Large Language Models: A Survey

Shervin Minaee[1], Tomas Mikolov[2], Narjes Nikzad[3], Meysam Chenaghlu[4]
Richard Socher[5], Xavier Amatriain[6], Jianfeng Gao[7]

[1] Applied Scientist, Amazon Inc
[2] Senior Researcher, CIIRC CTU
[3] Cologne University of Applied Sciences
[4] Staff Machine Learning Scientist, Ultimate.ai
[5] CEO, You.com
[6] VP of Product, AI and Compute Enablement, Google Inc
[7] VP of Deep Learning Group, Microsoft Research

Abstract—**Large Language Models (LLMs) have drawn a lot of attention due to their strong performance on a wide range of natural language tasks, since the release of ChatGPT in November 2022. LLMs' ability of general-purpose language understanding and generation is acquired by training billions of model's parameters on massive amounts of text data, as predicted by scaling laws [1], [2]. The research area of LLMs, while very recent, is evolving rapidly in many different ways. In this paper, we review some of the most prominent LLMs, including three popular LLM families (GPT, LLaMA, PaLM), and discuss their characteristics, contributions and limitations. We also give an overview of techniques developed to build, and augment LLMs. We then survey popular datasets prepared for LLM training, fine-tuning, and evaluation, review widely used LLM evaluation metrics, and compare the performance of several popular LLMs on a set of representative benchmarks. Finally, we conclude the paper by discussing open challenges and future research directions.**

I. Introduction

Language modeling is a long-standing research topic, dating back to the 1950s with Shannon's application of information theory to human language, where he measured how well simple n-gram language models predict or compress natural language text [3]. Since then, statistical language modeling became fundamental to many natural language understanding and generation tasks, ranging from speech recognition, machine translation, to information retrieval [4], [5], [6].

The recent advances on transformer-based large language models (LMs), pretrained on Web-scale text corpora, significantly extended the capabilities of language models (LLMs). For example, OpenAI's ChatGPT and GPT-4 can be used not only for natural language processing, but also as general task solvers to power Microsoft's Co-Pilot systems, for instance, can follow human instructions of complex new tasks performing multi-step reasoning when needed. LLMs are thus becoming the basic building block for the development of general-purpose AI agents or artificial general intelligence (AGI).

As the field of LLMs is moving fast, with new findings, models and techniques being published in a matter of months or weeks [7], [8], [9], [10], [11], AI researchers and practitioners often find it challenging to figure out the best recipes to build LLM-powered AI systems for their tasks. This paper gives a timely survey of the recent advances on LLMs. We hope this survey will prove a valuable and accessible resource for students, researchers and developers.

LLMs are large-scale, pre-trained, statistical language models based on neural networks. The recent success of LLMs is an accumulation of decades of research and development of language models, which can be categorized into four waves that have different starting points and velocity: statistical language models, neural language models, pre-trained language models and LLMs.

Statistical language models (SLMs) view text as a sequence of words, and estimate the probability of text as the product of their word probabilities. The dominating form of SLMs are Markov chain models known as the n-gram models, which compute the probability of a word conditioned on its immediate proceeding $n - 1$ words. Since word probabilities are estimated using word and n-gram counts collected from text corpora, the model needs to deal with data sparsity (i.e., assigning zero probabilities to unseen words or n-grams) by using *smoothing*, where some probability mass of the model is reserved for unseen n-grams [12]. N-gram models are widely used in many NLP systems. However, these models are incomplete in that they cannot fully capture the diversity and variability of natural language due to data sparsity.

Early neural language models (NLMs) [13], [14], [15], [16] deal with data sparsity by mapping words to low-dimensional continuous vectors (embedding vectors) and predict the next word based on the aggregation of the embedding vectors of its proceeding words using neural networks. The embedding vectors learned by NLMs define a hidden space where the semantic similarity between vectors can be readily computed as their distance. This opens the door to computing semantic similarity of any two inputs regardless their forms (e.g., queries vs. documents in Web search [17], [18], sentences in different languages in machine translation [19], [20]) or modalities (e.g., image and text in image captioning [21], [22]). Early NLMs are task-specific models, in that they are trained on task-specific data and their learned hidden space is task-specific.

Pre-trained language models (PLMs), unlike early NLMs, are task-agnostic. This generality also extends to the learned

hidden embedding space. The training and inference of PLMs follows the *pre-training and fine-tuning* paradigm, where language models with recurrent neural networks [23] or transformers [24], [25], [26] are pre-trained on Web-scale unlabeled text corpora for general tasks such as word prediction, and then finetuned to specific tasks using small amounts of (labeled) task-specific data. Recent surveys on PLMs include [8], [27], [28].

Large language models mainly refer to transformer-based neural language models [1] that contain tens to hundreds of billions of parameters, which are pre-trained on massive text data, such as PaLM [31], LLaMA [32], and GPT-4 [33], as summarized in Table III. Compared to PLMs, LLMs are not only much larger in model size, but also exhibit stronger language understanding and generation abilities, and more importantly, emergent abilities that are not present in smaller-scale language models. As illustrated in Fig. 1, these emergent abilities include (1) in-context learning, where LLMs learn a new task from a small set of examples presented in the prompt at inference time, (2) instruction following, where LLMs, after instruction tuning, can follow the instructions for new types of tasks without using explicit examples, and (3) multi-step reasoning, where LLMs can solve a complex task by breaking down that task into intermediate reasoning steps as demonstrated in the chain-of-thought prompt [34]. LLMs can also be augmented by using external knowledge and tools [35], [36] so that they can effectively interact with users and environment [37], and continually improve itself using feedback data collected through interactions (e.g. via reinforcement learning with human feedback (RLHF)).

