The Best LLMs Cheatsheet

Part 1

LLMs Introduction

- Definition and Overview
- Importance and Applications

Basic Concepts

- Neural Networks & Deep Learning
- Training Data & Datasets
- Parameters & Model Size

How LLMs Work

- Tokenization
- Attention Mechanism & Transformers
- Language Modeling Objectives

Popular LLM Architectures

- GPT
- BERT
- PaLM
- LLaMA

Training LLMs from Scratch

- Data Collection & Preparation
- Training Techniques
- Fine-tuning & Transfer Learning

Evaluation and Metrics

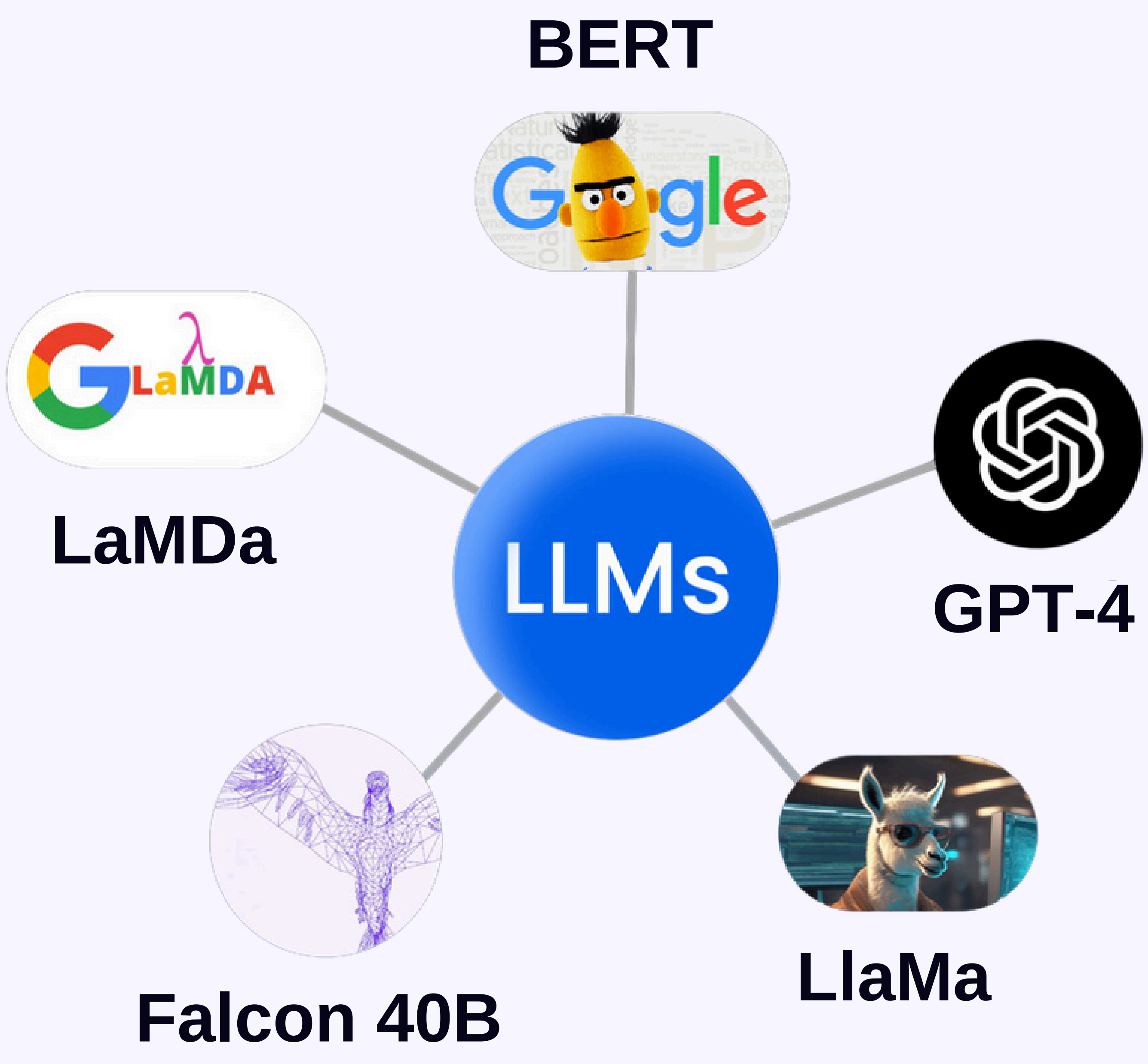
- Perplexity
- BLEU
- ROUGE
- GLUE

▨ These parts are covered in this cheatsheet

LLMs Introduction

- Large Language Models (LLMs) are advanced deep learning models designed to understand, generate, and manipulate human language. They are typically based on neural network architectures, such as transformers, which allow them to process large amounts of text data and learn patterns in human language.

