WIKIPEDIA
The Free Encyclopedia

Large language model

A **large language model** (**LLM**) is a deep-learning-based language model, embodied by an artificial neural network using an enormous amount of "parameters" ("neurons" in its layers with up to tens of millions to billions "weights" between them), that are (pre-)trained on many GPUs in relatively short time due to massive parallel processing of vast amounts of unlabeled texts containing up to trillions of tokens (parts of words) provided by corpora such as Wikipedia Corpus and Common Crawl, using self-supervised learning or semi-supervised learning,[1] resulting in a tokenized vocabulary with a probability distribution. LLMs can be upgraded by using additional GPUs to (pre-)train the model with even more parameters on even vaster amounts of unlabeled texts.[2]

The invention of the transformer algorithm, either unidirectional (such as used by GPT models) or bidirectional (such as used by BERT model), allows for such massively parallel processing.[3] Due to all above, most of the older (specialized) supervised models for specific tasks became outdated.[4]

In an implicit way, LLMs have acquired an embodied knowledge about syntax, semantics and "ontology" inherent in human language corpora, but also inaccuracies and biases present in the corpora.[4]

History

Precursors

The basic idea of LLMs, which is to start with a neural network as black box with randomized weights, using a simple repetitive architecture and (pre-)training it on a large language corpus, was not feasible until the 2010s when use of GPUs had enabled massively parallelized processing, which has gradually replaced the logical AI approach that has relied on symbolic programs.[5][6][7]

Precursors of LLMs included the Elman network,[8] in which a recurrent network was trained on simple sentences like "dog chases man". Then, the (pre-)trained model was used to convert each word into a vector (its 'internal representation'). These vectors were clustered by closeness into a tree. The tree was then found to have a structure. The verbs and nouns each belonged to one large cluster. Within the noun cluster, there are two clusters: inanimates and animates. And so on.

In the 1950s, without the modern GPUs enabling massively parallel processing, the idea to learn natural language by a simple repetitive architecture remained just an idea.[9][10] Later in 1990s, the IBM alignment models[11] for statistical machine translation announced the future success of LLMs.[12] An early work that uses corpus scraped from the Internet for word disambiguation (such as distinguishing "then" and "than") in 2001, used a 1-billion-word corpus, considered huge at the time.[13]

Lead-up to the transformer framework

The earliest "large" language models were built with recurrent architectures such as the long short-term memory (LSTM) (1997). After AlexNet (2012) demonstrated the effectiveness of large neural networks for image recognition, researchers applied large neural networks to other tasks. In 2014, two main techniques were proposed.

- The seq2seq model (380 million parameters) used two LSTMs to perform machine translation,[14] and the same approach was used in[15] (130 million parameters) but with a simplified architecture (GRU).
- The attention mechanism was proposed in 2014 paper by Bahdanau et al.,[16] where a seq2seq model was improved by adding an "attention mechanism" in the middle between the two LSTMs. This is "additive attention", which is not the same attention mechanism (scaled "dot product attention") as in Transformer, but it accomplishes a similar task.[17]

In 2016, Google Translate changed its technology from statistical machine translation to neural machine translation. It was a seq2seq with LSTM and attention. It took them 9 months to reach a higher level of performance than the previous system built over 10 years.[18][19]

The 2017 paper "Attention is all you need"[17] abstracted out the attention mechanism from 2014 paper by Bahdanau et al.,[16] and constructed the Transformer architecture around the attention mechanism. Whereas the seq2seq model have to process an input sequence one at a time like all recurrent networks, the Transformer architecture can be run in parallel over the sequence. This allows much larger models to be trained and used.

BERT and GPT

While there are many models with different names, most have underlying architectures being one of two types: BERT (2018)[20] is a bidirectional Transformer, while GPTs (2018+)[21][22] are unidirectional ("autoregressive") Transformers. These are the main architectures as of 2023.

Origin of the term and disambiguation

While the term of Large Language Models has itself emerged around 2018, it gained visibility in 2019 and 2020, with the release of DistilBERT[23] and Stochastic Parrots[24] papers respectively. Both focused on the "Large-scale pretrained models", citing as an example of LLMs the BERT family, starting at 110M parameters and referring to models in the 340M parameters range as "very large LMs".

Perhaps surprisingly, both cite the pre-transformer RNN-based ELMo - the 2018 architecture that inspired BERT - as the first LLM, given the number of parameters (94M), as well as the size of the pretraining dataset (>1B tokens).[25] Despite the comparable parameter size, the original Transformer is generally not considered as an LLM due to a smaller pretraining dataset (generally estimated in the 100M tokens range).

Overall, due to a smooth scaling in LLM model performance from ~100M parameters to 500B+ parameters and progressive unlocking of emergent capabilities such as multi-lingual translation, arithmetic, or programming code composition, all post-ELMo models are referred to by researchers as LLMs.[26][27][28][29]

Linguistic foundations

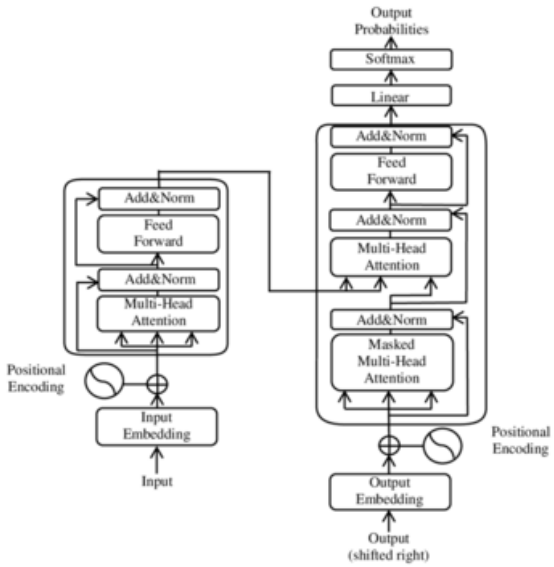
Cognitive Linguistics offers a scientific first principle direction for quantifying states-of-mind through natural language processing[30] to enable a computer to "understand" the contents of text and documents, including the contextual nuances of the language within them. The developmental trajectory of NLP, known as cognitive NLP sets the groundwork as a unitary language model for emulating intelligent behavior and apparent comprehension of natural language. The specialized form of the model considers a block of text, sentence, phrase or word as a token to create a vector database based on tokens presented before and after the token being analyzed. In its generalized form a token can be replaced by any contextually relevant symbol like a group of pixels, mathematical symbols, coding constructs, molecular formulae etc. for non-textual applications.[31]

Architecture

Large language models have most commonly used the transformer architecture, which, since 2018, has become the standard deep learning technique for sequential data.[4] An alternative line of architecture is the mixture of experts (MoE), which is often used in AI models developed by Google, starting with sparsely-gated MoE (2017),[32] and proceeding to Gshard (2021)[33] and GLaM (2022).[34]

All transformers have the same primary components:

- Tokenizers, which convert text into machine-readable symbols known as tokens
- Embedding layers, which convert the machine-readable symbols into semantically meaningful representations
- Transformer layers, which carry out the reasoning capabilities of the models



Transformer model architecture

Transformer layers come in two types known as *encoders* and *decoders*. While the transformer from the original paper was composed of both encoder layers and decoder layers, subsequent work has also explored encoder-only architectures (BERT) and decoder-only architectures (GPT) as well. While all three have their benefits and uses, decoder-only models are the dominant form at very large scales due to being substantially more efficient to train at scale.

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.[35] 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 $\mathbf{softmax}(y)$.

In the process, the vector y is usually called the unnormalized logit vector, and the vector $\mathbf{softmax}(y)$ is called the probability vector. Since the vector $\mathbf{softmax}(y)$ 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.

Context window

The context window of a LLM is the length of the longest sequence of tokens that a LLM can use to generate a token. If a LLM is to generate a token over a sequence longer than the context window, it would have to either truncate the sequence down to the context window, or use certain algorithmic modifications.