Through advanced usage and augmentation techniques, LLMs can be deployed as so-called AI agents: artificial entities that sense their environment, make decisions, and take actions. Previous research has focused on developing agents for specific tasks and domains. The emergent abilities demonstrated by LLMs make it possible to build general-purpose AI agents based on LLMs. While LLMs are trained to produce responses in static settings, AI agents need to take actions to interact with dynamic environment. Therefore, LLM-based agents often need to augment LLMs to e.g., obtain updated information from external knowledge bases, verify whether a system action produces the expected result, and cope with when things do not go as expected, etc. We will discuss in detail LLM-based agents in Section IV.

In the rest of this paper, Section II presents an overview of state of the art of LLMs, focusing on three LLM families (GPT, LLaMA and PaLM) and other representative models. Section III discusses how LLMs are built. Section IV discusses how LLMs are used, and augmented for real-world applications Sections V and VI review popular datasets and benchmarks for evaluating LLMs, and summarize the reported LLM evaluation results. Finally, Section VII concludes the paper by summarizing the challenges and future research directions.

II. LARGE LANGUAGE MODELS

In this section we start with a review of early pre-trained neural language models as they are the base of LLMs, and

---

then focus our discussion on three families of LLMs: GPT, LLaMA, and PaLM. Table I provides an overview of some of these models and their characteristics.

A. Early Pre-trained Neural Language Models

Language modeling using neural networks was pioneered by [38], [39], [40]. Bengio et al. [13] developed one of the first neural language models (NLMs) that are comparable to n-gram models. Then, [14] successfully applied NLMs to machine translation. The release of RNNLM (an open source NLM toolkit) by Mikolov [41], [42] helped significantly popularize NLMs. Afterwards, NLMs based on recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) [19] and gated recurrent unit (GRU) [20], were widely used for many natural language applications including machine translation, text generation and text classification [43].

Then, the invention of the Transformer architecture [44] marks another milestone in the development of NLMs. By applying self-attention to compute in parallel for every word in a sentence or document an "attention score" to model the influence each word has on another, Transformers allow for much more parallelization than RNNs, which makes it possible to efficiently pre-train very big language models on large amounts of data on GPUs. These pre-trained language models (PLMs) can be fine-tuned for many downstream tasks.

We group early popular Transformer-based PLMs, based on their neural architectures, into three main categories: encoder-only, decoder-only, and encoder-decoder models. Comprehensive surveys of early PLMs are provided in [43], [28].

1) Encoder-only PLMs: As the name suggests, the encoder-only models only consist of an encoder network. These models are originally developed for language understanding tasks, such as text classification, where the models need to predict a class label for an input text. Representative encoder-only models include BERT and its variants, e.g., RoBERTa, ALBERT, DeBERTa, XLM, XLNet, UNILM, as to be described below.

BERT (Birectional Encoder Representations from Transformers) [24] is one of the most widely used encoder-only language models. BERT consists of three modules: (1) an embedding module that converts input text into a sequence of embedding vectors, (2) a stack of Transformer encoders that converts embedding vectors into contextual representation vectors, and (3) a fully connected layer that converts the representation vectors (at the final layer) to one-hot vectors. BERT is pre-trained uses two objectives: masked language modeling (MLM) and next sentence prediction. The pre-trained BERT model can be fine-tuned by adding a classifier layer for many language understanding tasks, ranging from text classification, question answering to language inference. A high-level overview of BERT framework is shown in Fig 3. As BERT significantly improved state of the art on a wide range of language understanding tasks when it was published, the AI community was inspired to develop many similar encoder-only language models based on BERT.

