthings.pdf) (PDF). arXiv:2304.00612 (https://arxiv.org/abs/2304.00612).
6. "Papers with Code - MassiveText Dataset" (https://paperswithcode.com/dataset/massivetext). *paperswithcode.com*. Retrieved 2023-04-26.
7. Villalobos, Pablo; Sevilla, Jaime; Heim, Lennart; Besiroglu, Tamay; Hobbhahn, Marius; Ho, Anson (2022-10-25). "Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning". arXiv:2211.04325 (https://arxiv.org/abs/2211.04325) [cs.LG (https://arxiv.org/archive/cs.LG)].
8. Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; Buchatskaya, Elena; Cai, Trevor; Rutherford, Eliza; Casas, Diego de Las; Hendricks, Lisa Anne; Welbl, Johannes; Clark, Aidan; Hennigan, Tom; Noland, Eric; Millican, Katie; Driessche, George van den; Damoc, Bogdan (2022-03-29). "Training Compute-Optimal Large Language Models". arXiv:2203.15556 (https://arxiv.org/abs/2203.15556) [cs.CL (https://arxiv.org/archive/cs.CL)].
9. Kaplan, Jared; McCandlish, Sam; Henighan, Tom; Brown, Tom B.; Chess, Benjamin; Child, Rewon; Gray, Scott; Radford, Alec; Wu, Jeffrey; Amodei, Dario (2020). "Scaling Laws for Neural Language Models". *CoRR*. abs/2001.08361. arXiv:2001.08361 (https://arxiv.org/abs/2001.08361).
10. Caballero, Ethan; Gupta, Kshitij; Rish, Irina; Krueger, David (2022). Broken Neural Scaling Laws. International Conference on Learning Representations (ICLR), 2023.

11. Ornes, Stephen (March 16, 2023). "The Unpredictable Abilities Emerging From Large AI Models" (https://www.quantamagazine.org/the-unpredictable-abilities-emerging-from-large-ai-models-20230316/). *Quanta Magazine*.
12. Ji, Ziwei; Lee, Nayeon; Frieske, Rita; Yu, Tiezheng; Su, Dan; Xu, Yan; Ishii, Etsuko; Bang, Yejin; Dai, Wenliang; Madotto, Andrea; Fung, Pascale (November 2022). "Survey of Hallucination in Natural Language Generation" (https://dl.acm.org/doi/pdf/10.1145/3571730) (pdf). *ACM Computing Surveys*. Association for Computing Machinery. **55** (12): 1–38. arXiv:2202.03629 (https://arxiv.org/abs/2202.03629). doi:10.1145/3571730 (https://doi.org/10.1145%2F3571730). S2CID 246652372 (https://api.semanticscholar.org/CorpusID:246652372). Retrieved 15 January 2023.
13. "OpenAI API" (https://web.archive.org/web/20230423211308/https://platform.openai.com/tokenizer). *platform.openai.com*. Archived from the original (https://platform.openai.com/) on April 23, 2023. Retrieved 2023-04-30.
14. Zaib, Munazza; Sheng, Quan Z.; Emma Zhang, Wei (4 February 2020). "A Short Survey of Pre-trained Language Models for Conversational AI-A New Age in NLP" (https://www.researchgate.net/publication/338931711). *Proceedings of the Australasian Computer Science Week Multiconference*: 1–4. arXiv:2104.10810 (https://arxiv.org/abs/2104.10810). doi:10.1145/3373017.3373028 (https://doi.org/10.1145%2F3373017.3373028). ISBN 9781450376976. S2CID 211040895 (https://api.semanticscholar.org/CorpusID:211040895).
15. Jurafsky, Dan; Martin, James H. (7 January 2023). *Speech and Language Processing* (https://web.stanford.edu/~jurafsky/slp3/ed3book_jan72023.pdf) (PDF) (3rd edition draft ed.). Retrieved 24 May 2022.
