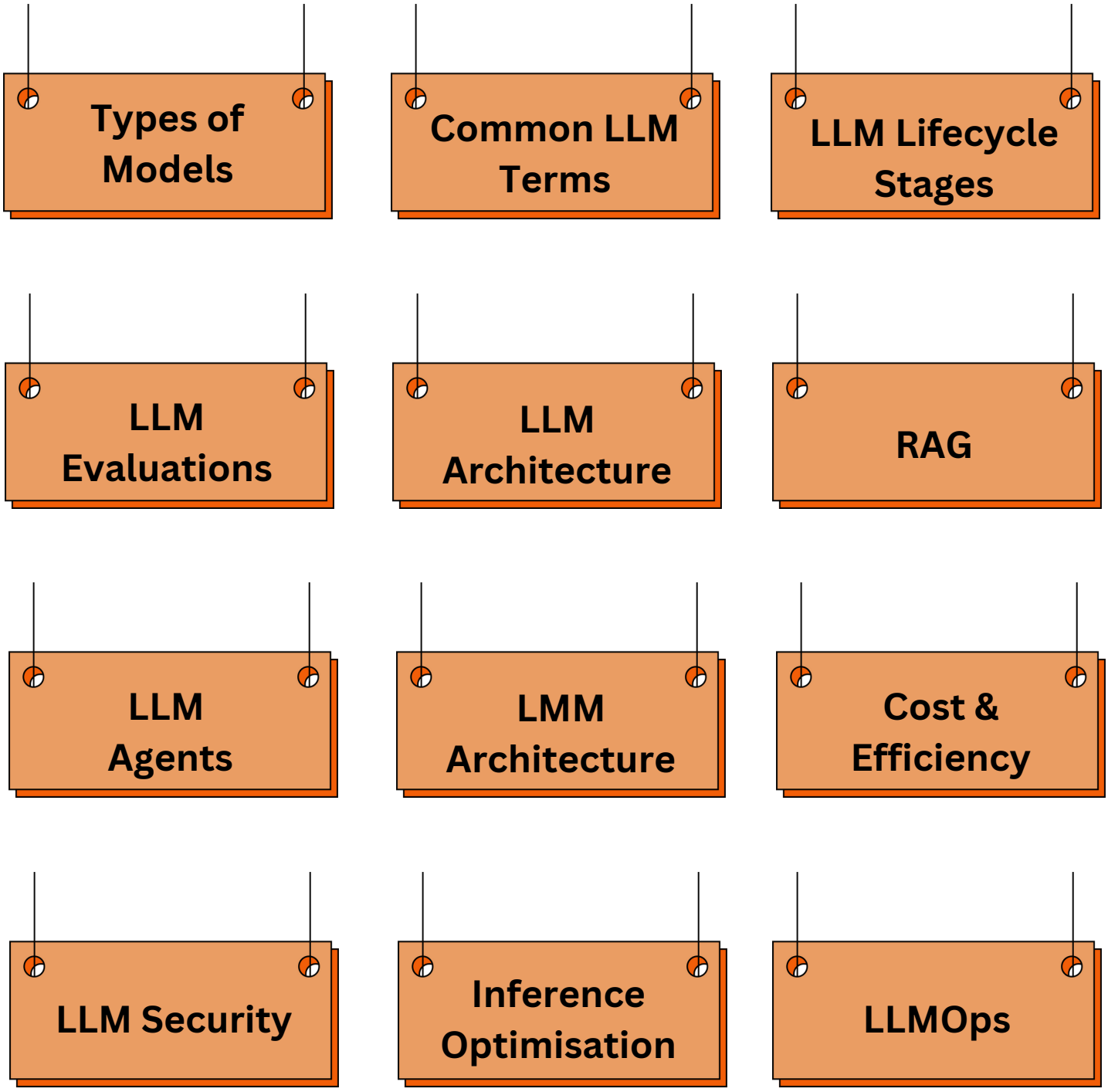
An work-in-progress

Generative AI Terminology

to get started



Abhinav Kimothi

Types of Models

Foundation Models

- Large AI models that have millions/billions of parameters and are trained on terabytes of generalized and unlabelled data.
- Designed to be general-purpose, providing a foundation for various AI applications.
- Examples: GPT 3.5, GPT4, Stable Diffusion, Llama, BERT, Gemini

Large Language Models (LLMs)