RoBERTa [25] significantly improves the robustness of BERT using a set of model design choices and training strategies, such as modifying a few key hyperparameters, removing the next-sentence pre-training objective and training with much
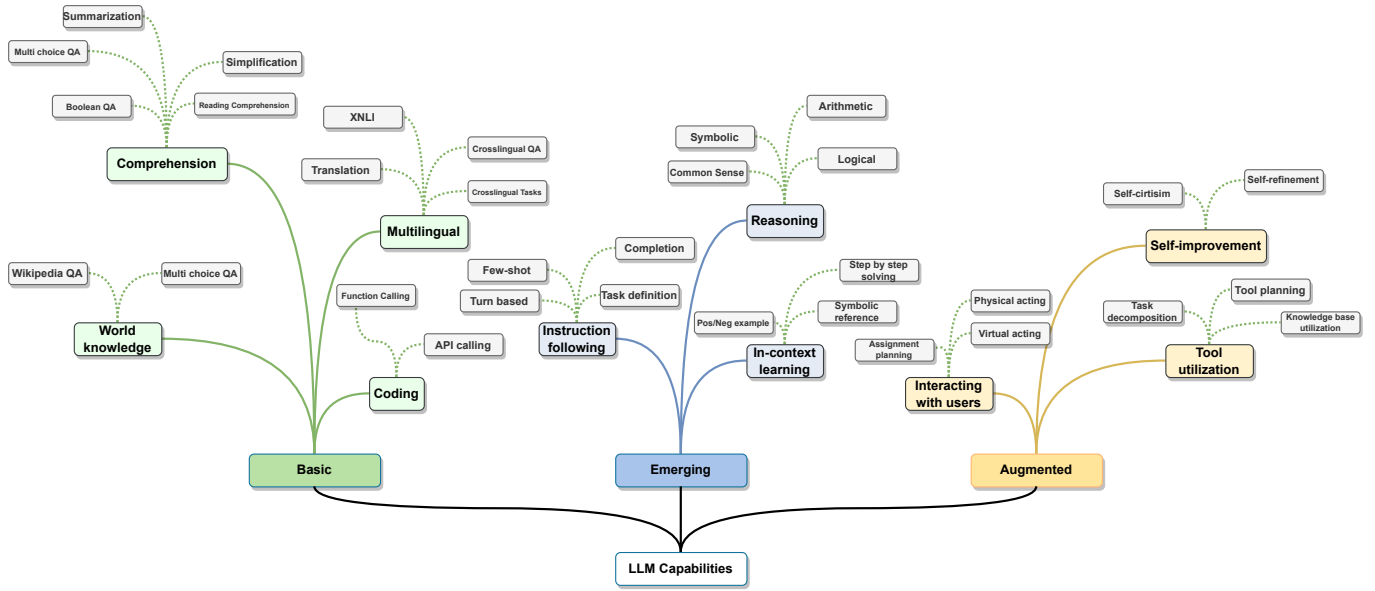
Fig. 1: LLM Capabilities.

larger mini-batches and learning rates. ALBERT [45] uses two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT: (1) splitting the embedding matrix into two smaller matrices, and (2) using repeating layers split among groups. DeBERTa (Decoding-enhanced BERT with disentangled attention) [26] improves the BERT and RoBERTa models using two novel techniques. The first is the disentangled attention mechanism, where each word is represented using two vectors that encode its content and position, respectively, and the attention weights among words are computed using disentangled matrices on their contents and relative positions, respectively. Second, an enhanced mask decoder is used to incorporate absolute positions in the decoding layer to predict the masked tokens in model pre-training. In addition, a novel virtual adversarial training method is used for fine-tuning to improve models' generalization. ELECTRA [46] uses a new pre-training task, known as replaced token detection (RTD), which is empirically proven to be more sample-efficient than MLM. Instead of masking the input, RTD corrupts it by replacing some tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens, a discriminative model is trained to predict whether a token in the corrupted input was replaced by a generated sample or not. RTD is more sample-efficient than MLM because the former is defined over all input tokens rather than just the small subset being masked out, as illustrated in Fig 4.

XLMs [47] extended BERT to cross-lingual language models using two methods: (1) a unsupervised method that only relies on monolingual data, and (2) a supervised method that leverages parallel data with a new cross-lingual language model objective, as illustrated in Fig 5. XLMs had obtained state-of-the-art results on cross-lingual classification, unsupervised and supervised machine translation, at the time they were proposed.

There are also encoder-only language models that leverage the advantages of auto-regressive (decoder) models for model training and inference. Two examples are XLNet and UNILM. XLNet [48] is based on Transformer-XL, pre-trained using a generalized autoregressive method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. UNILM (UNIfied pre-trained Language Model) [49] is pre-trained using three types of language modeling tasks: unidirectional, bidirectional, and sequence-to-sequence prediction. This is achieved by employing a shared Transformer network and utilizing specific self-attention masks to control what context the prediction is conditioned on, as illustrated in Fig 6. The pre-trained model can be fine-tuned for both natural language understanding and generation tasks.

2) Decoder-only PLMs: Two of the most widely used decoder-only PLMs are GPT-1 and GPT-2, developed by OpenAI. These models lay the foundation to more powerful LLMs subsequently, i.e., GPT-3 and GPT-4.

GPT-1 [50] demonstrates for the first time that good performance over a wide range of natural language tasks can be obtained by Generative Pre-Training (GPT) of a decoder-only Transformer model on a diverse corpus of unlabeled text in a self-supervised learning fashion (i.e., next word/token prediction), followed by discriminative fine-tuning on each specific downstream task (with much fewer samples), as illustrated in Fig 7. GPT-1 paves the way for subsequent GPT models, with each version improving upon the architecture and achieving better performance on various language tasks.