The context window of LLM tend to be on the order of 1,000 (1k) to 10k. In particular, OpenAI offers GPT-3.5 with context window from 4k to 16k as of June 2023.[36]

Training

In the pre-training, LLMs may be trained either to predict how the segment continues, or what is missing in the segment, given a segment from its training dataset.[37] It can be either

- autoregressive (i.e. predicting how the segment continues, the way GPTs do it): for example given a segment "I like to eat", the model predicts "ice cream", or
- "masked" (i.e. filling in the parts missing from the segment, the way "BERT"[38] does it): for example, given a segment "I like to `[__]` `[__]` cream", the model predicts that "eat" and "ice" are missing.

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.[38]

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.

Dataset size and compression

In 2018, the BookCorpus, consisting of 985 million words, was used as a training dataset for the OpenAI's first model, GPT-1.[39] In the same year, a combination of BookCorpus and English Wikipedia, totalling 3.3 billion words, was used as a training dataset for BERT.[38] Since then, corpora having up to trillions of tokens were used, increasing previous datasets by orders of magnitude.[38]

Typically, LLM are trained with full- or half-precision floating point numbers (float32 and float16). One float16 has 16 bits, or 2 bytes, and so one billion parameters require 2 gigabytes. The largest models typically have 100 billion parameters, requiring 200 gigabytes to load, which places them outside the range of most consumer electronics.

Post-training quantization[40] aims to decrease the space requirement by lowering precision of the parameters of a trained model, while preserving most of its performance.[41][42] The simplest form of quantization simply truncates all numbers to a given number of bits. It can be improved by using a different quantization codebook per layer. Further improvement can be done by applying different precisions to different parameters, with higher precision for particularly important parameters ("outlier weights").[43]

While quantized models are typically frozen, and only pre-quantized models are finetuned, quantized models can still be finetuned.[44]

Training cost

Advances in software and hardware have reduced the cost substantially since 2020, such that in 2023 training of a 12-billion-parameter LLM computational cost is 72,300 A100-GPU-hours, while in 2020 the cost of training a 1.5-billion-parameter LLM (which was two orders of magnitude smaller than the state of the art in 2020) was between $80 thousand and $1.6 million.[45][46][47] Since 2020, large sums were invested into increasingly large models. For example, training of the GPT-2 (i.e. a 1.5-billion-parameters model) in 2019 cost $50,000, while training of the PaLM (i.e. a 540-billion-parameters model) in 2022 cost $8 million.[48]

For Transformer-based LLM, training cost is much higher than inference cost. It costs 6 FLOPs per parameter to train on one token, whereas it costs 1 to 2 FLOPs per parameter to infer on one token.[49]

Application to downstream tasks

Between 2018 and 2020, the standard method for harnessing an LLM for a specific task was to fine tune the model with additional task-specific training. Only subsequently it has been discovered that LLMs, such as GPT-3, can solve various tasks without being specifically trained to do so. It suffices that they are "prompted", using few examples of similar problems and their respective solutions, instead.[4] Few-shot prompting has sometimes given even better results than the old fine-tuning in the areas of translation, question answering, cloze tasks, unscrambling words, and using a novel word in a sentence.[50] The creation and optimisation of such prompts is called prompt engineering.

From fine-tuning to prompting

The old approach was to fine-tune an existing pretrained language model by re-training (in a supervised fashion) it for a purpose of solving a specific problem (such as sentiment analysis, named-entity recognition, or part-of-speech tagging), which is achieved by introducing 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).[38]

In the new approach called prompting and popularized by GPT-3,[51] a LLM is provided a completion (via inference). In few-shot prompting, for example, the prompt includes a few examples of similar problem-solution pairs.[4]