- Foundation models, trained on the "Transformer Architecture", that can perform a wide array of Natural Language Processing (NLP) tasks like text generation, classification, summarization etc.
- LLMs have been considered game-changers because of their ability to generate coherent text.
- All LLMs are next "token" prediction models. They only generate the next word, given an input of a sequence of words.
- The term "Large" refers to the number of trained parameters in the models (billions of parameters)
- Examples: GPT 3.5, Llama2, Mistral, Falcon

Small Language Models (SLMs)

- SLMs are like LLMs but with lesser number of trained parameters (therefore called "Small")
- They are faster, require less memory and compute, but are not as adaptable and extensible as an LLM. Therefore used for very specific tasks.
- Example : TinyLlama, Pythia

Large Multimodal Models (LMMs)

- MultiModal refers to the ability of the model to not just process and generate text, but also other data modalities like image, video, speech, audio, etc.
- Example : LLaVA, LLaMA-Adapter V2, LAVIN

Vision Language Models (VLMs)

- VLMs and LMMs have been used interchangeably. The core difference being that vision models focus on modalities of image and video while MultiModal models refer to all data modalities. Vision Models are therefore a subset of MultiModal Models
- Examples : GPT4, Gemini, Mixtral, Flamingo, BLIP, Macaw LLM

Generative Image Models

- Like LLMs produce a text output only, Generative Image Models produce and image output
- Text-to-Image functionality generates an image based on a text input (prompt) and the Image-to-Image functionality can be used to generate variations of an input image.
- Typical underlying architecture of these models is a Diffusion model
- Examples : Dall E3, Midjourney, Stable Diffusion

Text-to-Speech (TTS)

- As the name suggests, TTS models take a piece of text as input and synthesise a voice output.

Speech-to-Text (STT)

- STT models take a voice input and generates a text transcript as the output

Common LLM Terms

Prompt

- **Instruction or question you provide to the AI to get a specific kind of response**
- **Interacting with LLMs differs from traditional programming paradigms. Instead of formalized code syntax, you provide natural language inputs to the models. This input is called a Prompt**

Completion

- **The output that is generated by the LLM for a given prompt is called a completion.**

Inference

- **The process of the model generating a completion for a given prompt is called inference.**

Tokens

- **Token is a unit of text (words or characters) that an LLM processes. It is the building block of input and output that an LLM works with.**

Parameters

- **It is the number of setting variables in a language model that the model learns from the data while training. The language ability of the LLMs has been attributed to the large number of parameters that are trained.**

Context Window

- Every LLM, by the nature of the architecture, can process upto a maximum number of tokens (prompt and completion).
- This maximum number of tokens is referred to as the context window of the model

Temperature

- Temperature is a parameter that controls the randomness in the output of the LLM.
- High temperature makes the output more diverse and creative, low temperature makes it more focused and deterministic.

Top N/P Sampling

- LLMs are next token generation models. This is done by selecting a token based on the probability distribution.
- Top N sampling chooses a token from the top 'N' highest probability tokens
- Top P sampling chooses a token from the highest probability tokens whose probability sums up to 'P'

Hallucinations

- Incorrect or fabricated information generated by the model.
- It is important to remember that LLMs are choose a token from a probability distribution. LLMs don't claim to reproduce accurate information.

Bias and Toxicity

- LLMs are trained on huge volumes of unstructured data. This data comes from various sources (predominantly, the open internet). The model may show favouritism or generate harmful content based on this training data.

LLM Lifecycle Stages

Pre-training

- Training a language model on a large dataset to learn general language patterns.
- It takes a huge amount of data
- Training happens over a long period of time
- Objective is next token prediction
- High Cost, High Expertise

Prompt Engineering

- In simple words, Prompt Engineering is the process of crafting effective prompts to get desired model behaviour.

Supervised Fine Tuning

- Fine Tuning is a supervised learning process, where you take a labelled dataset of prompt-completion pairs to adjust the weights of an LLM.
- Instruction Fine Tuning is a strategy where the LLM is trained on examples of Instructions and how the LLM should respond to those instructions. Instruction Fine Tuning leads to improved performance on the instruction task.
- Full Fine Tuning is where all the LLM parameters are updated. It requires enough memory to store and process all the gradients and other components.

Catastrophic Forgetting

- Fine Tuning on a single task can significantly improve the performance of the model on that task.
- However, because the model weights get updated, the instruct model's performance on other tasks (which the base model performed well on) can get reduced. This is called Catastrophic Forgetting.

