Large Language Model Fundamentals
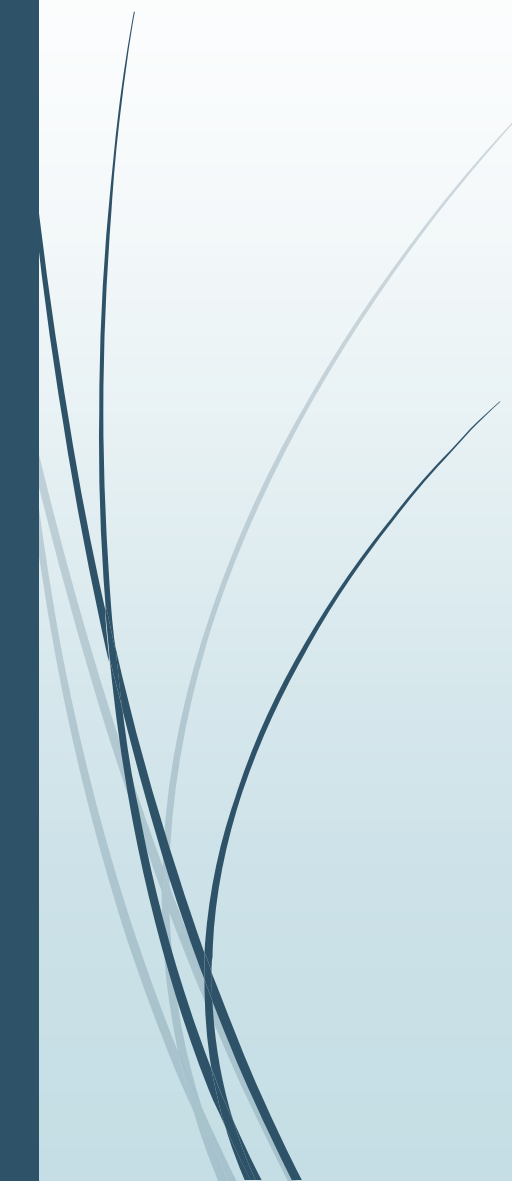
Jun Liang

2025/01/12

Motivation

- Make the audience understand the essence of artificial intelligence including LLMs.

- Share helpful information/tips on how to learn and use LLMs.
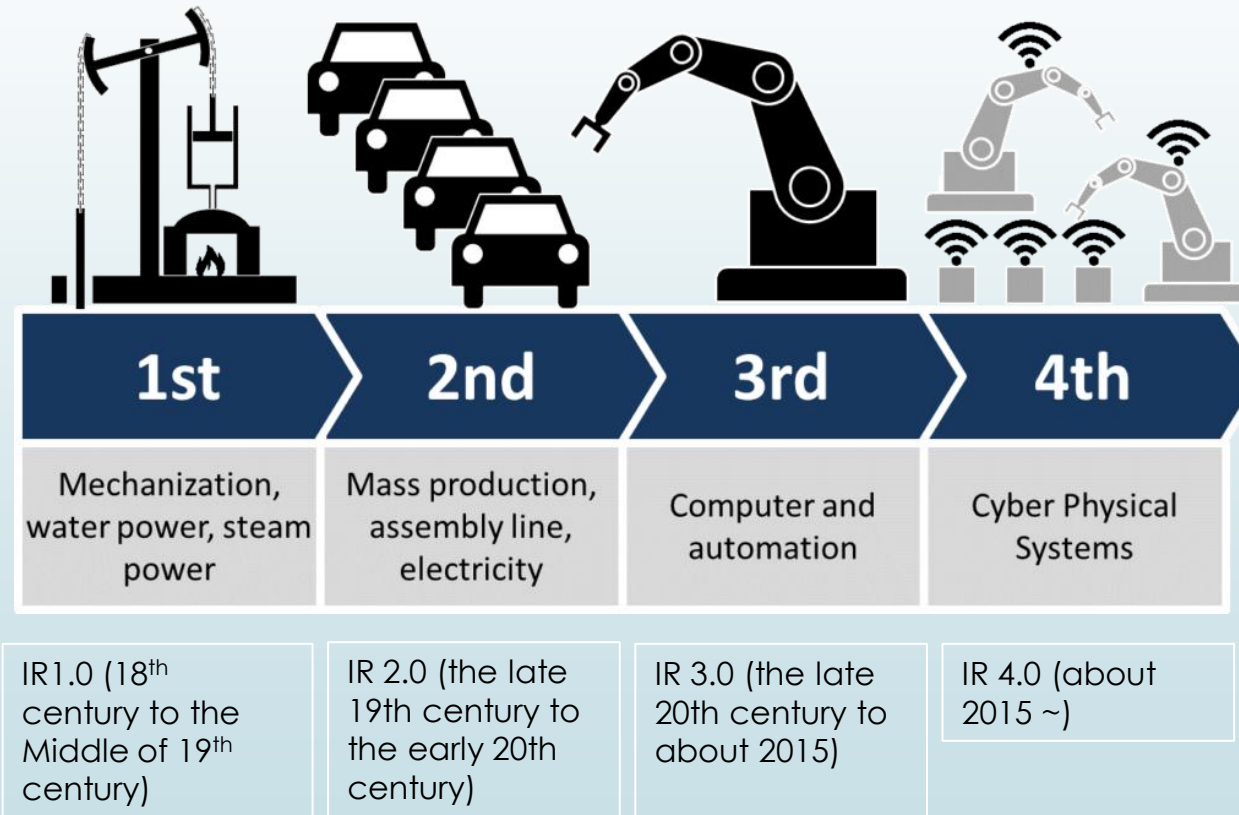
Agenda