- These models are "large" due to their vast number of parameters—often ranging from millions to hundreds of billions—enabling them to capture complex language structures and nuances.



Importance and Applications

- LLMs have revolutionized the field of natural language processing (NLP) by significantly improving the accuracy and fluency of language-related tasks. Their importance stems from their ability to:

- **Understand and Generate Text**: LLMs can perform various tasks, including text completion, summarization, translation, and question answering.

- **Adapt to Different Domains**: By pre-training on vast datasets and fine-tuning on specific tasks, LLMs can adapt to a wide range of domains, from medical research to customer service.

- **Facilitate Human-Machine Interaction**: They enable more natural and intuitive interactions between humans and machines, powering chatbots, virtual assistants, and other conversational agents.

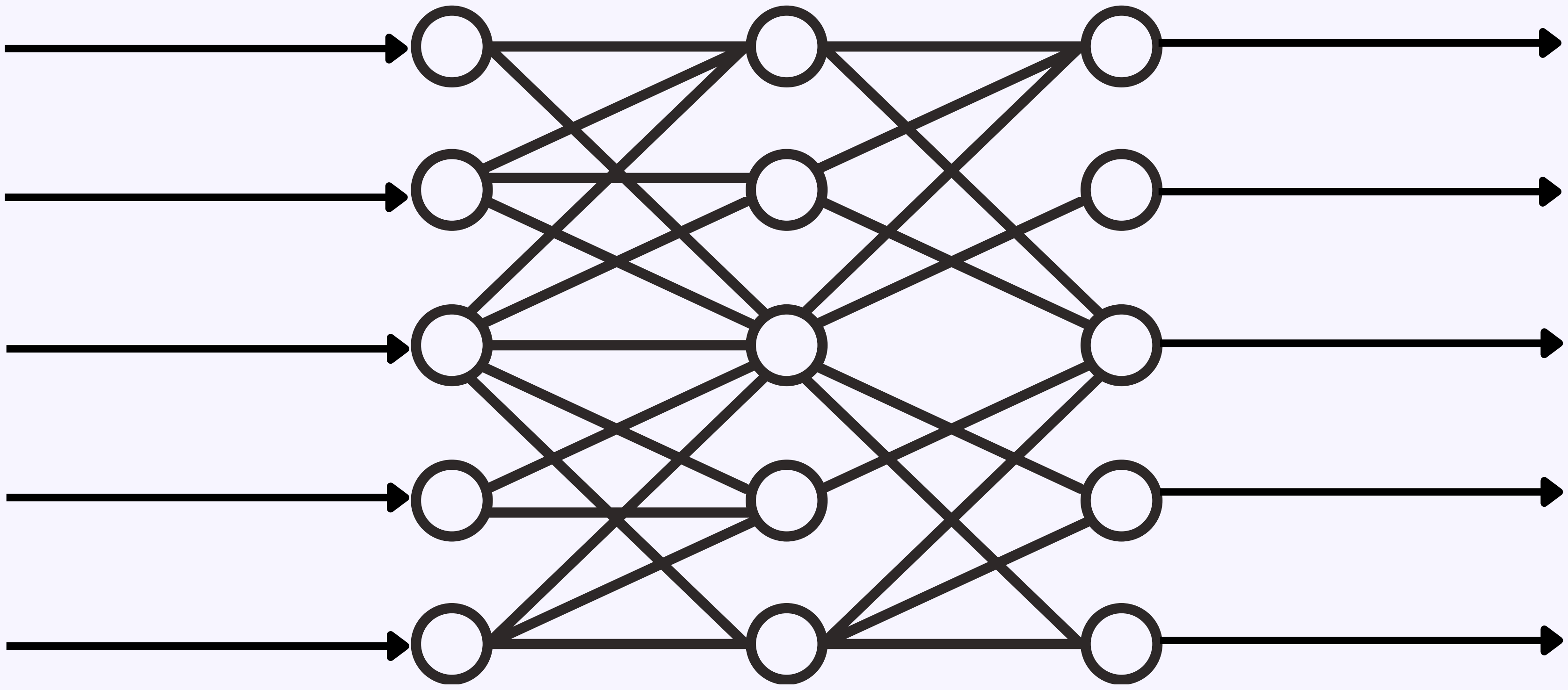
- **Application incude**: Content Creation, Customer Support, Data Analysis, Language Translation