GPT-2 [51] shows that language models are able to learn to perform specific natural language tasks without any explicit supervision when trained on a large WebText dataset consisting of millions of webpages. The GPT-2 model follows the model designs of GPT-1 with a few modifications: Layer normalization is moved to the input of each sub-block, additional layer normalization is added after the final self-attention block,
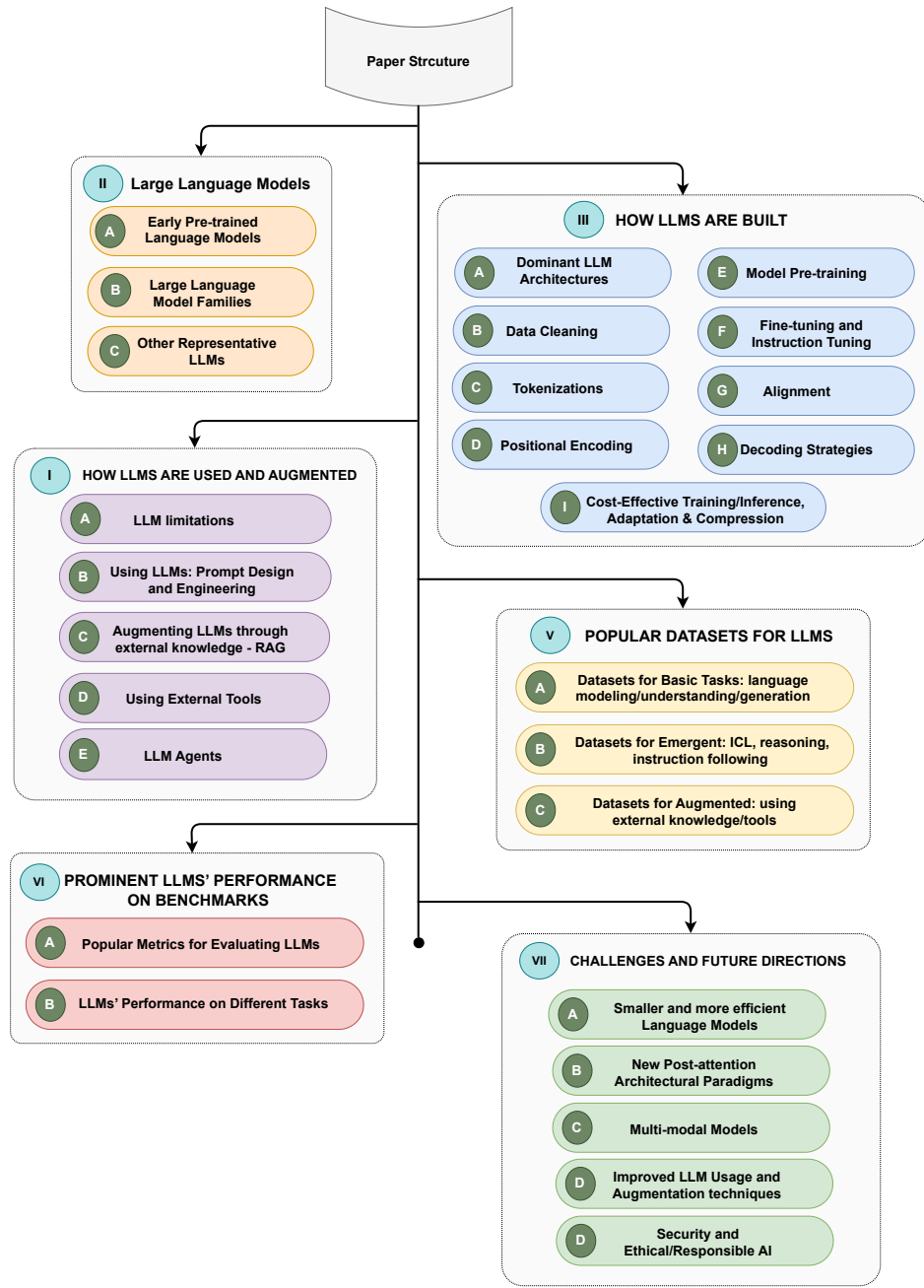
Fig. 2: The paper structure.

initialization is modified to account for the accumulation on the residual path and scaling the weights of residual layers, vocabulary size is expanded to 50,25, and context size is increased from 512 to 1024 tokens.

3) Encoder-Decoder PLMs: In [52], Raffle et al. shows that almost all NLP tasks can be cast as a sequence-to-sequence generation task. Thus, an encoder-decoder language model, by design, is a unified model in that it can perform all natural language understanding and generation tasks. Representative encoder-decoder PLMs we will review below are T5, mT5, MASS, and BART.

T5 [52] is a Text-to-Text Transfer Transformer (T5) model, where transfer learning is effectively exploited for NLP via an introduction of a unified framework in which all NLP tasks are cast as a text-to-text generation task. mT5 [53] is a multilingual variant of T5, which is pre-trained on a new Common Crawl-based dataset consisting of texts in 101 languages.

MASS (MAsked Sequence to Sequence pre-training) [54] adopts the encoder-decoder framework to reconstruct a sentence fragment given the remaining part of the sentence. The encoder takes a sentence with randomly masked fragment (several consecutive tokens) as input, and the decoder predicts the masked fragment. In this way, MASS jointly trains the