Below is a sentiment analysis example, labeling the sentiment of a movie review:[51]

```
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.[45][50]

Instruction tuning

Often, instruction tuning is necessary because otherwise an artificial neural network, in response to user 's instruction "Write an essay about the main themes represented in Hamlet," may generate a response such as "If you submit the essay after March 17th, your grade will be reduced by 10% for each day of delay" based on the frequency of this textual sequence in the corpus. It is only through instruction tuning that the model learns what the response should actually contain for specific instructions.

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).[52]

Finetuning by 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.[53]

Tool use

There are certain tasks that, in principle, cannot be solved by any LLM, at least not without the use of external tools or additional software. An example of such a task is responding to the user's input '354 * 139 = ', provided that the LLM has not already encountered a continuation of this calculation in its training corpus. In such cases, the LLM needs to resort to running program code that calculates the result, which can then be included in its response. Another example is 'What is the time now? It is ', where a separate program interpreter would need to execute a code to get system time on the computer, so LLM could include it in its reply.[54][55] This basic strategy can be sophisticated with multiple attempts of generated programs, and other sampling strategies.[56]

Generally, in order to get an LLM to use tools, one must finetune it for tool-use. If the number of tools is finite, then finetuning may be done just once. If the number of tools can grow arbitrarily, as with online API services, then the LLM can be finetuned to be able to read API documentation and call API correctly.[57][58]

A simpler form of tool use is *Retrieval Augmented Generation*: augment an LLM with document retrieval, sometimes using a vector database. Given a query, a document retriever is called to retrieve the most relevant (usually measured by first encoding the query and the documents into vectors, then finding the documents with vectors closest in Euclidean norm to the query vector). The LLM then generates an output based on both the query and the retrieved documents.[59]

Agency

An LLM is a language model, which is not an agent as it has no goal, but it can be used as a component of an intelligent agent.[60] Researchers have described several methods for such integrations.

The ReAct ("Reason+Act") method constructs an agent out of an LLM, using the LLM as a planner. The LLM is prompted to "think out loud". Specifically, the language model is prompted with a textual description of the environment, a goal, a list of possible actions, and a record of the actions and observations so far. It generates one or more thoughts before generating an action, which is then executed in the environment.[61] The linguistic description of the environment given to the LLM planner can even be the LaTeX code of a paper describing the environment.[62]

In the DEPS ("Describe, Explain, Plan and Select") method, an LLM is first connected to the visual world via image descriptions, then it is prompted to produce plans for complex tasks and behaviors based on its pretrained knowledge and environmental feedback it receives.[63]

The Reflexion method[64] constructs an agent that learns over multiple episodes. At the end of each episode, the LLM is given the record of the episode, and prompted to think up "lessons learned", which would help it perform better at a subsequent episode. These "lessons learned" are given to the agent in the subsequent episodes.

Monte Carlo tree search can use an LLM as rollout heuristic. When a programmatic world model is not available, an LLM can also be prompted with a description of the environment to act as world model.[65]

For open-ended exploration, an LLM can be used to score observations for their "interestingness", which can be used as a reward signal to guide a normal (non-LLM) reinforcement learning agent.[66] Alternatively, it can propose increasingly difficult tasks for curriculum learning.[67] Instead of outputting individual actions, an LLM planner can also construct "skills", or functions for complex

action sequences. The skills can be stored and later invoked, allowing increasing levels of abstraction in planning.[67]

LLM-powered agents can keep a long-term memory of its previous contexts, and the memory can be retrieved in the same way as Retrieval Augmented Generation. Multiple such agents can interact socially.[68]

Multimodality

Multimodality means "having several modalities", and a "modality" means a type of input, such as video, image, audio, text, proprioception, etc.[69] There have been many AI models trained specifically to ingest one modality and output another modality, such as AlexNet for image to label,[70] visual question answering for image-text to text,[71] and speech recognition for speech to text. A review article of multimodal LLM is.[72]

A common method to create multimodal models out of an LLM is to "tokenize" the output of a trained encoder. Concretely, one can construct a LLM that can understand images as follows: take a trained LLM, and take a trained image encoder E. Make a small multilayered perceptron f, so that for any image y, the post-processed vector $f(E(y))$ has the same dimensions as an encoded token. That is an "image token". Then, one can interleave text tokens and image tokens. The compound model is then finetuned on an image-text dataset. This basic construction can be applied with more sophistication to improve the model. The image encoder may be frozen to improve stability.[73]

Flamingo demonstrated the effectiveness of the tokenization method, finetuning a pair of pretrained language model and image encoder to perform better on visual question answering than models trained from scratch.[74] Google PaLM model was finetuned into a multimodal model PaLM-E using the tokenization method, and applied to robotic control.[75] LLaMA models have also been turned multimodal using the tokenization method, to allow image inputs,[76] and video inputs.[77]

GPT-4 can use both text and image as inputs,[78] while Google Gemini is expected to be multimodal.[79]

Properties

Pretraining datasets

Large language models (LLMs) are generally pre-trained on vast amounts of textual data that span a wide variety of domains and languages.[80] Some well-known source of pre-training data include Common Crawl, The Pile, MassiveText,[81] Wikipedia, and GitHub. While the majority of open-source LLMs utilize publicly available data, private data may also be used for pre-training.[82] The pre-training data is obtained by preprocessing raw text through various steps, such as de-duplication, filtering out high-toxicity sequences, discarding low-quality data, and more.[83] It is estimated that the stock of language data grows 7% yearly, and the high-quality language data is within 4.6-17 trillion words as of 2022 October.[84] The extensive use of pre-training data in LLMs leads to data contamination,[85] which occurs when the evaluation data is included in the pre-training data, thereby affecting model performance during benchmark evaluation.

Scaling laws and emergent abilities

The following four hyper-parameters characterize a LLM:

- cost of (pre-)training (C),
- size of the artificial neural network itself, such as number of parameters N (i.e. amount of neurons in its layers, amount of weights between them and biases),
- size of its (pre-)training dataset (i.e. number of tokens in corpus, D),
- performance after (pre-)training.

They are 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:[86]

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

where the variables are

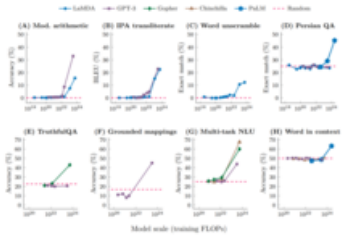
- C is the cost of training the model, in FLOPs.
- N is the number of parameters in the model.
- D is the number of tokens in the training set.

- L is the average negative log-likelihood loss per token (nats/token), achieved by the trained LLM on the test dataset.

and the statistical hyper-parameters are

- $C_0 = 6$, meaning that it costs 6 FLOPs per parameter to train on one token.[49] Note that training cost is much higher than inference cost, where it costs 1 to 2 FLOPs per parameter to infer on one token.
- $\alpha = 0.34, \beta = 0.28, A = 406.4, B = 410.7, L_0 = 1.69$