16. Zhu, Yukun; Kiros, Ryan; Zemel, Rich; Salakhutdinov, Ruslan; Urtasun, Raquel; Torralba, Antonio; Fidler, Sanja (December 2015). "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books" (https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Zhu_Aligning_Books_and_ICCV_2015_paper.pdf) (PDF). *2015 IEEE International Conference on Computer Vision (ICCV)*: 19–27. arXiv:1506.06724 (https://arxiv.org/abs/1506.06724). doi:10.1109/ICCV.2015.11 (https://doi.org/10.1109%2FICCV.2015.11). ISBN 978-1-4673-8391-2. S2CID 6866988 (https://api.semanticscholar.org/CorpusID:6866988). Retrieved 11 April 2023.
17. Wiggers, Kyle (28 April 2022). "The emerging types of language models and why they matter" (https://techcrunch.com/2022/04/28/the-emerging-types-of-language-models-and-why-they-matter/). *TechCrunch*.
18. Biderman, Stella; Schoelkopf, Hailey; Anthony, Quentin; Bradley, Herbie; Khan, Mohammad Aflah; Purohit, Shivanshu; Prashanth, USVSN Sai (April 2023). "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". arXiv:2304.01373 (https://arxiv.org/abs/2304.01373) [cs.CL (https://arxiv.org/archive/cs.CL)].
19. Brown, Tom B.; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared; Dhariwal, Prafulla; Neelakantan, Arvind; Shyam, Pranav; Sastry, Girish; Askell, Amanda; Agarwal, Sandhini; Herbert-Voss, Ariel; Krueger, Gretchen; Henighan, Tom; Child, Rewon; Ramesh, Aditya; Ziegler, Daniel M.; Wu, Jeffrey; Winter, Clemens; Hesse, Christopher; Chen, Mark; Sigler, Eric; Litwin, Mateusz; Gray, Scott; Chess, Benjamin; Clark, Jack; Berner, Christopher; McCandlish, Sam; Radford, Alec; Sutskever, Ilya; Amodei, Dario (Dec 2020). Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.F.; Lin, H. (eds.). "Language Models are Few-Shot Learners" (https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf) (PDF). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **33**: 1877–1901.
20. Bosma, Maarten; Wei, Jason (6 October 2021). "Introducing FLAN: More generalizable Language Models with Instruction Fine-Tuning" (https://ai.googleblog.com/2021/10/introducing-flan-more-generalizable.html). *Google Research*.
21. Wang, Yizhong; Kordi, Yeganeh; Mishra, Swaroop; Liu, Alisa; Smith, Noah A.; Khashabi, Daniel; Hajishirzi, Hannaneh (2022). "Self-Instruct: Aligning Language Model with Self Generated Instructions". arXiv:2212.10560 (https://arxiv.org/abs/2212.10560) [cs.CL (https://arxiv.org/archive/cs.CL)].

22. Ouyang, Long; Wu, Jeff; Jiang, Xu; Almeida, Diogo; Wainwright, Carroll L.; Mishkin, Pamela; Zhang, Chong; Agarwal, Sandhini; Slama, Katarina; Ray, Alex; Schulman, John; Hilton, Jacob; Kelton, Fraser; Miller, Luke; Simens, Maddie; Askell, Amanda; Welinder, Peter; Christiano, Paul; Leike, Jan; Lowe, Ryan (2022). "Training language models to follow instructions with human feedback". arXiv:2203.02155 (https://arxiv.org/abs/2203.02155) [cs.CL (https://arxiv.org/archive/cs.CL)].
23. Clark, Christopher; Lee, Kenton; Chang, Ming-Wei; Kwiatkowski, Tom; Collins, Michael; Toutanova, Kristina (2019). "BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions". arXiv:1905.10044 (https://arxiv.org/abs/1905.10044) [cs.CL (https://arxiv.org/archive/cs.CL)].
24. Wayne Xin Zhao; Zhou, Kun; Li, Junyi; Tang, Tianyi; Wang, Xiaolei; Hou, Yupeng; Min, Yingqian; Zhang, Beichen; Zhang, Junjie; Dong, Zican; Du, Yifan; Yang, Chen; Chen, Yushuo; Chen, Zhipeng; Jiang, Jinhao; Ren, Ruiyang; Li, Yifan; Tang, Xinyu; Liu, Zikang; Liu, Peiyu; Nie, Jian-Yun; Wen, Ji-Rong (2023). "A Survey of Large Language Models". arXiv:2303.18223 (https://arxiv.org/abs/2303.18223) [cs.CL (https://arxiv.org/archive/cs.CL)].