TABLE I: High-level Overview of Popular Language Models

Type	Model Name	#Parameters	Release	Base Models	Open Source	#Tokens	Training dataset
Encoder-Only	BERT	110M, 340M	2018	-	✓	137B	BooksCorpus, English Wikipedia
	RoBERTa	355M	2019	-	✓	2.2T	BooksCorpus, English Wikipedia, CC-NEWS, STORIES (a subset of Common Crawl), Reddit
	ALBERT	12M, 18M, 60M, 235M	2019	-	✓	137B	BooksCorpus, English Wikipedia
	DeBERTa	-	2020	-	✓	-	BooksCorpus, English Wikipedia, STORIES, Reddit content
	XLNet	110M, 340M	2019	-	✓	32.89B	BooksCorpus, English Wikipedia, Giga5, Common Crawl, ClueWeb 2012-B
Decoder-only	GPT-1	120M	2018	-	✓	1.3B	BooksCorpus
	GPT-2	1.5B	2019	-	✓	10B	Reddit outbound
Encoder-Decoder	T5 (Base)	223M	2019	-	✓	156B	Common Crawl
	MT5 (Base)	300M	2020	-	✓	-	New Common Crawl-based dataset in 101 languages (m Common Crawl)
	BART (Base)	139M	2019	-	✓	-	Corrupting text
GPT Family	GPT-3	125M, 350M, 760M, 1.3B, 2.7B, 6.7B, 13B, 175B	2020		×	300B	Common Crawl (filtered), WebText2, Books1, Books2, Wikipedia
	CODEX	12B	2021	GPT	✓	-	Public GitHub software repositories
	WebGPT	760M, 13B, 175B	2021	GPT-3	×	-	ELI5
	GPT-4	1.76T	2023	-	×	13T	-
LLaMA Family	LLaMA1	7B, 13B, 33B, 65B	2023	-	✓	1T, 1.4T	Online sources
	LLaMA2	7B, 13B, 34B, 70B	2023	-	✓	2T	Online sources
	Alpaca	7B	2023	LLaMA1	✓	-	GPT-3.5
	Vicuna-13B	13B	2023	LLaMA1	✓	-	GPT-3.5
	Koala	13B	2023	LLaMA	✓	-	Dialogue data
	Mistral-7B	7.3B	2023	-	✓	-	-
	Code Llama	34	2023	LLaMA2	✓	500B	Publicly available code
	LongLLaMA	3B, 7B	2023	OpenLLaMA	✓	1T	-
	LLaMA-Pro-8B	8.3B	2024	LLaMA2-7B	✓	80B	Code and math corpora
	TinyLlama-1.1B	1.1B	2024	LLaMA1.1B	✓	3T	SlimPajama, Starcoderdata
PaLM Family	PaLM	8B, 62B, 540B	2022	-	×	780B	Web documents, books, Wikipedia, conversations, GitHub code
	U-PaLM	8B, 62B, 540B	2022	-	×	1.3B	Web documents, books, Wikipedia, conversations, GitHub code
	PaLM-2	340B	2023	-	✓	3.6T	Web documents, books, code, mathematics, conversational data
	Med-PaLM	540B	2022	PaLM	×	780B	HealthSearchQA, MedicationQA, LiveQA
	Med-PaLM 2	-	2023	PaLM 2	×	-	MedQA, MedMCQA, HealthSearchQA, LiveQA, MedicationQA
Other Popular LLMs	FLAN	137B	2021	LaMDA-PT	✓	-	Web documents, code, dialog data, Wikipedia
	Gopher	280B	2021	-	×	300B	MassiveText
	ERNIE 4.0	10B	2023	-	×	4TB	Chinese text
	Retro	7.5B	2021	-	×	600B	MassiveText
	LaMDA	137B	2022	-	×	168B	public dialog data and web documents
	ChinChilla	70B	2022	-	×	1.4T	MassiveText
	Galactia-120B	120B	2022	-		450B	
	CodeGen	16.1B	2022	-	✓	-	THE PILE, BIGQUERY, BIGPYTHON
	BLOOM	176B	2022	-	✓	366B	ROOTS
	Zephyr	7.24B	2023	Mistral-7B	✓	800B	Synthetic data
	Grok-0	33B	2023	-	×	-	Online source
	ORCA-2	13B	2023	LLaMA2	-	2001B	-
	StartCoder	15.5B	2023	-	✓	35B	GitHub
	MPT	7B	2023	-	✓	1T	RedPajama, m Common Crawl, S2ORC, Common Crawl
	Mixtral-8x7B	46.7B	2023	-	✓	-	Instruction dataset
	Falcon 180B	180B	2023	-	✓	3.5T	RefinedWeb
	Gemini	1.8B, 3.25B	2023		✓	-	Web documents, books, and code, image data, audio data, video data
	DeepSeek-Coder	1.3B, 6.7B, 33B	2024	-	✓	2T	GitHub's Markdown and StackExchange
	DocLLM	1B,7B	2024	-	×	2T	IIT-CDIP Test Collection 1.0, DocBank

encoder and decoder for language embedding and generation, respectively.

BART [55] uses a standard sequence-to-sequence translation model architecture. It is pre-trained by corrupting text with an arbitrary noising function, and then learning to reconstruct the original text.

B. Large Language Model Families

Large language models (LLMs) mainly refer to transformer-based PLMs that contain tens to hundreds of billions of parameters. Compared to PLMs reviewed above, LLMs are not only much larger in model size, but also exhibit stronger language understanding and generation and emergent abilities that are not present in smaller-scale models. In what follows, we review three LLM families: GPT, LLaMA, and PaLM, as illustrated in Fig 8.