When one subtracts out from the y-axis the best performance that can be achieved even with infinite scaling of the x-axis quantity, large models' performance, measured on various tasks, seems to be a linear extrapolation of other (smaller-sized and medium-sized) models' performance on a log-log plot. However, sometimes the line's slope transitions from one slope to another at point(s) referred to as break(s)[87] in downstream scaling laws, appearing as a series of linear segments connected by arcs; it seems that larger models acquire "emergent abilities" at this point(s).[51][88] These abilities are discovered rather than programmed-in or designed, in some cases only after the LLM has been publicly deployed.[2]



At point(s) referred to as breaks,[87] the lines change their slopes, appearing on a log-log plot as a series of linear segments connected by arcs.

The emergent abilities include:

- reported arithmetics, decoding the International Phonetic Alphabet, unscrambling a word's letters, disambiguate word in context,[51][89][90] converting spatial words, cardinal directions, and color terms represented in text (for example, replying "northeast" upon [0, 0, 1; 0, 0, 0; 0, 0, 0]),[91] and others.
- chain-of-thought prompting: Model outputs are improved by chain-of-thought prompting only when model size exceeds 62B. Smaller models perform better when prompted to answer immediately, without chain of thought.[92]
- identifying offensive content in paragraphs of Hinglish (a combination of Hindi and English), and generating a similar English equivalent of Kiswahili proverbs.[93]

Schaeffer *et. al.* argue that the emergent abilities are not unpredictably acquired, but predictably acquired according to a smooth scaling law. The authors considered a toy statistical model of an LLM solving multiple-choice questions, and showed that this statistical model, modified to account for other types of tasks, applies to these tasks as well.[31]

Let x be the number of parameter count, and y be the performance of the model.

- When $y = \text{average } Pr(\text{correct token})$, then $(\log x, y)$ is an exponential curve (before it hits the plateau at one), which looks like emergence.
- When $y = \text{average } \log(Pr(\text{correct token}))$, then the $(\log x, y)$ plot is a straight line (before it hits the plateau at zero), which does not look like emergence.
- When $y = \text{average } Pr(\text{the most likely token is correct})$, then $(\log x, y)$ is a step-function, which looks like emergence.

Interpretation

Large language models by themselves are "black boxes", and it is not clear how they can perform linguistic tasks. There are several methods for understanding how LLM work.

Mechanistic interpretability aims to reverse-engineer LLM by discovering symbolic algorithms that approximate the inference performed by LLM. One example is Othello-GPT, where a small Transformer is trained to predict legal Othello moves. It is found that there is a linear representation of Othello board, and modifying the representation changes the predicted legal Othello moves in the correct way.[94][95] In another example, a small Transformer is trained on Karel programs. Similar to the Othello-GPT example, there is a linear representation of Karel program semantics, and modifying the representation changes output in the correct way. The model also generates correct programs that are on average shorter than those in the training set.[96]

In another example, the authors trained small transformers on modular arithmetic addition. The resulting models were reverse-engineered, and it turned out they used discrete Fourier transform.[97]

Understanding and intelligence

NLP researchers were evenly split when asked, in a 2022 survey, whether (untuned) LLMs "could (ever) understand natural language in some nontrivial sense".[98] Proponents of "LLM understanding" believe that some LLM abilities, such as mathematical reasoning, imply an ability to "understand" certain concepts. A Microsoft team argued in 2023 that GPT-4 "can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more" and that GPT-4 "could reasonably be viewed as an early (yet

still incomplete) version of an artificial general intelligence system": "Can one reasonably say that a system that passes exams for software engineering candidates is not *really* intelligent?"[99][100] Some researchers characterize LLMs as "alien intelligence".[101][102] For example, Conjecture CEO Connor Leahy considers untuned LLMs to be like inscrutable alien "Shoggoths", and believes that RLHF tuning creates a "smiling facade" obscuring the inner workings of the LLM: "If you don't push it too far, the smiley face stays on. But then you give it [an unexpected] prompt, and suddenly you see this massive underbelly of insanity, of weird thought processes and clearly non-human understanding."[103][104]

In contrast, some proponents of the "LLMs lack understanding" school believe that existing LLMs are "simply remixing and recombining existing writing",[102] or point to the deficits existing LLMs continue to have in prediction skills, reasoning skills, agency, and explainability.[98] For example, GPT-4 has natural deficits in planning and in real-time learning.[100] 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".[105] Neuroscientist Terrence Sejnowski has argued that "The diverging opinions of experts on the intelligence of LLMs suggests that our old ideas based on natural intelligence are inadequate".[98]

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.[38] 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.[50]

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").[106] 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."[106]). Otherwise, the task is considered "closed book", and the model must draw on knowledge retained during training.[107] Some examples of commonly used question answering datasets include TruthfulQA, Web Questions, TriviaQA, and SQuAD.[107]

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, ____".[50]

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.[108][107]

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.[109] In addition, there are cases of "shortcut learning" wherein AIs sometimes "cheat" on multiple-choice tests by using statistical correlations in superficial test question wording in order to guess the correct responses, without necessarily understanding the actual question being asked.[98]

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.[110]

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.[111]

BERT selects b) as the most likely completion, though the correct answer is d).[111]

Wider impact

In 2023, *Nature Biomedical Engineering* wrote that "it is no longer possible to accurately distinguish" human-written text from text created by large language models, and that "It is all but certain that general-purpose large language models will rapidly proliferate... It is a rather safe bet that they will change many industries over time."[112] Goldman Sachs suggested in 2023 that generative language AI could increase global GDP by 7% in the next ten years, and could expose to automation 300 million jobs globally.[113][114] Some

commenters expressed concern over accidental or deliberate creation of misinformation, or other forms of misuse.[115] For example, the availability of large language models could reduce the skill-level required to commit bioterrorism; biosecurity researcher Kevin Esvelt has suggested that LLM creators should exclude from their training data papers on creating or enhancing pathogens.[116]

List

Name	Release date[a]	Developer	Number of parameters[b]	Corpus size	Training cost (petaFLOP-day)	License[c]	Notes
BERT	2018	Google	340 million[117]	3.3 billion words[117]	9[118]	Apache 2.0[119]	An early and influential language model,[4] but encoder-only and thus not built to be prompted or generative[120]
XLNet	2019	Google	~340 million[121]	33 billion words			An alternative to BERT; designed as encoder-only[122][123]
GPT-2	2019	OpenAI	1.5 billion[124]	40GB[125] (~10 billion tokens)[126]	17[127]	MIT[128]	general-purpose model based on transformer architecture
GPT-3	2020	OpenAI	175 billion[45]	300 billion tokens[126]	3640[129]	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.[130]
GPT-Neo	March 2021	EleutherAI	2.7 billion[131]	825 GiB[132]	90[127]	MIT[133]	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.[133]
GPT-J	June 2021	EleutherAI	6 billion[134]	825 GiB[132]	200[135]	Apache 2.0	GPT-3-style language model
Megatron-Turing NLG	October 2021[136]	Microsoft and Nvidia	530 billion[137]	338.6 billion tokens[137]	16000[127]	Restricted web access	Standard architecture but trained on a supercomputing cluster.
Ernie 3.0 Titan	December 2021	Baidu	260 billion[138]	4 Tb		Proprietary	Chinese-language LLM. Ernie Bot is based on this model.
Claude[139]	December 2021	Anthropic	52 billion[140]	400 billion tokens[140]		Closed beta	Fine-tuned for desirable behavior in conversations.[141]
GLaM (Generalist Language Model)	December 2021	Google	1.2 trillion[34]	1.6 trillion tokens[34]	5600[34]	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[142]	300 billion tokens[143]	5833[144]	Proprietary	
