

Large Language Models: A Survey

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu
Richard Socher, Xavier Amatriain, Jianfeng Gao

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 (LLMs), 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 preceding $n - 1$ 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 preceding 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 (LLMs) mainly refer to transformer-based neural language models¹ 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

¹Recently, several very promising non-transformer LLMs have been proposed, such as the LLMs based on structured state space models [29], [30]. See Section VII for more details.

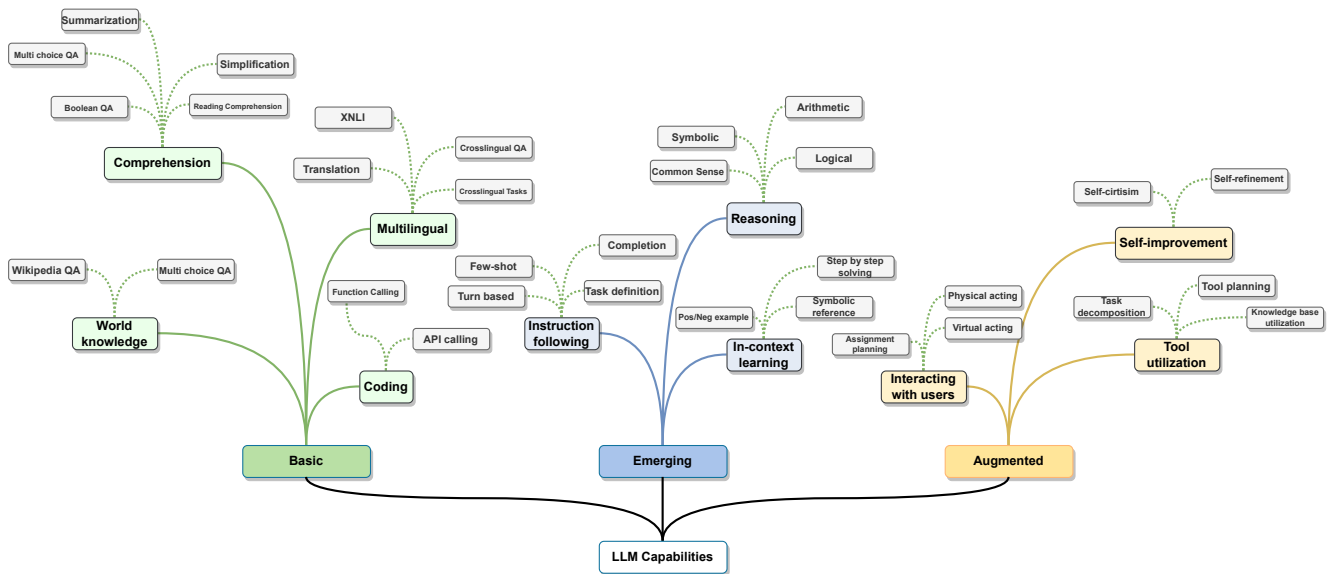


Fig. 1: LLM Capabilities.

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.