- Artificial Intelligence (AI)
  - IR 4.0
  - What is AI?
- Large Language Model (LLM)
  - What is LLM?
  - LLM Key Features
    - Scalability
      - Scaling Laws
      - Prompt Engineering
    - Usability
      - Retrieval Augmented Generation (RAG)
      - AI Agent
  - LLM Training
    - Fine-tuning, Parameter Efficient Fine Tuning (PEFT)
 - LLM Evaluation
  - Tips

Industry Revolution (IR)

- Four phases in IR

| 1st	2nd	3rd	4th
Mechanization, water power, steam power	Mass production, assembly line, electricity	Computer and automation	Cyber Physical Systems
IR1.0 (18th century to the Middle of 19th century)	IR 2.0 (the late 19th century to the early 20th century)	IR 3.0 (the late 20th century to about 2015)	IR 4.0 (about 2015 ~)

What is AI?

- A field that includes **machine learning** and **neural networks**.

  - Machine Learning (ML)

    - A subset of AI that includes neural networks. Has strong background on programming, algorithm development, statistical methodologies, and big data analytics.

    - Compared to deep learning, some people call it **shallow machine learning**.

  - Neural Network (NN)/Artificial Neural Network (ANN)

    - A technique to process data in a way that **mimics the human brain**. Neural networks can **learn from experience** and **identify patterns** in data.

      - **Multi-layer perceptron (MLP)**

        - **Fully connected/Dense layer feedforward networks**: Each node connects to all nodes in the next layer.

        - **At least three layers of nodes**: An input layer, one or more hidden layers, and an output layer.

        - **Non-linear Activation function**: to learn more complex patterns that are not possible with a purely linear model.

  - Deep learning

    - More than tens or even hundreds layers

    - More complicated structures to enhance the problems of neural networks
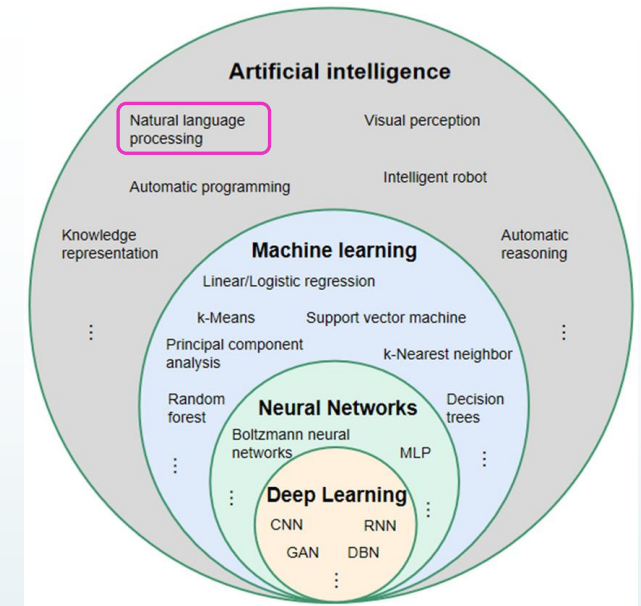
Relationship between artificial intelligence, machine learning, neural network, and deep learning.