LaMDA (Language Models for Dialog Applications)	January 2022	Google	137 billion[145]	1.56T words,[145] 168 billion tokens[143]	4110[146]	Proprietary	Specialized for response generation in conversations.
GPT-NeoX	February 2022	EleutherAI	20 billion[147]	825 GiB[132]	740[135]	Apache 2.0	based on the Megatron architecture
Chinchilla	March 2022	DeepMind	70 billion[148]	1.4 trillion tokens[148][143]	6805[144]	Proprietary	Reduced-parameter model trained on more data. Used in the Sparrow bot.
PaLM (Pathways Language Model)	April 2022	Google	540 billion[149]	768 billion tokens[148]	29250[144]	Proprietary	aimed to reach the practical limits of model scale
OPT (Open Pretrained Transformer)	May 2022	Meta	175 billion[150]	180 billion tokens[151]	310[135]	Non-commercial research[d]	GPT-3 architecture with some adaptations from Megatron
YaLM 100B	June 2022	Yandex	100 billion[152]	1.7TB[152]	2500[127]	Apache 2.0	English-Russian model based on Microsoft's Megatron-LM.
Minerva	June 2022	Google	540 billion[153]	38.5B tokens from webpages filtered for mathematical content and from papers submitted to the arXiv preprint server[153]	32000[127]	Proprietary	LLM trained for solving "mathematical and scientific questions using step-by-step reasoning".[154] Minerva is based on PaLM model, further trained on mathematical and scientific data.
BLOOM	July 2022	Large collaboration led by Hugging Face	175 billion[155]	350 billion tokens (1.6TB)[156]	2100[127]	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[157]	unknown	CC-BY-NC-4.0	Trained on scientific text and modalities.
AlexaTM (Teacher Models)	November 2022	Amazon	20 billion[158]	1.3 trillion[159]		public web API[160]	bidirectional sequence-to-sequence architecture
LLaMA (Large Language Model Meta AI)	February 2023	Meta	65 billion[161]	1.4 trillion[161]	6300[162][127]	Non-commercial research[e]	Trained on a large 20-language corpus to aim for better performance with fewer parameters.[161] Researchers from Stanford University trained a fine-tuned model based on LLaMA weights, called Alpaca.[163]
GPT-4	March 2023	OpenAI	Exact number unknown, approximately 1 trillion [f]	Unknown	240000 (estimated)[127]	public web API	Available for ChatGPT Plus users and used in several products.
Cerebras-GPT	March 2023	Cerebras	13 billion[165]		270[135]	Apache 2.0	Trained with Chinchilla formula.

Falcon	March 2023	Technology Innovation Institute	40 billion[166]	1 trillion tokens, from RefinedWeb (filtered web text corpus)[167] plus some "curated corpora".[168]	2800[162]	Apache 2.0[169]	Training cost around 2700 petaFLOP-days, 75% that of GPT-3.
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[170]		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[171]		Proprietary	
OpenAssistant[172]	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[173]	3.6 trillion tokens[173]	85000[162]	Proprietary	Used in Bard chatbot.[174]
Llama 2	July 2023	Meta	70 billion[175]	2 trillion tokens[175]		Llama 2 license	Successor of LLaMA.

Further reading

- Jurafsky, Dan, Martin, James. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (https://web.stanford.edu/~jurafsky/slp3/ed3book_jan72023.pdf), 3rd Edition draft, 2023.
- Phuong, Mary; Hutter, Marcus (2022). "Formal Algorithms for Transformers". arXiv:2207.09238 (https://arxiv.org/abs/2207.09238) [cs.LG (https://arxiv.org/archive/cs.LG)].
- Eloundou, Tyna; Manning, Sam; Mishkin, Pamela; Rock, Daniel (2023). "GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models". arXiv:2303.10130 (https://arxiv.org/abs/2303.10130) [econ.GN (https://arxiv.org/archive/econ.GN)].
- Eldan, Ronen; Li, Yuanzhi (2023). "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?". arXiv:2305.07759 (https://arxiv.org/abs/2305.07759) [cs.CL (https://arxiv.org/archive/cs.CL)].
- Frank, Michael C. (27 June 2023). "Baby steps in evaluating the capacities of large language models" (https://www.nature.com/articles/s44159-023-00211-x). *Nature Reviews Psychology*: 1–2. doi:10.1038/s44159-023-00211-x (https://doi.org/10.1038%2Fs44159-023-00211-x). ISSN 2731-0574 (https://www.worldcat.org/issn/2731-0574). S2CID 259713140 (https://api.semanticscholar.org/CorpusID:259713140). Retrieved 2 July 2023.
- Zhao, Wayne Xin; et al. (2023). "A Survey of Large Language Models". arXiv:2303.18223 (https://arxiv.org/abs/2303.18223) [cs.CL (https://arxiv.org/archive/cs.CL)].
- Kaddour, Jean; et al. (2023). "Challenges and Applications of Large Language Models". arXiv:2307.10169 (https://arxiv.org/abs/2307.10169) [cs.CL (https://arxiv.org/archive/cs.CL)].

See also

- Foundation models
- Generative AI

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 ..."[164] 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. Bowman, Samuel R. (2023). "Eight Things to Know about Large Language Models". arXiv:2304.00612 (https://arxiv.org/abs/2304.00612) [cs.CL (https://arxiv.org/archive/cs.CL)].
3. "Better Language Models and Their Implications" (https://openai.com/blog/better-language-models/). *OpenAI*. 2019-02-14. Archived (https://web.archive.org/web/20201219132206/https://openai.com/blog/better-language-models/) from the original on 2020-12-19. Retrieved 2019-08-25.
4. 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).
5. Chomsky, N. (September 1956). "Three models for the description of language" (https://ieeexplore.ieee.org/document/1056813). *IRE Transactions on Information Theory*. **2** (3): 113–124. doi:10.1109/TIT.1956.1056813 (https://doi.org/10.1109%2FTIT.1956.1056813). ISSN 2168-2712 (https://www.worldcat.org/issn/2168-2712). S2CID 19519474 (https://api.semanticscholar.org/CorpusID:19519474).
6. Winograd, Terry (1972-01-01). "Understanding natural language" (https://dx.doi.org/10.1016/0010-0285%2872%2990002-3). *Cognitive Psychology*. **3** (1): 1–191. doi:10.1016/0010-0285(72)90002-3 (https://doi.org/10.1016%2F0010-0285%2872%2990002-3). ISSN 0010-0285 (https://www.worldcat.org/issn/0010-0285).
7. Baker, C. L.; McCarthy, John J., eds. (1981). *The Logical Problem of Language Acquisition*. The MIT Press. ISBN 978-0-262-52389-9.
8. Elman, Jeffrey L. (March 1990). "Finding Structure in Time" (http://doi.wiley.com/10.1207/s15516709cog1402_1). *Cognitive Science*. **14** (2): 179–211. doi:10.1207/s15516709cog1402_1 (https://doi.org/10.1207%2Fs15516709cog1402_1). S2CID 2763403 (https://api.semanticscholar.org/CorpusID:2763403).
9. Shannon, C. E. (January 1951). "Prediction and Entropy of Printed English" (https://dx.doi.org/10.1002/j.1538-7305.1951.tb01366.x). *Bell System Technical Journal*. **30** (1): 50–64. doi:10.1002/j.1538-7305.1951.tb01366.x (https://doi.org/10.1002%2Fj.1538-7305.1951.tb01366.x). ISSN 0005-8580 (https://www.worldcat.org/issn/0005-8580). S2CID 9101213 (https://api.semanticscholar.org/CorpusID:9101213).
10. Miller, George A.; Chomsky, Noam (1963), Luce, D. (ed.), "Finitary Models of Language Users" (https://philpapers.org/rec/MILFMO), *Handbook of Mathematical Psychology*, John Wiley & Sons., pp. 2–419, retrieved 2023-06-27
11. Brown, Peter F. (1993). "The mathematics of statistical machine translation: Parameter estimation". *Computational Linguistics* (19): 263–311.
12. Gal, Yarin; Blunsom, Phil (12 June 2013). "A Systematic Bayesian Treatment of the IBM Alignment Models" (https://web.archive.org/web/20160304071924/http://mlg.eng.cam.ac.uk/yarin/PDFs/PY-IBM_presentation.pdf) (PDF). University of Cambridge. Archived from the original (http://mlg.eng.cam.ac.uk/yarin/PDFs/PY-IBM_presentation.pdf) (PDF) on 4 Mar 2016. Retrieved 26 October 2015.
13. Banko, Michele; Brill, Eric (2001). "Scaling to very very large corpora for natural language disambiguation" (https://dx.doi.org/10.3115/1073012.1073017). *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*. Morristown, NJ, USA: Association for Computational Linguistics: 26–33. doi:10.3115/1073012.1073017 (https://doi.org/10.3115%2F1073012.1073017). S2CID 6645623 (https://api.semanticscholar.org/CorpusID:6645623).
14. Sutskever, Ilya; Vinyals, Oriol; Le, Quoc V (2014). "Sequence to Sequence Learning with Neural Networks" (https://proceedings.neurips.cc/paper_files/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **27**. arXiv:1409.3215 (https://arxiv.org/abs/1409.3215).
15. Cho, Kyunghyun; van Merrienboer, Bart; Bahdanau, Dzmitry; Bengio, Yoshua (2014). "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches" (https://dx.doi.org/10.3115/v1/w14-4012). *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Stroudsburg, PA, USA: Association for Computational Linguistics: 103–111. doi:10.3115/v1/w14-4012 (https://doi.org/10.3115%2Fv1%2Fw14-4012). S2CID 11336213 (https://api.semanticscholar.org/CorpusID:11336213).
16. Bahdanau, Dzmitry; Cho, Kyunghyun; Bengio, Yoshua (2014-09-01). "Neural Machine Translation by Jointly Learning to Align and Translate". arXiv:1409.0473 (https://arxiv.org/abs/1409.0473) [cs.CL (https://arxiv.org/archive/cs.CL)].

17. Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz; Polosukhin, Illia (2017). "Attention is All you Need" (https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **30**.
18. Lewis-Kraus, Gideon (2016-12-14). "The Great A.I. Awakening" (https://web.archive.org/web/20230524052626/https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html). *The New York Times*. ISSN 0362-4331 (https://www.worldcat.org/issn/0362-4331). Archived from the original (https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html) on 24 May 2023. Retrieved 2023-06-22.
19. Wu, Yonghui; et al. (2016-09-01). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". arXiv:1609.08144 (https://arxiv.org/abs/1609.08144) [cs.CL (https://arxiv.org/archive/cs.CL)].
20. 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)].
21. "Improving language understanding with unsupervised learning" (https://openai.com/research/language-unsupervised). *openai.com*. June 11, 2018. Archived (https://web.archive.org/web/20230318210736/https://openai.com/research/language-unsupervised) from the original on 2023-03-18. Retrieved 2023-03-18.
22. *finetune-transformer-lm* (https://github.com/openai/finetune-transformer-lm), OpenAI, June 11, 2018, retrieved 2023-05-01
23. Sanh, Victor; Debut, Lysandre; Chaumond, Julien; Wolf, Thomas (2019-10-02). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter?". arXiv:1910.01108 (https://arxiv.org/abs/1910.01108) [cs.CL (https://arxiv.org/archive/cs.CL)].
24. Emily M., Bender; Gebru, Timnit; McMillan-Major, Angelina; Mitchell, Margaret (2021-03-01). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜" (https://dl.acm.org/doi/10.1145/3442188.3445922). *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: ACM. pp. 610–623. doi:10.1145/3442188.3445922 (https://doi.org/10.1145%2F3442188.3445922).
25. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018). "Deep contextualized word representations". arXiv:1802.05365 (https://arxiv.org/abs/1802.05365) [cs.CL (https://arxiv.org/archive/cs.CL)].
26. Ganguli, Deep; Hernandez, Danny; Lovitt, Liane; et al. (2022-06-20). "Predictability and Surprise in Large Generative Models". *2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM. pp. 1747–1764. doi:10.1145/3531146.3533229 (https://doi.org/10.1145%2F3531146.3533229). ISBN 9781450393522.
27. Kaplan, Jared; McCandlish, Sam; Henighan, Tom; et al. (2020). "Scaling Laws for Neural Language Models". arXiv:2001.08361 (https://arxiv.org/abs/2001.08361) [cs.LG (https://arxiv.org/archive/cs.LG)].
28. Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; et al. (2022). "Training Compute-Optimal Large Language Models". arXiv:2203.15556 (https://arxiv.org/abs/2203.15556) [cs.CL (https://arxiv.org/archive/cs.CL)].
29. Chowdhery, Aakanksha; Narang, Sharan; Devlin, Jacob; et al. (2022). "PaLM: Scaling Language Modeling with Pathways". arXiv:2204.02311 (https://arxiv.org/abs/2204.02311) [cs.CL (https://arxiv.org/archive/cs.CL)].
30. Kjell (2019). "Semantic measures: Using natural language processing to measure, differentiate, and describe psychological constructs" (https://www.researchgate.net/publication/326141366). *Psychological Methods*. **24** (1): 92–115. doi:10.1037/met0000191 (https://doi.org/10.1037%2Fmet0000191). PMID 29963879 (https://pubmed.ncbi.nlm.nih.gov/29963879). S2CID 49642731 (https://api.semanticscholar.org/CorpusID:49642731).
31. Schaeffer, Rylan; Miranda, Brando; Koyejo, Sanmi (2023-04-01). "Are Emergent Abilities of Large Language Models a Mirage?". arXiv:2304.15004 (https://arxiv.org/abs/2304.15004) [cs.AI (https://arxiv.org/archive/cs.AI)].
32. Shazeer, Noam; Mirhoseini, Azalia; Maziarz, Krzysztof; Davis, Andy; Le, Quoc; Hinton, Geoffrey; Dean, Jeff (2017-01-01). "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". arXiv:1701.06538 (https://arxiv.org/abs/1701.06538) [cs.LG (https://arxiv.org/archive/cs.LG)].
33. Lepikhin, Dmitry; Lee, HyoukJoong; Xu, Yuanzhong; Chen, Dehao; Firat, Orhan; Huang, Yanping; Krikun, Maxim; Shazeer, Noam; Chen, Zhifeng (2021-01-12). "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding". arXiv:2006.16668 (https://arxiv.org/abs/2006.16668) [cs.CL (https://arxiv.org/archive/cs.CL)].
34. 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.
35. "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.
36. "OpenAI API" (https://archive.is/koLNQ). *platform.openai.com*. Archived from the original (https://platform.openai.com/) on 16 Jun 2023. Retrieved 2023-06-20.

37. 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).
38. 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.
39. 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)*. pp. 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.
40. Nagel, Markus; Amjad, Rana Ali; Baalen, Mart Van; Louizos, Christos; Blankevoort, Tijmen (2020-11-21). "Up or Down? Adaptive Rounding for Post-Training Quantization" (https://proceedings.mlr.press/v119/nagel20a.html). *Proceedings of the 37th International Conference on Machine Learning*. PMLR: 7197–7206.
41. Polino, Antonio; Pascanu, Razvan; Alistarh, Dan (2018-02-01). "Model compression via distillation and quantization". arXiv:1802.05668 (https://arxiv.org/abs/1802.05668) [cs.NE (https://arxiv.org/archive/cs.NE)].
42. Frantar, Elias; Ashkboos, Saleh; Hoefler, Torsten; Alistarh, Dan (2022-10-01). "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers". arXiv:2210.17323 (https://arxiv.org/abs/2210.17323) [cs.LG (https://arxiv.org/archive/cs.LG)].
43. Dettmers, Tim; Svirschevski, Ruslan; Egiazarian, Vage; Kuznedelev, Denis; Frantar, Elias; Ashkboos, Saleh; Borzunov, Alexander; Hoefler, Torsten; Alistarh, Dan (2023-06-01). "SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression". arXiv:2306.03078 (https://arxiv.org/abs/2306.03078) [cs.CL (https://arxiv.org/archive/cs.CL)].
44. Dettmers, Tim; Pagnoni, Artidoro; Holtzman, Ari; Zettlemoyer, Luke (2023-05-01). "QLoRA: Efficient Finetuning of Quantized LLMs". arXiv:2305.14314 (https://arxiv.org/abs/2305.14314) [cs.LG (https://arxiv.org/archive/cs.LG)].
45. 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*.
46. Sharir, Or; Peleg, Barak; Shoham, Yoav (2020). "The Cost of Training NLP Models: A Concise Overview". arXiv:2004.08900 (https://arxiv.org/abs/2004.08900) [cs.CL (https://arxiv.org/archive/cs.CL)].
47. 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)].
48. Vincent, James (3 April 2023). "AI is entering an era of corporate control" (https://www.theverge.com/23667752/ai-progress-2023-report-stanford-corporate-control). *The Verge*. Retrieved 19 June 2023.
49. 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". arXiv:2001.08361 (https://arxiv.org/abs/2001.08361) [cs.LG (https://arxiv.org/archive/cs.LG)].
50. 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.
51. 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).
52. 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)].
53. 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)].

54. Gao, Luyu; Madaan, Aman; Zhou, Shuyan; Alon, Uri; Liu, Pengfei; Yang, Yiming; Callan, Jamie; Neubig, Graham (2022-11-01). "PAL: Program-aided Language Models". arXiv:2211.10435 (https://arxiv.org/abs/2211.10435) [cs.CL (https://arxiv.org/archive/cs.CL)].
55. "PAL: Program-aided Language Models" (https://reasonwithpal.com/). *reasonwithpal.com*. Retrieved 2023-06-12.
56. Paranjape, Bhargavi; Lundberg, Scott; Singh, Sameer; Hajishirzi, Hannaneh; Zettlemoyer, Luke; Tulio Ribeiro, Marco (2023-03-01). "ART: Automatic multi-step reasoning and tool-use for large language models". arXiv:2303.09014 (https://arxiv.org/abs/2303.09014) [cs.CL (https://arxiv.org/archive/cs.CL)].
57. Liang, Yaobo; Wu, Chenfei; Song, Ting; Wu, Wenshan; Xia, Yan; Liu, Yu; Ou, Yang; Lu, Shuai; Ji, Lei; Mao, Shaoguang; Wang, Yun; Shou, Linjun; Gong, Ming; Duan, Nan (2023-03-01). "TaskMatrix.AI: Completing Tasks by Connecting Foundation Models with Millions of APIs". arXiv:2303.16434 (https://arxiv.org/abs/2303.16434) [cs.AI (https://arxiv.org/archive/cs.AI)].
58. Patil, Shishir G.; Zhang, Tianjun; Wang, Xin; Gonzalez, Joseph E. (2023-05-01). "Gorilla: Large Language Model Connected with Massive APIs". arXiv:2305.15334 (https://arxiv.org/abs/2305.15334) [cs.CL (https://arxiv.org/archive/cs.CL)].
59. Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman; Küttler, Heinrich; Lewis, Mike; Yih, Wen-tau; Rocktäschel, Tim; Riedel, Sebastian; Kiela, Douwe (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" (https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **33**: 9459–9474. arXiv:2005.11401 (https://arxiv.org/abs/2005.11401).
60. Huang, Wenlong; Abbeel, Pieter; Pathak, Deepak; Mordatch, Igor (2022-06-28). "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents" (https://proceedings.mlr.press/v162/huang22a.html). *Proceedings of the 39th International Conference on Machine Learning*. PMLR: 9118–9147. arXiv:2201.07207 (https://arxiv.org/abs/2201.07207).