25. Huyen, Chip (18 October 2019). "Evaluation Metrics for Language Modeling" (https://thegradient.pub/understanding-evaluation-metrics-for-language-models/). *The Gradient*.
26. Srivastava, Aarohi; et al. (2022). "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models". arXiv:2206.04615 (https://arxiv.org/abs/2206.04615) [cs.CL (https://arxiv.org/archive/cs.CL)].
27. Lin, Stephanie; Hilton, Jacob; Evans, Owain (2021). "TruthfulQA: Measuring How Models Mimic Human Falsehoods". arXiv:2109.07958 (https://arxiv.org/abs/2109.07958) [cs.CL (https://arxiv.org/archive/cs.CL)].
28. Zellers, Rowan; Holtzman, Ari; Bisk, Yonatan; Farhadi, Ali; Choi, Yejin (2019). "HellaSwag: Can a Machine Really Finish Your Sentence?". arXiv:1905.07830 (https://arxiv.org/abs/1905.07830) [cs.CL (https://arxiv.org/archive/cs.CL)].
29. Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2 (https://arxiv.org/abs/1810.04805v2) [cs.CL (https://arxiv.org/archive/cs.CL)].
30. "BERT" (https://github.com/google-research/bert). March 13, 2023 – via GitHub.
31. Patel, Ajay; Li, Bryan; Rasooli, Mohammad Sadegh; Constant, Noah; Raffel, Colin; Callison-Burch, Chris (2022). "Bidirectional Language Models Are Also Few-shot Learners". arXiv:2209.14500 (https://arxiv.org/abs/2209.14500) [cs.LG (https://arxiv.org/archive/cs.LG)].
32. "BERT, RoBERTa, DistilBERT, XLNet: Which one to use?" (https://www.kdnuggets.com/bert-roberta-distilbert-xlnet-which-one-to-use.html).
33. Naik, Amit Raja (September 23, 2021). "Google Introduces New Architecture To Reduce Cost Of Transformers" (https://analyticsindiamag.com/google-introduces-new-architecture-to-reduce-cost-of-transformers/). *Analytics India Magazine*.
34. Yang, Zhilin; Dai, Zihang; Yang, Yiming; Carbonell, Jaime; Salakhutdinov, Ruslan; Le, Quoc V. (2 January 2020). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". arXiv:1906.08237 (https://arxiv.org/abs/1906.08237) [cs.CL (https://arxiv.org/archive/cs.CL)].
35. "GPT-2: 1.5B Release" (https://openai.com/blog/gpt-2-1-5b-release/). *OpenAI*. 2019-11-05. Archived (https://web.archive.org/web/20191114074358/https://openai.com/blog/gpt-2-1-5b-release/) from the original on 2019-11-14. Retrieved 2019-11-14.
36. "Better language models and their implications" (https://openai.com/research/better-language-models). *openai.com*.
37. "OpenAI's GPT-3 Language Model: A Technical Overview" (https://lambdalabs.com/blog/demystifying-gpt-3). *lambdalabs.com*. 3 June 2020.
38. "gpt-2" (https://github.com/openai/gpt-2). *GitHub*. Retrieved 13 March 2023.
39. "ChatGPT: Optimizing Language Models for Dialogue" (https://openai.com/blog/chatgpt/). *OpenAI*. 2022-11-30. Retrieved 2023-01-13.
40. "GPT Neo" (https://github.com/EleutherAI/gpt-neo). March 15, 2023 – via GitHub.

41. Gao, Leo; Biderman, Stella; Black, Sid; Golding, Laurence; Hoppe, Travis; Foster, Charles; Phang, Jason; He, Horace; Thite, Anish; Nabeshima, Noa; Presser, Shawn; Leahy, Connor (31 December 2020). "The Pile: An 800GB Dataset of Diverse Text for Language Modeling". arXiv:2101.00027 (https://arxiv.org/abs/2101.00027) [cs.CL (https://arxiv.org/archive/cs.CL)].