Basic Concepts and Terminology

Neural Networks and Deep Learning

- Neural Networks are computational models inspired by the human brain's structure, consisting of layers of interconnected nodes (neurons) that process and learn from data. Deep Learning is a subset of machine learning that uses neural networks with many layers (deep architectures) to model complex patterns in large datasets.

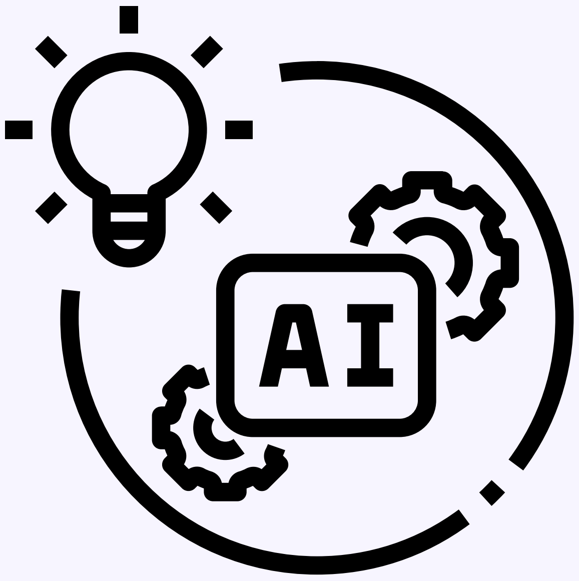
- LLMs, like GPT and BERT, are built on deep neural networks, specifically transformer architectures, which excel at processing sequences, such as text.



Training Data and Datasets

Training Data is the foundational fuel for LLMs. It consists of massive collections of text from diverse sources, such as books, articles, websites, and scientific papers. These datasets help the model learn the patterns, syntax, semantics, and context of human language. Popular datasets for training LLMs include:

- **Common Crawl**: A dataset derived from web scraping, providing a large, diverse text corpus.
- **Wikipedia and BooksCorpus**: High-quality texts from Wikipedia and books, often used to improve understanding of structured language.
- **Specialized Datasets**: Domain-specific datasets, such as medical or legal texts, used to fine-tune LLMs for specific tasks.



Parameters and Model Size

Parameters are the components of the model that are learned from the training data. They include weights and biases that help the model make predictions. The Model Size is typically measured by the number of parameters it contains. For example:

- Small Models: Contain millions to a few hundred million parameters (e.g., GPT-2).
- Medium to Large Models: Contain hundreds of millions to a few billion parameters (e.g., BERT, GPT-3).
- Very Large Models: Contain tens to hundreds of billions of parameters (e.g., GPT-4, PaLM).

A larger number of parameters generally allows the model to capture more complex patterns in data, but it also requires more computational resources for training and inference.