Reinforcement Learning from Human Feedback

- Reinforcement Learning is a type of machine learning in which an agent learns to make decisions related to a specific goal by taking actions in an environment, with the objective of maximising some notion of a cumulative reward
- In RLHF, the agent (our fine-tuned instruct LLM) in its environment (Context Window) takes one action (of generating text) from all available actions in the action space (the entire vocabulary of tokens/words in the LLM).
- The outcome of this action (the generated text) is evaluated by a human and is given a reward if the outcome (the generated text) aligns with the goal. If the outcome does not align with the goal, it is given a negative reward or no reward. This is an iterative process and each step is called a rollout. The model weights are adjusted in a manner that the total rewards at the end of the process are maximised.
- One of the primary objective of RLHF is to align with the human values of Helpfulness, Honesty and Harmlessness (HHH)

Reward Model

- In RLHF, Instead of a human giving a feedback continually, a classification model called the Reward Model is trained on human generated training examples

Reinforcement Learning from AI Feedback

- Scaling human feedback for RLHF can be challenging due to the significant human effort required to produce the trained reward model. As the number of models and use cases increases, human effort becomes a limited resource, necessitating methods to scale human feedback.
- First proposed in 2022 by researchers at Anthropic, Constitutional AI is an approach to scale supervision and address some unintended consequences of RLHF. Constitutional AI involves training models using a set of rules and principles that govern the model's behaviour, forming a "constitution".
- The training process for Constitutional AI involves two phases: supervised learning and reinforcement learning.
- In the supervised learning phase, the model is prompted with harmful scenarios and asked to critique its own responses based on constitutional principles. The revised responses, conforming to the rules, are used to fine-tune the model.
- The reinforcement learning phase, known as reinforcement learning from AI feedback (RLAIF), uses the fine-tuned model to generate responses based on constitutional principles

In Context Learning

- In context learning is when a large language model follows the instructions to generate responses without any other modelling intervention.

Few Shot Learning

- Teaching the LLM to do something specific with just a handful of examples provided within the prompt

LLM Evaluation

Perplexity

- **A measure of how well a language model predicts a sample. It gauges how surprised the model would be by new data.**

BLEU

- **A metric for evaluating the quality of machine-generated text. Measures how well the generated text matches reference text**

ROUGE

- **A set of metrics for evaluating automatic summarization and machine translation.. Assesses the overlap between the generated text and reference text**

BIG-bench (Google)

- **The Beyond the Imitation Game benchmark (BIG-bench) is a specific test designed to challenge and assess the capabilities of big models**

ARC

- **Abstraction and Reasoning Corpus (ARC) benchmark for assessing reasoning abilities in language models. Focuses on how well the model can understand and reason through different scenarios.**

HellaSwag

- **HellaSwag is a challenge dataset for evaluating common sense NLI that is specially hard for state-of-the-art models, though its questions are trivial for humans (>95% accuracy)**

MMLU

- **Massive Multitask Language Understanding focuses on zero-shot and few-shot evaluations, mirroring human evaluation methods.**

TruthfulQA

- **A benchmark for assessing the truthfulness of language models' responses. Tests the model's accuracy and reliability in providing truthful information.**

GLUE

- **General Language Understanding Evaluation (GLUE) benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems.**

SuperGLUE

- **SuperGLUE is a benchmark for evaluating the performance of language models, specifically designed to assess comprehensive language comprehension beyond traditional NLP tasks.**
- **It includes more challenging tasks, diverse task formats, and comprehensive human baselines, making it a sophisticated and widely accepted benchmark for assessing the capabilities of language models.**

HELM

- **HELM (Holistic Evaluation of Language Models) enhances transparency in understanding language models (LMs) by taxonomizing use cases and metrics. HELM aims to be a living benchmark continuously updated with new scenarios, metrics, and models.**

LLM Architecture

Tokenization

- **Breaking down text into smaller units (tokens) for processing.**

Recurrent Neural Network (RNN)

- **A type of neural network designed for sequential data processing. Allows the model to maintain memory of past inputs, crucial for understanding context in language.**

Transformer

- **A type of neural network architecture designed for parallelization in processing sequential data. A more efficient way for models to understand and generate sequences of data.**

Encoder

- **Part of a model responsible for processing input data. It "encodes" or transforms input information into a format the model can work with.**

Decoder

- **Part of a model responsible for generating output based on the encoded input. It "decodes" the information, turning the model's understanding into a meaningful response.**

Attention

- **Mechanism in neural networks that allows the model to focus on specific parts of input data.**
  - **Self-Attention**
    - **An attention mechanism where the model pays attention to different parts of its own input. It helps the model weigh the importance of different words within the same input.**
  - **Multi-Headed Self-Attention**
    - **Using multiple self-attention mechanisms in parallel. Enhances the model's ability to capture various aspects of input information simultaneously.**
  - **Attention Map**
    - **A visual representation of where the model focuses its attention. A map showing which parts of the input are more crucial for generating the output.**