61. Yao, Shunyu; Zhao, Jeffrey; Yu, Dian; Du, Nan; Shafran, Izhak; Narasimhan, Karthik; Cao, Yuan (2022-10-01). "ReAct: Synergizing Reasoning and Acting in Language Models". arXiv:2210.03629 (https://arxiv.org/abs/2210.03629) [cs.CL (https://arxiv.org/archive/cs.CL)].
62. Wu, Yue; Prabhumoye, Shrimai; Min, So Yeon (24 May 2023). "SPRING: GPT-4 Out-performs RL Algorithms by Studying Papers and Reasoning". arXiv:2305.15486 (https://arxiv.org/abs/2305.15486) [cs.AI (https://arxiv.org/archive/cs.AI)].
63. Wang, Zihao; Cai, Shaofei; Liu, Anji; Ma, Xiaojian; Liang, Yitao (2023-02-03). "Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents". arXiv:2302.01560 (https://arxiv.org/abs/2302.01560) [cs.AI (https://arxiv.org/archive/cs.AI)].
64. Shinn, Noah; Cassano, Federico; Labash, Beck; Gopinath, Ashwin; Narasimhan, Karthik; Yao, Shunyu (2023-03-01). "Reflexion: Language Agents with Verbal Reinforcement Learning". arXiv:2303.11366 (https://arxiv.org/abs/2303.11366) [cs.AI (https://arxiv.org/archive/cs.AI)].
65. Hao, Shibo; Gu, Yi; Ma, Haodi; Jiahua Hong, Joshua; Wang, Zhen; Zhe Wang, Daisy; Hu, Zhiting (2023-05-01). "Reasoning with Language Model is Planning with World Model". arXiv:2305.14992 (https://arxiv.org/abs/2305.14992) [cs.CL (https://arxiv.org/archive/cs.CL)].
66. Zhang, Jenny; Lehman, Joel; Stanley, Kenneth; Clune, Jeff (2 June 2023). "OMNI: Open-endedness via Models of human Notions of Interestingness". arXiv:2306.01711 (https://arxiv.org/abs/2306.01711) [cs.AI (https://arxiv.org/archive/cs.AI)].
67. "Voyager | An Open-Ended Embodied Agent with Large Language Models" (https://voyager.minedojo.org/). *voyager.minedojo.org*. Retrieved 2023-06-09.
68. Park, Joon Sung; O'Brien, Joseph C.; Cai, Carrie J.; Ringel Morris, Meredith; Liang, Percy; Bernstein, Michael S. (2023-04-01). "Generative Agents: Interactive Simulacra of Human Behavior". arXiv:2304.03442 (https://arxiv.org/abs/2304.03442) [cs.HC (https://arxiv.org/archive/cs.HC)].
69. Kiros, Ryan; Salakhutdinov, Ruslan; Zemel, Rich (2014-06-18). "Multimodal Neural Language Models" (https://proceedings.mlr.press/v32/kiros14.html). *Proceedings of the 31st International Conference on Machine Learning*. PMLR: 595–603.
70. Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **25**.
71. Antol, Stanislaw; Agrawal, Aishwarya; Lu, Jiasen; Mitchell, Margaret; Batra, Dhruv; Zitnick, C. Lawrence; Parikh, Devi (2015). "VQA: Visual Question Answering" (https://openaccess.thecvf.com/content_iccv_2015/html/Antol_VQA_Visual_Question_ICCV_2015_paper.html): 2425–2433.
72. Yin, Shukang; Fu, Chaoyou; Zhao, Sirui; Li, Ke; Sun, Xing; Xu, Tong; Chen, Enhong (2023-06-01). "A Survey on Multimodal Large Language Models". arXiv:2306.13549 (https://arxiv.org/abs/2306.13549) [cs.CV (https://arxiv.org/archive/cs.CV)].
73. Li, Junnan; Li, Dongxu; Savarese, Silvio; Hoi, Steven (2023-01-01). "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". arXiv:2301.12597 (https://arxiv.org/abs/2301.12597) [cs.CV (https://arxiv.org/archive/cs.CV)].

74. Alayrac, Jean-Baptiste; Donahue, Jeff; Luc, Pauline; Miech, Antoine; Barr, Iain; Hasson, Yana; Lenc, Karel; Mensch, Arthur; Millican, Katherine; Reynolds, Malcolm; Ring, Roman; Rutherford, Eliza; Cabi, Serkan; Han, Tengda; Gong, Zhitao (2022-12-06). "Flamingo: a Visual Language Model for Few-Shot Learning" (https://proceedings.neurips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html). *Advances in Neural Information Processing Systems*. **35**: 23716–23736. arXiv:2204.14198 (https://arxiv.org/abs/2204.14198).
75. Driess, Danny; Xia, Fei; Sajjadi, Mehdi S. M.; Lynch, Corey; Chowdhery, Aakanksha; Ichter, Brian; Wahid, Ayzaan; Tompson, Jonathan; Vuong, Quan; Yu, Tianhe; Huang, Wenlong; Chebotar, Yevgen; Sermanet, Pierre; Duckworth, Daniel; Levine, Sergey (2023-03-01). "PaLM-E: An Embodied Multimodal Language Model". arXiv:2303.03378 (https://arxiv.org/abs/2303.03378) [cs.LG (https://arxiv.org/archive/cs.LG)].
76. Liu, Haotian; Li, Chunyuan; Wu, Qingyang; Lee, Yong Jae (2023-04-01). "Visual Instruction Tuning". arXiv:2304.08485 (https://arxiv.org/abs/2304.08485) [cs.CV (https://arxiv.org/archive/cs.CV)].
77. Zhang, Hang; Li, Xin; Bing, Lidong (2023-06-01). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". arXiv:2306.02858 (https://arxiv.org/abs/2306.02858) [cs.CL (https://arxiv.org/archive/cs.CL)].
78. OpenAI (2023-03-27). "GPT-4 Technical Report". arXiv:2303.08774 (https://arxiv.org/abs/2303.08774) [cs.CL (https://arxiv.org/archive/cs.CL)].
79. Pichai, Sundar, *Google Keynote (Google I/O '23)* (https://www.youtube.com/watch?v=cNfINi5CNbY&t=931s), timestamp 15:31, retrieved 2023-07-02
80. Anil, Rohan; et al. (2023). "PaLM 2 Technical Report". arXiv:2305.10403 (https://arxiv.org/abs/2305.10403) [cs.CL (https://arxiv.org/archive/cs.CL)].
81. "Papers with Code - MassiveText Dataset" (https://paperswithcode.com/dataset/massivetext). *paperswithcode.com*. Retrieved 2023-04-26.
82. Wu, Shijie; Irsoy, Ozan; Lu, Steven; Dabravolski, Vadim; Dredze, Mark; Gehrmann, Sebastian; Kambadur, Prabhanjan; Rosenberg, David; Mann, Gideon (2023). "BloombergGPT: A Large Language Model for Finance". arXiv:2303.17564 (https://arxiv.org/abs/2303.17564) [cs.LG (https://arxiv.org/archive/cs.LG)].
83. Dodge, Jesse; Sap, Maarten; Marasović, Ana; Agnew, William; Ilharco, Gabriel; Groeneveld, Dirk; Mitchell, Margaret; Gardner, Matt (2021). "Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus". arXiv:2104.08758 (https://arxiv.org/abs/2104.08758) [cs.CL (https://arxiv.org/archive/cs.CL)].
84. 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)].
85. Brown, Tom B.; et al. (2020). "Language Models are Few-Shot Learners". arXiv:2005.14165 (https://arxiv.org/abs/2005.14165) [cs.CL (https://arxiv.org/archive/cs.CL)].
86. 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)].
87. Caballero, Ethan; Gupta, Kshitij; Rish, Irina; Krueger, David (2022). "Broken Neural Scaling Laws". arXiv:2210.14891 (https://arxiv.org/abs/2210.14891) [cs.LG (https://arxiv.org/archive/cs.LG)].
88. "137 emergent abilities of large language models" (https://www.jasonwei.net/blog/emergence). *Jason Wei*. Retrieved 2023-06-24.
89. Pilehvar, Mohammad Taher; Camacho-Collados, Jose (June 2019). "WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations" (https://aclanthology.org/N19-1128). *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics: 1267–1273. doi:10.18653/v1/N19-1128 (https://doi.org/10.18653%2Fv1%2FN19-1128). S2CID 102353817 (https://api.semanticscholar.org/CorpusID:102353817).
90. "WiC: The Word-in-Context Dataset" (https://pilehvar.github.io/wic/). *pilehvar.github.io*. Retrieved 2023-06-27.
91. Patel, Roma; Pavlick, Ellie (2021-10-06). "Mapping Language Models to Grounded Conceptual Spaces" (https://openreview.net/forum?id=gJcEM8sxHK).
92. *A Closer Look at Large Language Models Emergent Abilities (https://www.notion.so/A-Closer-Look-at-Large-Language-Models-Emergent-Abilities-493876b55df5479d80686f68a1abd72f)* (Yao Fu, Nov 20, 2022)
93. 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*.
94. Li, Kenneth; Hopkins, Aspen K.; Bau, David; Viégas, Fernanda; Pfister, Hanspeter; Wattenberg, Martin (2022-10-01). "Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task". arXiv:2210.13382 (https://arxiv.org/abs/2210.13382) [cs.LG (https://arxiv.org/archive/cs.LG)].
95. "Large Language Model: world models or surface statistics?" (https://thegradient.pub/othello/). *The Gradient*. 2023-01-21. Retrieved 2023-06-12.
96. Jin, Charles; Rinard, Martin (2023-05-01). "Evidence of Meaning in Language Models Trained on Programs". arXiv:2305.11169 (https://arxiv.org/abs/2305.11169) [cs.LG (https://arxiv.org/archive/cs.LG)].

97. Nanda, Neel; Chan, Lawrence; Lieberum, Tom; Smith, Jess; Steinhardt, Jacob (2023-01-01). "Progress measures for grokking via mechanistic interpretability". arXiv:2301.05217 (https://arxiv.org/abs/2301.05217) [cs.LG (https://arxiv.org/archive/cs.LG)].
98. Mitchell, Melanie; Krakauer, David C. (28 March 2023). "The debate over understanding in AI's large language models". *Proceedings of the National Academy of Sciences*. **120** (13): e2215907120. arXiv:2210.13966 (https://arxiv.org/abs/2210.13966). Bibcode:2023PNAS..12015907M (https://ui.adsabs.harvard.edu/abs/2023PNAS..12015907M). doi:10.1073/pnas.2215907120 (https://doi.org/10.1073%2Fpnas.2215907120). PMC 10068812. PMID 36943882 (https://pubmed.ncbi.nlm.nih.gov/36943882).
99. Metz, Cade (16 May 2023). "Microsoft Says New A.I. Shows Signs of Human Reasoning" (https://www.nytimes.com/2023/05/16/technology/microsoft-ai-human-reasoning.html). *The New York Times*.
100. Bubeck, Sébastien; Chandrasekaran, Varun; Eldan, Ronen; Gehrke, Johannes; Horvitz, Eric; Kamar, Ece; Lee, Peter; Lee, Yin Tat; Li, Yuanzhi; Lundberg, Scott; Nori, Harsha; Palangi, Hamid; Ribeiro, Marco Tulio; Zhang, Yi (2023). "Sparks of Artificial General Intelligence: Early experiments with GPT-4". arXiv:2303.12712 (https://arxiv.org/abs/2303.12712) [cs.CL (https://arxiv.org/archive/cs.CL)].