1) The GPT Family: Generative Pre-trained Transformers (GPT) are a family of decoder-only Transformer-based language models, developed by OpenAI. This family consists of GPT-1, GPT-2, GPT-3, InstrucGPT, ChatGPT, GPT-4,
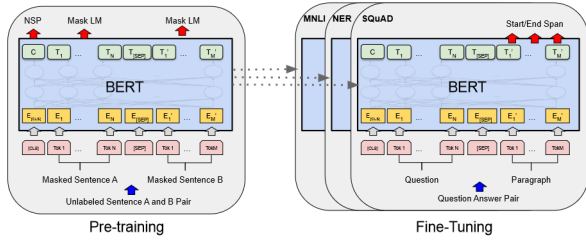
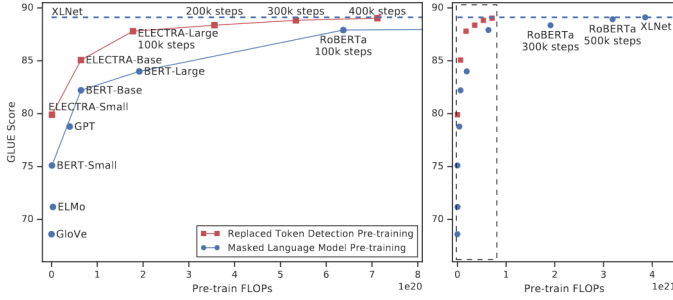Fig. 3: Overall pre-training and fine-tuning procedures for BERT. Courtesy of [24]

Fig. 4: A comparison between replaced token detection and masked language modeling. Courtesy of [46].
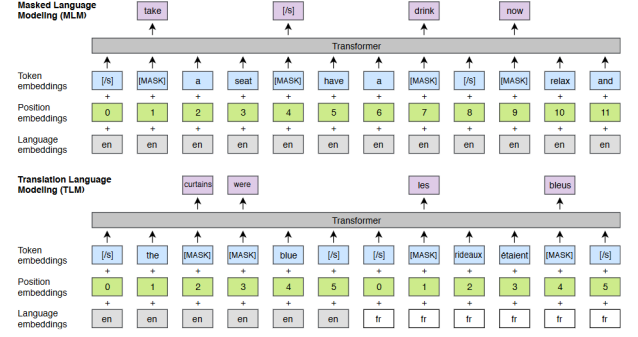

Fig. 5: Cross-lingual language model pretraining. The MLM objective is similar to BERT, but with continuous streams of text as opposed to sentence pairs. The TLM objective extends MLM to pairs of parallel sentences. To predict a masked English word, the model can attend to both the English sentence and its French translation, and is encouraged to align English and French representations. Courtesy of [47].
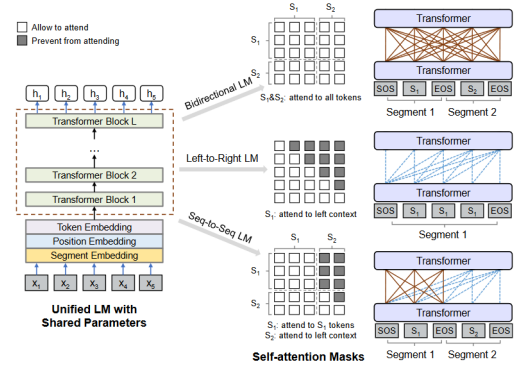

Fig. 6: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). Courtesy of [49].

CODEX, and WebGPT. Although early GPT models, such as GPT-1 and GPT-2, are open-source, recent models, such as GPT-3 and GPT-4, are close-source and can only be accessed via APIs. GPT-1 and GPT-2 models have been discussed in the early PLM subsection. We start with GPT-3 below.

GPT-3 [56] is a pre-trained autoregressive language model with 175 billion parameters. GPT-3 is widely considered as the first LLM in that not only it is much larger than previous PLMs, but also for the first time demonstrates emergent abilities that are not observed in previous smaller PLMs. GPT-3 shows the emergent ability of in-context learning, which means GPT-3 can be applied to any downstream tasks without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieved strong performance on many NLP tasks, including translation, question-answering, and the cloze tasks, as well as several ones that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, 3-digit arithmetic. Fig 9 plots the performance of GPT-3 as a function of the number of examples in in-context prompts.

CODEX [57], released by OpenAI in March 2023, is a general-purpose programming model that can parse natural language and generate code in response. CODEX is a descendant of GPT-3, fine-tuned for programming applications on code corpora collected from GitHub. CODEX powers Microsoft's GitHub Copilot.

WebGPT [58] is another descendant of GPT-3, fine-tuned to answer open-ended questions using a text-based web browser, facilitating users to search and navigate the web. Specifically,

WebGPT is trained in three steps. The first is for WebGPT to learn to mimic human browsing behaviors using human demonstration data. Then, a reward function is learned to predict human preferences. Finally, WebGPT is refined to optimize the reward function via reinforcement learning and rejection sampling.

To enable LLMs to follow expected human instructions, InstructGPT [59] is proposed to align language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, a dataset of labeler demonstrations of the desired model behavior is collected. Then GPT-3 is fine-tuned on this dataset. Then, a dataset of human-ranked model outputs is collected to further fine-tune the model using reinforcement learning. The method is known Reinforcement Learning from Human Feedback (RLHF), as shown in 10. The resultant InstructGPT models have shown improvements in truthfulness and reductions in toxic output generation while having minimal performance
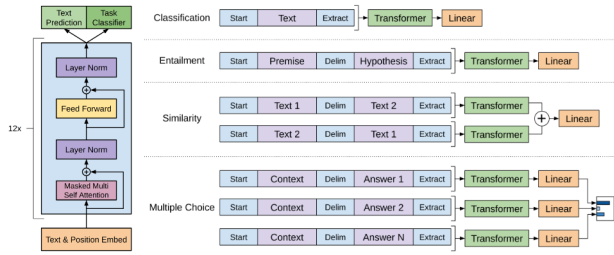
Fig. 7: High-level overview of GPT pretraining, and fine-tuning steps. Courtesy of OpenAI.

regressions on public NLP datasets.