Encoder Only Models

- **Models that only have the encoding part and don't generate output directly. Used when the focus is on understanding and representing input information rather than generating responses.**

Causal Language Modeling

- **Causal language modeling involves predicting the next word or sequence of words in a sentence based on the preceding context, emphasizing the temporal order and causal relationships within the language.**

Decoder Only Models

- **Models that only generate output based on pre-encoded information. Useful when the focus is on generating responses without the need for understanding new input**

Masked Language Modeling

- **Masked language modeling is a technique in natural language processing where certain tokens in a sequence are intentionally masked, and the model is trained to predict those masked tokens based on the surrounding context.**

Sequence-to-Sequence Models

- **Models designed to transform input sequences into output sequences. Ideal for tasks like language translation, summarization, and conversation.**

Embeddings

- **Embeddings in natural language processing refer to the mapping of words, phrases, or sentences into numerical vectors in a continuous vector space. These embeddings are learned through representation learning techniques, capturing semantic relationships and contextual information.**

Retrieval Augmented Generation

Retrieval Augmented Generation (RAG)

- RAG improves the efficacy of LLM applications by retrieving data and documents relevant to a question or task and providing them as context for the LLM

Vector Databases

- Databases that store data in vector form, enabling efficient similarity searches.

Retrievers

- Components in a RAG system that retrieve relevant information from a vector database.

Naive RAG

- Basic implementation without additional complexities or optimizations
- Retrieve → Read

Advanced RAG

- A more sophisticated approach to Retrieval Augmented Generation.
- Rewrite → Retrieve → Rerank → Read

Modular RAG

- A Retrieval Augmented Generation system with separate and interchangeable components
- Modules like memory, search, routing, fusion etc.

Chunking

- **The splitting of longer text into manageable sizes for faster search and managing context windows**

RAG Evaluations

- **Context Relevance**
  - **Is the Retrieved Context relevant to the Query?**
- **Answer Relevance**
  - **Is the Response relevant to the Query?**
- **Context Recall**
  - **Is the Retrieved Context aligned to the Ground Truth?**
- **Context Precision**
  - **is the Retrieved Context ordered correctly?**
- **Groundedness/Faithfulness**
  - **How well the generated output is grounded in the retrieved information.**
- **Accuracy**
  - **The correctness of the generated output.**

LLM Agents

Agent

- **The entity or system that utilizes a language model to perform tasks.**

Memory

- **The ability of an agent or model to retain information from previous interactions.**

Planning

- **The organization of actions in LLM Agents to achieve specific goals.**

Tools

- **Additional components or resources used by an agent to enhance performance.**

ReAct

- **Reason and Act, determines how the model attributes information to generate responses.**

Chain-of-thought

- **The logical sequence of ideas or reasoning followed by an agent.**

Tree-of-thought

- **A hierarchical structure of interconnected ideas or information.**

Task-Question Decomposition

- **Breaking down a task or question into smaller, manageable components.**

Reflection

- **The ability of an LLM agent to think about and analyze its own thought processes.**

LMM Architecture

Generative Adversarial Network (GAN)

- A type of neural network framework involving a generator and a discriminator.
- The generator creates content, and the discriminator evaluates its quality, fostering improvement.

Variational Auto-encoder (VAE)

- A type of neural network designed for generating new data points. Focuses on encoding and decoding data in a way that allows for meaningful generation.

Modalities

- Different forms or types of data, such as text, images, or audio.

Multimodal Embedding Space

- A shared space where representations of different modalities are aligned.

Contrastive Language-Image Pretraining (CLIP)

- CLIP (Contrastive Language-Image Pre-Training), by OpenAI, is a neural network trained on a variety of (image, text) pairs

Contrastive Learning

- A training method where the model learns by contrasting positive and negative examples.

Vision Encoder

- The part of the model responsible for processing visual information. Handles the encoding of visual data, making it understandable for the overall model.

Cost & Efficiency

Graphics Processing Unit (GPU)

- A GPU is a specialized hardware component used to accelerate the training and inference processes of neural networks.

Parameter Efficient Fine Tuning (PEFT)

- Full fine tuning, like pre-training, requires memory not just to store the model, but also other parameters like optimisers, gradients etc.
- Parameter Efficient Fine Tuning or PEFT fine tunes only a subset of model parameters and, in some cases, do not touch the original weights at all.
- Because PEFT only retrains a subset of parameters, Catastrophic Forgetting can be avoided.

Quantization

- Quantization is a technique that reduces the precision of numerical representations in a model, leading to more efficient computation.