101. "ChatGPT is more like an 'alien intelligence' than a human brain, says futurist" (https://www.zdnet.com/article/chatgpt-is-more-like-an-alien-intelligence-than-a-human-brain-says-futurist/). *ZDNET*. 2023. Retrieved 12 June 2023.
102. Newport, Cal (13 April 2023). "What Kind of Mind Does ChatGPT Have?" (https://www.newyorker.com/science/annals-of-artificial-intelligence/what-kind-of-mind-does-chatgpt-have). *The New Yorker*. Retrieved 12 June 2023.
103. Roose, Kevin (30 May 2023). "Why an Octopus-like Creature Has Come to Symbolize the State of A.I." (https://www.nytimes.com/2023/05/30/technology/shoggoth-meme-ai.html) *The New York Times*. Retrieved 12 June 2023.
104. "The A to Z of Artificial Intelligence" (https://time.com/6271657/a-to-z-of-artificial-intelligence/). *Time Magazine*. 13 April 2023. Retrieved 12 June 2023.
105. 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.
106. 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)].
107. 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)].
108. Huyen, Chip (18 October 2019). "Evaluation Metrics for Language Modeling" (https://thegradient.pub/understanding-evaluation-metrics-for-language-models/). *The Gradient*.
109. 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)].
110. 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)].
111. 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)].
112. "Prepare for truly useful large language models". *Nature Biomedical Engineering*. **7** (2): 85–86. 7 March 2023. doi:10.1038/s41551-023-01012-6 (https://doi.org/10.1038%2Fs41551-023-01012-6). PMID 36882584 (https://pubmed.ncbi.nlm.nih.gov/36882584). S2CID 257403466 (https://api.semanticscholar.org/CorpusID:257403466).
113. "Your job is (probably) safe from artificial intelligence" (https://www.economist.com/finance-and-economics/2023/05/07/your-job-is-probably-safe-from-artificial-intelligence). *The Economist*. 7 May 2023. Retrieved 18 June 2023.
114. "Generative AI Could Raise Global GDP by 7%" (https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html). *Goldman Sachs*. Retrieved 18 June 2023.
115. Alba, Davey (1 May 2023). "AI chatbots have been used to create dozens of news content farms" (https://www.japantimes.co.jp/news/2023/05/01/business/tech/ai-fake-news-content-farms/). *The Japan Times*. Retrieved 18 June 2023.
116. "Could chatbots help devise the next pandemic virus?" (https://www.science.org/content/article/could-chatbots-help-devise-next-pandemic-virus). *Science*. 14 June 2023. doi:10.1126/science.adj2463 (https://doi.org/10.1126%2Fscience.adj2463).
117. 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)].
118. Prickett, Nicole Hemsoth (2021-08-24). "Cerebras Shifts Architecture To Meet Massive AI/ML Models" (https://www.nextplatform.com/2021/08/24/cerebras-shifts-architecture-to-meet-massive-ai-ml-models/). *The Next Platform*. Retrieved 2023-06-20.
119. "BERT" (https://github.com/google-research/bert). March 13, 2023 – via GitHub.

120. 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)].
121. "BERT, RoBERTa, DistilBERT, XLNet: Which one to use?" (https://www.kdnuggets.com/bert-roberta-distilbert-xlnet-which-one-to-use.html).
122. 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*.
123. 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)].
124. "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.
125. "Better language models and their implications" (https://openai.com/research/better-language-models). *openai.com*.
126. "OpenAI's GPT-3 Language Model: A Technical Overview" (https://lambdalabs.com/blog/demystifying-gpt-3). *lambdalabs.com*. 3 June 2020.
127. "Parameter, Compute and Data Trends in Machine Learning" (https://docs.google.com/spreadsheets/d/1AAIebjNsnJj_uKALHbXNfn3_YsT6sHXtCU0q7OIPuc4/edit?usp=embed_facebook). *Google Docs*. Retrieved 2023-06-20.
128. "gpt-2" (https://github.com/openai/gpt-2). *GitHub*. Retrieved 13 March 2023.
129. Table D.1 in 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 (May 28, 2020). "Language Models are Few-Shot Learners". arXiv:2005.14165v4 (https://arxiv.org/abs/2005.14165v4) [cs.CL (https://arxiv.org/archive/cs.CL)].
130. "ChatGPT: Optimizing Language Models for Dialogue" (https://openai.com/blog/chatgpt/). *OpenAI*. 2022-11-30. Retrieved 2023-01-13.
131. "GPT Neo" (https://github.com/EleutherAI/gpt-neo). March 15, 2023 – via GitHub.
132. 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)].
133. 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*.
134. "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.
135. Dey, Nolan; Gosal, Gurpreet; Zhiming; Chen; Khachane, Hemant; Marshall, William; Pathria, Ribhu; Tom, Marvin; Hestness, Joel (2023-04-01). "Cerebras-GPT: Open Compute-Optimal Language Models Trained on the Cerebras Wafer-Scale Cluster". arXiv:2304.03208 (https://arxiv.org/abs/2304.03208) [cs.LG (https://arxiv.org/archive/cs.LG)].
136. 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*.
137. 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) [cs.CL (https://arxiv.org/archive/cs.CL)].
138. 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) [cs.CL (https://arxiv.org/archive/cs.CL)].
139. "Product" (https://www.anthropic.com/product). *Anthropic*. Retrieved 14 March 2023.
140. 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)].
141. 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)].

142. "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.
143. 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)].
144. Table 20 of *PaLM: Scaling Language Modeling with Pathways (https://storage.googleapis.com/pathways-language-model/PaLM-paper.pdf)*
145. 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.
146. Thoppilan, Romal; De Freitas, Daniel; Hall, Jamie; Shazeer, Noam; Kulshreshtha, Apoorv; Cheng, Heng-Tze; Jin, Alicia; Bos, Taylor; Baker, Leslie; Du, Yu; Li, YaGuang; Lee, Hongrae; Zheng, Huaixiu Steven; Ghafouri, Amin; Menegali, Marcelo (2022-01-01). "LaMDA: Language Models for Dialog Applications". arXiv:2201.08239 (https://arxiv.org/abs/2201.08239) [cs.CL (https://arxiv.org/archive/cs.CL)].
147. 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.
148. 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*.
149. 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.
150. "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*.
151. 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)].
152. Khrushchev, Mikhail; Vasilev, Ruslan; Petrov, Alexey; Zinov, Nikolay (2022-06-22), *YaLM 100B* (https://github.com/yandex/YaLM-100B), retrieved 2023-03-18
153. 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)].
154. "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.
155. 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).
156. "bigscience/bloom · Hugging Face" (https://huggingface.co/bigscience/bloom). *huggingface.co*.
157. 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)].
158. "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.
159. 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)].
160. "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.
161. "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.
162. "The Falcon has landed in the Hugging Face ecosystem" (https://huggingface.co/blog/falcon). *huggingface.co*. Retrieved 2023-06-20.
163. "Stanford CRFM" (https://crfm.stanford.edu/2023/03/13/alpaca.html). *crfm.stanford.edu*.
164. "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.

165. 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*.
166. "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*.
167. Penedo, Guilherme; Malartic, Quentin; Hesslow, Daniel; Cojocaru, Ruxandra; Cappelli, Alessandro; Alobeidli, Hamza; Pannier, Baptiste; Almazrouei, Ebtesam; Launay, Julien (2023-06-01). "The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only". arXiv:2306.01116 (https://arxiv.org/abs/2306.01116) [cs.CL (https://arxiv.org/archive/cs.CL)].
168. "tiiuae/falcon-40b · Hugging Face" (https://huggingface.co/tiiuae/falcon-40b). *huggingface.co*. 2023-06-09. Retrieved 2023-06-20.
169. 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
170. 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) [cs.LG (https://arxiv.org/archive/cs.LG)].
171. 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) [cs.CL (https://arxiv.org/archive/cs.CL)].
172. 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)].
173. 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.
174. "Introducing PaLM 2" (https://blog.google/technology/ai/google-palm-2-ai-large-language-model/). *Google*. May 10, 2023.
175. "Introducing Llama 2: The Next Generation of Our Open Source Large Language Model" (https://ai.meta.com/llama/). *Meta AI*. 2023. Retrieved 2023-07-19.