The most important milestone of LLM development is the launch of ChatGPT (Chat Generative Pre-trained Transformer) [60] on November 30, 2022. ChatGPT is chatbot that enables users to steer a conversation to complete a wide range of tasks such as question answering, information seeking, text summarization, and more. ChatGPT is powered by GPT-3.5 (and later by GPT-4), a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

GPT-4 [33] is the latest and most powerful LLM in the GPT family. Launched in March, 2023, GPT-4 is a multi-modal LLM in that it can take image and text as inputs and produce text outputs. While still less capable than humans in some of the most challenging real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the top 10% of test takers, as shown in Fig 11. Like early GPT models, GPT-4 was first pre-trained to predict next tokens on large text corpora, and then fine-tuned with RLHF to align model behaviors with human-desired ones.

2) The LLaMA Family: LLaMA is a collection of foundation language models, released by Meta. Unlike GPT models, LLaMA models are open-source, i.e., model weights are released to the research community under a noncommercial license. Thus, the LLaMA family grows rapidly as these models are widely used by many research groups to develop better open-source LLMs to compete the closed-source ones or to develop task-specific LLMs for mission-critical applications.

The first set of LLaMA models [32] was released in February 2023, ranging from 7B to 65B parameters. These models are pre-trained on trillions of tokens, collected from publicly available datasets. LLaMA uses the transformer architecture of GPT-3, with a few minor architectural modifications, including (1) using a SwiGLU activation function instead of ReLU, (2) using rotary positional embeddings instead of absolute positional embedding, and (3) using root-mean-squared layer-normalization instead of standard layer-normalization. The open-source LLaMA-13B model outperforms the proprietary GPT-3 (175B) model on most benchmarks, making it a good baseline for LLM research.

In July 2023, Meta, in partnership with Microsoft, released the LLaMA-2 collection [61], which include both foundation language models and Chat models finetuned for dialog, known as LLaMA-2 Chat. The LLaMA-2 Chat models were reported to outperform other open-source models on many public benchmarks. Fig 12 shows the training process of LLaMA-2 Chat. The process begins with pre-training LLaMA-2 using publicly available online data. Then, an initial version of LLaMA-2 Chat is built via supervised fine-tuning. Subsequently, the model is iteratively refined using RLHF, rejection sampling and proximal policy optimization. In the RLHF stage, the accumulation of human feedback for revising the reward model is crucial to prevent the reward model from being changed too much, which could hurt the stability of LLaMA model training.

Alpaca [62] is fine-tuned from the LLaMA-7B model using 52K instruction-following demonstrations generated in the style of self-instruct using GPT-3.5 (text-davinci-003). Alpaca is very cost-effective for training, especially for academic research. On the self-instruct evaluation set, Alpaca performs similarly to GPT-3.5, despite that Alpaca is much smaller.

The Vicuna team has developed a 13B chat model, Vicuna-13B, by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. Preliminary evaluation using GPT-4 as a evaluator shows that Vicuna-13B achieves more than 90% quality of OpenAI's ChatGPT, and Google's Bard while outperforming other models like LLaMA and Stanford Alpaca in more than 90% of cases. 13 shows the relative response quality of Vicuna and a few other well-known models by GPT-4. Another advantage of Vicuna-13B is its relative limited computational demand for model training. The training cost of Vicuna-13B is merely $300.

Like Alpaca and Vicuna, the Guanaco models [63] are also finetuned LLaMA models using instruction-following data. But the finetuning is done very efficiently using QLoRA such that finetuning a 65B parameter model can be done on a single 48GB GPU. QLoRA back-propagates gradients through a frozen, 4-bit quantized pre-trained language model into Low Rank Adapters (LoRA). The best Guanaco model outperforms all previously released models on the Vicuna benchmark, reaching 99.3% of the performance level of ChatGPT while only requiring 24 hours of fine-tuning on a single GPU.

Koala [64] is yet another instruction-following language model built on LLaMA, but with a specific focus on interaction data that include user inputs and responses generated by highly capable closed-source chat models such as ChatGPT. The Koala-13B model performs competitively with state-of-the-art chat models according to human evaluation based on real-world user prompts.

Mistral-7B [65] is a 7B-parameter language model engineered for superior performance and efficiency. Mistral-7B outperforms the best open-source 13B model (LLaMA-2-13B) across all evaluated benchmarks, and the best open-source 34B model (LLaMA-34B) in reasoning, mathematics, and code generation. This model leverages grouped-query attention for faster inference, coupled with sliding window attention to effectively handle sequences of arbitrary length with a reduced inference cost.