(MLP: multi-layer perception/preceptron; CNN: convolutional neural network; RNN: recurrent neural network; DBN: deep belief network; GAN: generative adversative network)
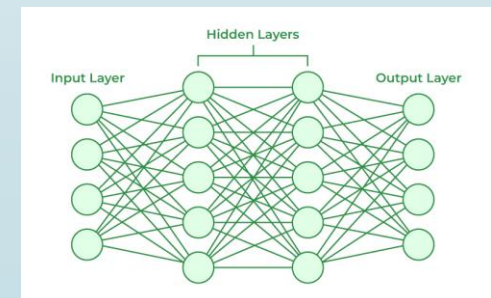
An example of MLP neural networks architecture

Wrap up

- Concept
    - Artificial Intelligence
        - Creating intelligent machines to mimic human brains/intelligence.
    - Machine Learning
        - Finding algorithms to enable computers to learn from data. Especially, finding the rules of data.
    - Neural Network
        - Computational models with interconnected artificial neurons.
   - Deep Learning
        - Neural networks with more and deeper layers of hierarchical representation of learning.
- Relation
    - **Deep learning** is a **subfield of machine learning** which uses **artificial neural networks** to enable **artificial intelligence** systems to learn complex patterns/regularities from data.

Machine Learning and Neural Network

- The three main types of machine learning.
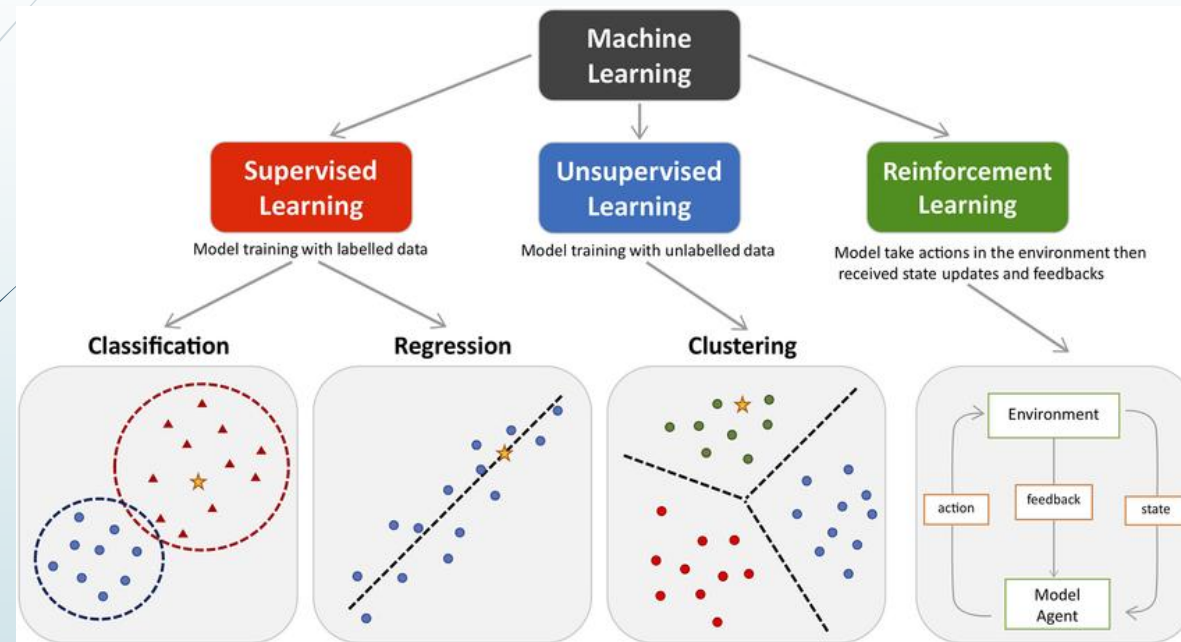
Classification involves predicting discrete labels (e.g., spam or not spam)