Pre-training and Fine-tuning

- **Pre-training** is the initial phase where an LLM is trained on a massive dataset in an unsupervised or self-supervised manner to learn general language patterns. During this phase, the model learns to predict the next word in a sentence, fill in missing words, or generate coherent text, thereby understanding grammar, facts, and context.

- **Fine-tuning** involves training the pre-trained model on a smaller, task-specific dataset to adapt it to particular applications. For example, an LLM pre-trained on general text can be fine-tuned on a dataset of legal documents to specialize in legal language and context. Fine-tuning helps the model become more accurate and effective in specific tasks, such as sentiment analysis, translation, or question answering.

How LLMs Work

Tokenization

- **Tokenization** is the process of breaking down text into smaller units, called tokens, which could be words, subwords, or even individual characters. For example, the sentence "ChatGPT is awesome!" might be tokenized into ["ChatGPT", "is", "awesome", "!"]. Tokenization is a crucial first step in processing language because it converts human-readable text into a format that the model can understand and manipulate.

- **Word-level Tokenization**: Splits text into individual words but may struggle with out-of-vocabulary words (words not seen during training).

- **Subword-level Tokenization** (Byte-Pair Encoding, WordPiece): Breaks down words into meaningful subwords, allowing the model to handle rare or unseen words more effectively.

- **Character-level Tokenization**: Breaks text into individual characters, useful for capturing the fine-grained structure of languages but often less efficient for longer texts.

Attention Mechanism and Transformers

- The Attention Mechanism allows the model to focus on specific parts of the input sequence when generating output, effectively determining which words or tokens are more relevant to the current task. It assigns different weights to different words in a sentence, allowing the model to capture context and dependencies more accurately.

- Transformers are neural network architectures built on the attention mechanism. Unlike earlier models that processed text sequentially, transformers can process all words in a sequence simultaneously (in parallel), making them highly efficient and capable of capturing long-range dependencies in text. The key components of transformers are:

  - Self-Attention
  - Multi-Head Attention
  - Feedforward Neural Networks
  - Positional Encoding

Language Modeling Objectives

LLMs are trained using different language modeling objectives depending on their architecture:

- **Masked Language Model (MLM)**: Used by models like BERT. In MLM, some tokens in the input are randomly masked, and the model is trained to predict these masked tokens. This helps the model understand bidirectional context—considering both the left and right context around a word.

- **Causal Language Model (CLM)**: Used by models like GPT. CLM predicts the next word in a sequence given all the previous words. It is unidirectional, meaning it only considers the left context (past tokens) to predict the next token, making it useful for generative tasks where the output is generated sequentially, like text completion.

Inference and Sampling Methods

Inference is the process of generating text or performing a task using a trained LLM. During inference, the model takes an input prompt and produces a relevant output based on the patterns and information it has learned during training.

- Sampling Methods are techniques used to decide which word or token to generate next during inference. Common methods include:

➤ **Greedy Search**: Selects the token with the highest probability at each step. It is fast but can produce repetitive or less diverse outputs.

➤ **Beam Search**: Evaluates multiple possible outputs at each step and keeps the most likely sequences. It balances between exploration (trying different options) and exploitation (choosing the best path), often producing more coherent results.

➤ **Top-k Sampling**: Chooses from the top-k most probable tokens at each step, allowing for more diverse outputs by adding randomness.

➤ **Nucleus Sampling (Top-p Sampling)**: Chooses from the smallest set of tokens whose cumulative probability exceeds a certain threshold (p), balancing between diversity and coherence.

Popular LLM Architectures

GPT (Generative Pre-trained Transformer)

- GPT (Generative Pre-trained Transformer) is an LLM developed by OpenAI, known for its generative capabilities. GPT models are based on a Causal Language Model (CLM), where the goal is to predict the next word in a sequence, given all the previous words. The architecture is unidirectional, focusing on the left-to-right context, making GPT particularly powerful for tasks that involve generating coherent and contextually relevant text, such as text completion, storytelling, and creative writing.