The LLaMA family is growing rapidly, as more instruction-following models have been built on LLaMA or LLaMA-2, including Code LLaMA [66], Gorilla [67], Giraffe [68],
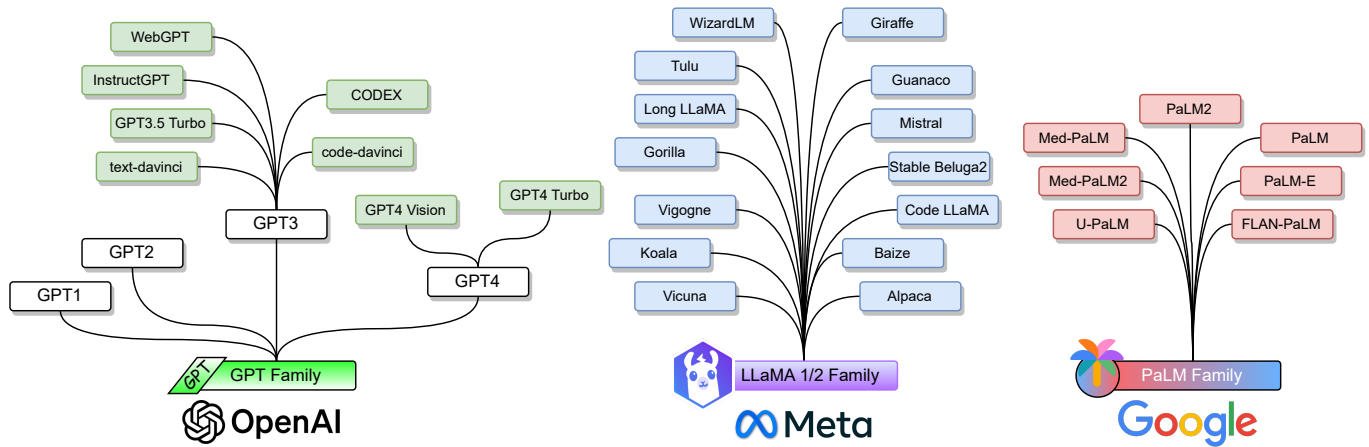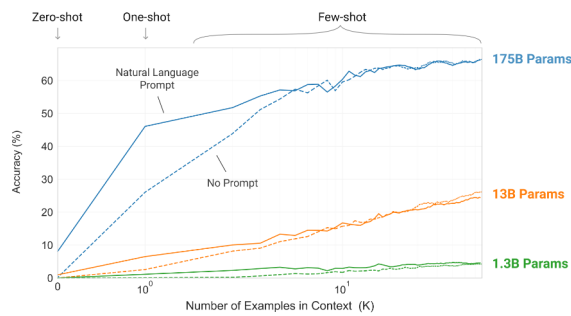
Fig. 8: Popular LLM Families.

Fig. 9: GPT-3 shows that larger models make increasingly efficient use of in-context information. It shows in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description. Courtesy of [56].
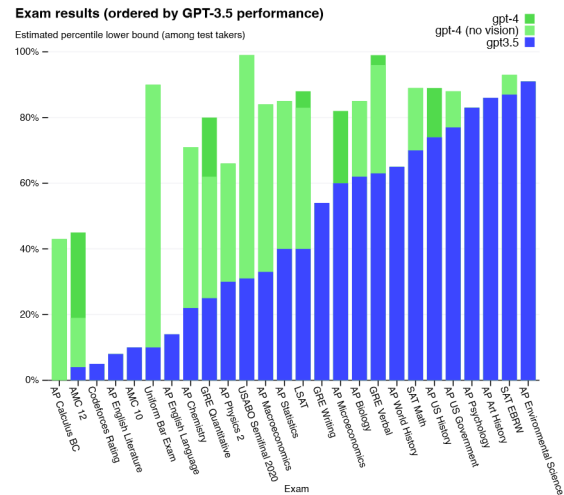

Fig. 10: The high-level overview of RLHF. Courtesy of [59].

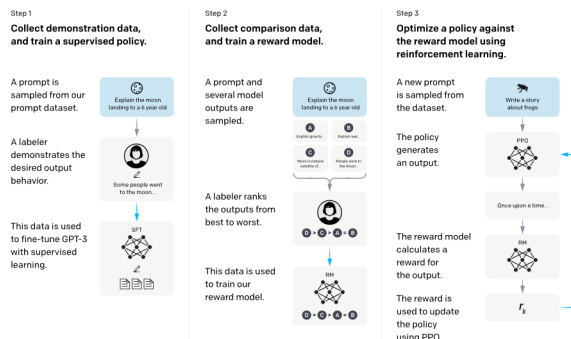Fig. 11: GPT-4 performance on academic and professional exams, compared with GPT 3.5. Courtesy of [33].

Vigogne [69], Tulu 65B [70], Long LLaMA [71], and Stable Beluga2 [72], just to name a few.

3) The PaLM Family: The PaLM (Pathways Language Model) family are developed by Google. The first PaLM model [31] was announced in April 2022 and remained private until March 2023. It is a 540B parameter transformer-based LLM. The model is pre-trained on a high-quality text corpus consisting of 780 billion tokens that comprise a wide range of natural language tasks and use cases. PaLM is pre-trained on 6144 TPU v4 chips using the Pathways system, which enables highly efficient training across multiple TPU Pods. PaLM demonstrates continued benefits of scaling by achieving state-of-the-art few-shot learning results on hundreds of language understanding and generation benchmarks. PaLM-540B outperforms not only state-of-the-art fine-tuned models on a suite of multi-step reasoning tasks, but also on par with humans on the recently released BIG-bench benchmark.

The U-PaLM models of 8B, 62B, and 540B scales are continually trained on PaLM with UL2R, a method of continue training LLMs on a few steps with UL2's mixture-of-denoiser objective [73]. An approximately 2x computational savings rate is reported.