42. Iyer, Abhishek (15 May 2021). "GPT-3's free alternative GPT-Neo is something to be excited about" (https://venturebeat.com/ai/gpt-3s-free-alternative-gpt-neo-is-something-to-be-excited-about/). *VentureBeat*.
43. "GPT-J-6B: An Introduction to the Largest Open Source GPT Model | Forefront" (https://www.forefront.ai/blog-posts/gpt-j-6b-an-introduction-to-the-largest-open-sourced-gpt-model). *www.forefront.ai*. Retrieved 2023-02-28.
44. Alvi, Ali; Kharya, Paresh (11 October 2021). "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model" (https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/). *Microsoft Research*.
45. Smith, Shaden; Patwary, Mostofa; Norick, Brandon; LeGresley, Patrick; Rajbhandari, Samyam; Casper, Jared; Liu, Zhun; Prabhumoye, Shrimai; Zerveas, George; Korthikanti, Vijay; Zhang, Elton; Child, Rewon; Aminabadi, Reza Yazdani; Bernauer, Julie; Song, Xia (2022-02-04). "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model". arXiv:2201.11990 (https://arxiv.org/abs/2201.11990).
46. Wang, Shuohuan; Sun, Yu; Xiang, Yang; Wu, Zhihua; Ding, Siyu; Gong, Weibao; Feng, Shikun; Shang, Junyuan; Zhao, Yanbin; Pang, Chao; Liu, Jiaxiang; Chen, Xuyi; Lu, Yuxiang; Liu, Weixin; Wang, Xi; Bai, Yangfan; Chen, Qiuliang; Zhao, Li; Li, Shiyong; Sun, Peng; Yu, Dianhai; Ma, Yanjun; Tian, Hao; Wu, Hua; Wu, Tian; Zeng, Wei; Li, Ge; Gao, Wen; Wang, Haifeng (December 23, 2021). "ERNIE 3.0 Titan: Exploring Larger-scale Knowledge Enhanced Pre-training for Language Understanding and Generation". arXiv:2112.12731 (https://arxiv.org/abs/2112.12731).
47. "Product" (https://www.anthropic.com/product). *Anthropic*. Retrieved 14 March 2023.
48. Askell, Amanda; Bai, Yuntao; Chen, Anna; et al. (9 December 2021). "A General Language Assistant as a Laboratory for Alignment". arXiv:2112.00861 (https://arxiv.org/abs/2112.00861) [cs.CL (https://arxiv.org/archive/cs.CL)].
49. Bai, Yuntao; Kadavath, Saurav; Kundu, Sandipan; et al. (15 December 2022). "Constitutional AI: Harmlessness from AI Feedback". arXiv:2212.08073 (https://arxiv.org/abs/2212.08073) [cs.CL (https://arxiv.org/archive/cs.CL)].
50. Dai, Andrew M; Du, Nan (December 9, 2021). "More Efficient In-Context Learning with GLaM" (https://ai.googleblog.com/2021/12/more-efficient-in-context-learning-with.html). *ai.googleblog.com*. Retrieved 2023-03-09.
51. "Language modelling at scale: Gopher, ethical considerations, and retrieval" (https://www.deepmind.com/blog/language-modelling-at-scale-gopher-ethical-considerations-and-retrieval). *www.deepmind.com*. Retrieved 20 March 2023.
52. Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; et al. (29 March 2022). "Training Compute-Optimal Large Language Models". arXiv:2203.15556 (https://arxiv.org/abs/2203.15556) [cs.CL (https://arxiv.org/archive/cs.CL)].
53. Cheng, Heng-Tze; Thoppilan, Romal (January 21, 2022). "LaMDA: Towards Safe, Grounded, and High-Quality Dialog Models for Everything" (https://ai.googleblog.com/2022/01/lamda-towards-safe-grounded-and-high.html). *ai.googleblog.com*. Retrieved 2023-03-09.
54. Black, Sidney; Biderman, Stella; Hallahan, Eric; et al. (2022-05-01). *GPT-NeoX-20B: An Open-Source Autoregressive Language Model* (https://aclanthology.org/2022.bigscience-1.9/). Proceedings of BigScience Episode #5 -- Workshop on Challenges & Perspectives in Creating Large Language Models. Vol. Proceedings of BigScience Episode #5 -- Workshop on Challenges & Perspectives in Creating Large Language Models. pp. 95–136. Retrieved 2022-12-19.
55. Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; Sifre, Laurent (12 April 2022). "An empirical analysis of compute-optimal large language model training" (https://www.deepmind.com/blog/an-empirical-analysis-of-compute-optimal-large-language-model-training). *Deepmind Blog*.

56. Narang, Sharan; Chowdhery, Aakanksha (April 4, 2022). "Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance" (https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html). *ai.googleblog.com*. Retrieved 2023-03-09.
57. "Democratizing access to large-scale language models with OPT-175B" (https://ai.facebook.com/blog/democratizing-access-to-large-scale-language-models-with-opt-175b/). *ai.facebook.com*.
58. Zhang, Susan; Roller, Stephen; Goyal, Naman; Artetxe, Mikel; Chen, Moya; Chen, Shuohui; Dewan, Christopher; Diab, Mona; Li, Xian; Lin, Xi Victoria; Mihaylov, Todor; Ott, Myle; Shleifer, Sam; Shuster, Kurt; Simig, Daniel; Koura, Punit Singh; Sridhar, Anjali; Wang, Tianlu; Zettlemoyer, Luke (21 June 2022). "OPT: Open Pre-trained Transformer Language Models". arXiv:2205.01068 (https://arxiv.org/abs/2205.01068) [cs.CL (https://arxiv.org/archive/cs.CL)].
59. Khrushchev, Mikhail; Vasilev, Ruslan; Petrov, Alexey; Zinov, Nikolay (2022-06-22), *YaLM 100B* (https://github.com/yandex/YaLM-100B), retrieved 2023-03-18
60. Lewkowycz, Aitor; Andreassen, Anders; Dohan, David; Dyer, Ethan; Michalewski, Henryk; Ramasesh, Vinay; Slone, Ambrose; Anil, Cem; Schlag, Imanol; Gutman-Solo, Theo; Wu, Yuhuai; Neyshabur, Behnam; Gur-Ari, Guy; Misra, Vedant (30 June 2022). "Solving Quantitative Reasoning Problems with Language Models". arXiv:2206.14858 (https://arxiv.org/abs/2206.14858) [cs.CL (https://arxiv.org/archive/cs.CL)].
61. "Minerva: Solving Quantitative Reasoning Problems with Language Models" (https://ai.googleblog.com/2022/06/minerva-solving-quantitative-reasoning.html). *ai.googleblog.com*. 30 June 2022. Retrieved 20 March 2023.
62. Ananthaswamy, Anil (8 March 2023). "In AI, is bigger always better?" (https://www.nature.com/articles/d41586-023-00641-w). *Nature*. **615** (7951): 202–205. Bibcode:2023Natur.615..202A (https://ui.adsabs.harvard.edu/abs/2023Natur.615..202A). doi:10.1038/d41586-023-00641-w (https://doi.org/10.1038%2Fd41586-023-00641-w). PMID 36890378 (https://pubmed.ncbi.nlm.nih.gov/36890378). S2CID 257380916 (https://api.semanticscholar.org/CorpusID:257380916).
63. "bigscience/bloom · Hugging Face" (https://huggingface.co/bigscience/bloom). *huggingface.co*.
64. Taylor, Ross; Kardas, Marcin; Cucurull, Guillem; Scialom, Thomas; Hartshorn, Anthony; Saravia, Elvis; Poulton, Andrew; Kerkez, Viktor; Stojnic, Robert (16 November 2022). "Galactica: A Large Language Model for Science". arXiv:2211.09085 (https://arxiv.org/abs/2211.09085) [cs.CL (https://arxiv.org/archive/cs.CL)].
65. "20B-parameter Alexa model sets new marks in few-shot learning" (https://www.amazon.science/blog/20b-parameter-alexa-model-sets-new-marks-in-few-shot-learning). *Amazon Science*. 2 August 2022.