Low Rank Adaptation (LoRA)

- LoRA involves adapting the model's parameters to a lower rank, reducing computational complexity while maintaining performance.

Soft Prompting

- Soft prompting is a technique that involves providing gentle guidance to the model during the generation process, allowing flexibility in responses.

Fully Sharded Data Parallel (FSDP)

- **FSDP is a parallelization strategy that involves fully sharding the model's parameters across multiple devices, improving efficiency in distributed training.**

Distributed Data Parallel (DDP)

- **DDP is a parallelization strategy where the model's parameters are distributed across multiple devices for parallel training.**

LLM Security

Prompt Injection

- **Prompt injections is manipulation of an LLM with maliciously designed prompt prompts, causing it to ignore filters or execute unwanted commands.**
- **The attacker may overwrite the system prompt, gaining access to functions and data accessible by LLM.**

Data Leakage

- **LLM may accidentally reveal sensitive data like PII through responses.**

Training Data Poisoning

- **During the pre-training or fine-tuning stage, incorrect, biased or dangerous data may be introduced in the training set.**

Deployment & Inference Optimisation

Latency

- **Latency is the time delay between sending a request to a model and receiving the corresponding response, crucial for real-time applications.**

Throughput

- **Throughput refers to the number of requests a model can handle in a given time period, indicating its processing speed.**

Pruning

- **Pruning involves removing unnecessary or less important connections in a neural network to reduce its size and improve efficiency.**

Distillation

- **Distillation is the process of training a smaller model to mimic the behavior of a larger, more complex model, transferring knowledge and improving efficiency**

Flash Attention

- **Flash attention is an optimized attention mechanism designed for efficient computation, particularly in deployment scenarios with limited resources.**

KV Cache

- **KV Cache is a mechanism that stores precomputed key-value pairs, facilitating fast retrieval and reducing computation during the inference process.**

Positional Encoding

- **Positional encoding is a technique used in sequence models to inject information about the position of tokens in the input sequence, aiding the model in understanding the order of the data.**

Speculative Decoding

- **Speculative decoding involves predicting multiple possible outcomes for a given input, allowing the model to consider different potential responses.**

LLMOps (Providers)

Model Access and Training and FineTuning

- OpenAI
- HuggingFace
- Google Vertex AI
- Anthropic
- AWS Bedrock
- AWS Sagemaker Jumpstart

Data Loading

- Snorkel AI
- LlamaIndex
- LangChain

Vector DB and Indexing

- Pinecone
- Chroma
- FAISS
- Weviate
- Milvus

Application Framework

- LangChain
- LlamaIndex
- Haystack

PromptEngineering

- W&B
- PromptLayer
- TruLens
- TruEra

Evaluation

- TruLens
- TruEra
- Ragas

Deployment Frameworks

- Vllm
- TensorRT-LLM

Deployment and Inferencing

- AWS
- GCP
- OpenAI
- Azure

Monitoring

- HoneyHive
- TruEra

Proprietary LLM/VLMs

- GPT (3/3.5/4) by OpenAI
- Gemini by Google
- Claude2 by Anthropic
- Jurassic by A121
- Cohere

Open Source LLMs

- Llama2 by Meta
- Phi2 by Microsoft
- Mixtral by Mistral
- Falcon by TII
- Vicuna by LMSYS

Hello!
I'm Abhinav...
This is a work-in-progress collection of terms. Please subscribe for updates on these notes.

Get Access to Live updates

If you have any feedback, corrections, modifications, additions to the list, please share with me

LinkedIn Github Medium Insta email X Linktree Gumroad

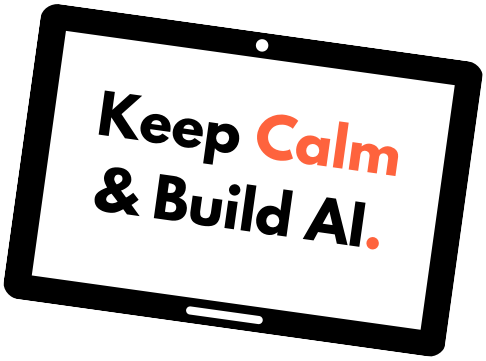
Also, check these out

Detailed Notes from **Generative AI with Large Language Models** Course by **Deeplearning.ai** and **AWS**.

DOWNLOAD EBOOK

A Simple Introduction to Retrieval Augmented Generation with LangChain & LlamaIndex using OpenAI/HF

DOWNLOAD EBOOK



Keep Calm & Build AI.

Abhinav Kimothi