- Key characteristics of GPT:

➤ Transformer Decoder Architecture: Uses only the decoder part of the transformer, optimized for generating text.

➤ Pre-training and Fine-tuning: Pre-trained on a large corpus of diverse text and fine-tuned on specific tasks to adapt to different applications.

➤ Generative Focus: Excellent at tasks where generating new text is essential (e.g., dialogue generation, content creation).

➤ Versions: Includes GPT, GPT-2, GPT-3, and GPT-4, with each new version incorporating larger model sizes and improved performance.

BERT

Bidirectional Encoder
Representations from Transformers

- BERT (Bidirectional Encoder Representations from Transformers) is a model developed by Google that introduced a new way of understanding language context by considering both the left and right sides of a word simultaneously, hence "bidirectional." BERT is designed for Masked Language Modeling (MLM), where random words in a sentence are masked, and the model is trained to predict these words based on their context. This makes BERT highly effective for tasks that require deep understanding and contextual awareness, such as sentiment analysis, named entity recognition, and question answering.

- Key characteristics of BERT:

➤ Transformer Encoder Architecture: Utilizes the encoder part of the transformer to focus on understanding the meaning of the input text.

➤ Bidirectional Context: Reads text from both directions to capture full context, enhancing comprehension for NLP tasks.

➤ Pre-training on Large Datasets: Pre-trained on vast amounts of text from sources like Wikipedia, BooksCorpus, etc.

➤ Fine-tuning for Specific Tasks: Can be fine-tuned for specific NLP tasks using small, task-specific datasets.

➤ Variants: Includes BERT, RoBERTa, DistilBERT, and others that build on or modify the original BERT architecture for different needs.

T5 Text-to-Text Transfer Transformer

- T5 (Text-to-Text Transfer Transformer), developed by Google, is an innovative approach to NLP tasks by framing every problem as a text-to-text task. This means both the input and output are always text strings, regardless of the task (e.g., translation, summarization, or sentiment analysis). T5 is trained on a diverse set of NLP tasks, which allows it to generalize well across different applications.

- Key characteristics of T5:

➤ **Unified Text-to-Text Framework**: Handles all NLP tasks in a consistent format by treating every input and output as text.

➤ **Transformer Architecture**: Uses both the encoder and decoder parts of the transformer, providing flexibility in understanding and generating text.

➤ **Multi-task Learning**: Trained on a large collection of different NLP tasks simultaneously, improving generalization and adaptability.

➤ **Pre-training and Fine-tuning**: Like other LLMs, T5 is pre-trained on massive datasets and can be fine-tuned for specific applications.

➤ **Scalability**: Available in various sizes (small, base, large, XL, XXL) to balance between computational efficiency and performance.

LLaMA, PaLM, and Others

- **LLaMA (Large Language Model Meta AI)**: A family of foundational language models by Meta (formerly Facebook), optimized for efficiency and performance. LLaMA is designed to democratize access to LLM capabilities by providing smaller, more efficient models that are still competitive in performance with much larger counterparts. It aims to provide robust language understanding and generation with reduced computational requirements.

- **PaLM (Pathways Language Model)**: Developed by Google, PaLM is a large, dense language model that leverages the Pathways system, designed to enable a single model to be trained on many tasks, with better efficiency and resource utilization. PaLM is highly scalable, designed to learn from vast amounts of data across multiple modalities, and is geared toward advanced NLP tasks like reasoning, commonsense understanding, and creative text generation.

Other Notable Models

- **XLNet**: Combines ideas from both BERT and autoregressive models to capture bidirectional context while addressing limitations of BERT’s masked language modeling.

- **Megatron-Turing NLG**: A large-scale, generative model jointly developed by NVIDIA and Microsoft, designed for natural language generation tasks.

- **GLaM (Gated Language Model)**: Developed by Google, GLaM uses a mixture-of-experts approach, allowing it to dynamically activate different parts of the model for different tasks, making it more efficient in resource use.

Moreover,
we are offering a

Free Certification

on LLMs, check the link
in the description

@Harshit Ahluwalia