66. Soltan, Saleh; Ananthakrishnan, Shankar; FitzGerald, Jack; et al. (3 August 2022). "AlexaTM 20B: Few-Shot Learning Using a Large-Scale Multilingual Seq2Seq Model". arXiv:2208.01448 (https://arxiv.org/abs/2208.01448) [cs.CL (https://arxiv.org/archive/cs.CL)].
67. "AlexaTM 20B is now available in Amazon SageMaker JumpStart | AWS Machine Learning Blog" (https://aws.amazon.com/blogs/machine-learning/alexatm-20b-is-now-available-in-amazon-sagemaker-jumpstart/). *aws.amazon.com*. 17 November 2022. Retrieved 13 March 2023.
68. "Introducing LLaMA: A foundational, 65-billion-parameter large language model" (https://ai.facebook.com/blog/large-language-model-llama-meta-ai/). *Meta AI*. 24 February 2023.
69. "Stanford CRFM" (https://crfm.stanford.edu/2023/03/13/alpaca.html). *crfm.stanford.edu*.
70. "GPT-4 Technical Report" (https://cdn.openai.com/papers/gpt-4.pdf) (PDF). *OpenAI*. 2023. Archived (https://web.archive.org/web/20230314190904/https://cdn.openai.com/papers/gpt-4.pdf) (PDF) from the original on March 14, 2023. Retrieved March 14, 2023.
71. Dey, Nolan (March 28, 2023). "Cerebras-GPT: A Family of Open, Compute-efficient, Large Language Models" (https://www.cerebras.net/blog/cerebras-gpt-a-family-of-open-compute-efficient-large-language-models/). *Cerebras*.
72. "Abu Dhabi-based TII launches its own version of ChatGPT" (https://fastcompanyme.com/news/abu-dhabi-based-tii-launches-its-own-version-of-chatgpt/). *tii.ae*.

73. UAE’s Falcon 40B, World’s Top-Ranked AI Model from Technology Innovation Institute, is Now Royalty-Free (https://www.businesswire.com/news/home/20230531005608/en/UAE’s-Falcon-40B-World’s-Top-Ranked-AI-Model-from-Technology-Innovation-Institute-is-Now-Royalty-Free), 31 May 2023
74. Wu, Shijie; Irsoy, Ozan; Lu, Steven; Dabravolski, Vadim; Dredze, Mark; Gehrmann, Sebastian; Kambadur, Prabhanjan; Rosenberg, David; Mann, Gideon (March 30, 2023). "BloombergGPT: A Large Language Model for Finance". arXiv:2303.17564 (https://arxiv.org/abs/2303.17564).
75. Ren, Xiaozhe; Zhou, Pingyi; Meng, Xinfan; Huang, Xinjing; Wang, Yadao; Wang, Weichao; Li, Pengfei; Zhang, Xiaoda; Podolskiy, Alexander; Arshinov, Grigory; Bout, Andrey; Piontkovskaya, Irina; Wei, Jiansheng; Jiang, Xin; Su, Teng; Liu, Qun; Yao, Jun (March 19, 2023). "PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing". arXiv:2303.10845 (https://arxiv.org/abs/2303.10845).
76. Köpf, Andreas; Kilcher, Yannic; von Rütte, Dimitri; Anagnostidis, Sotiris; Tam, Zhi-Rui; Stevens, Keith; Barhoum, Abdullah; Duc, Nguyen Minh; Stanley, Oliver; Nagyfi, Richárd; ES, Shahul; Suri, Sameer; Glushkov, David; Dantuluri, Arnav; Maguire, Andrew (2023-04-14). "OpenAssistant Conversations -- Democratizing Large Language Model Alignment". arXiv:2304.07327 (https://arxiv.org/abs/2304.07327) [cs.CL (https://arxiv.org/archive/cs.CL)].
77. Elias, Jennifer (16 May 2023). "Google's newest A.I. model uses nearly five times more text data for training than its predecessor" (https://www.cnbc.com/2023/05/16/googles-palm-2-uses-nearly-five-times-more-text-data-than-predecessor.html). *CNBC*. Retrieved 18 May 2023.
78. "Introducing PaLM 2" (https://blog.google/technology/ai/google-palm-2-ai-large-language-model/). *Google*. May 10, 2023.