Regression involves through the moving trend of the previous data to predict the next data's moving trend. The moving trends are continuous values (e.g., house prices).
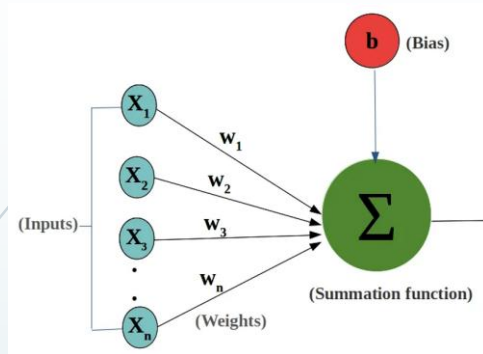
Identifies groups of similar data points (e.g., Social network analysis).

The agent learns by taking actions in an environment to maximize cumulative reward over time (e.g., AlphaGo).
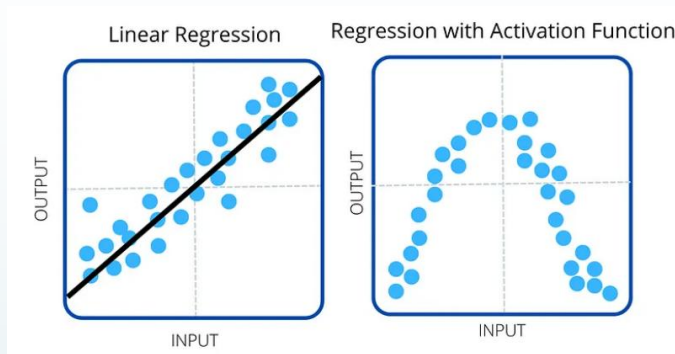
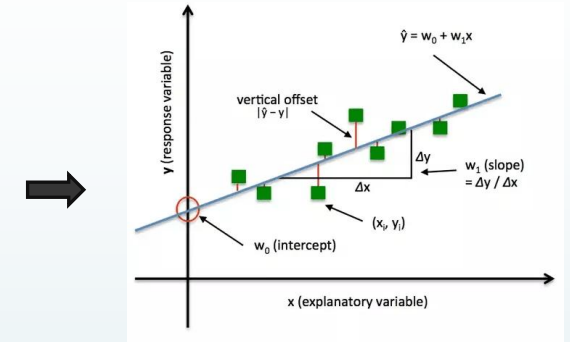**The three main types of machine learning**

The learning process of Regression in neural network
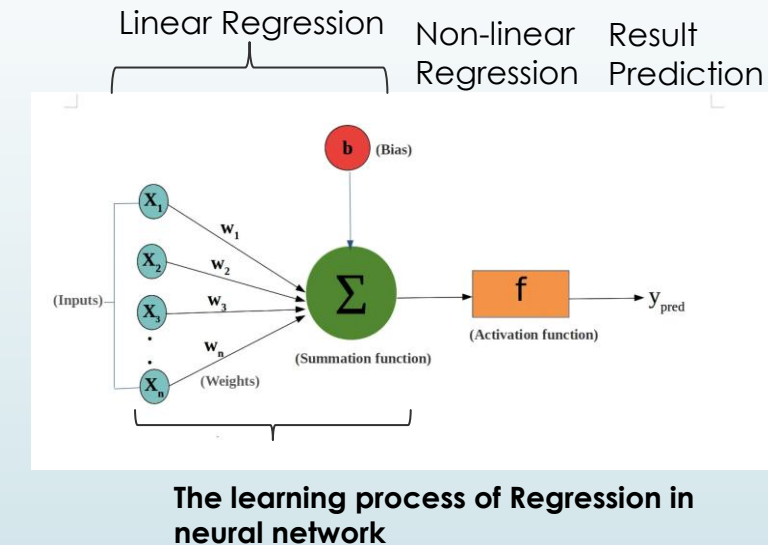
Weights, Bias, Activation function, Loss function in neural network

Weights, Bias

Activation function

Loss function (using back propagation)

- Example: You want to buy a house,

  - x_1 stands for the Area of the house; y_1 stands for the function to decide the price to **buy** a house

    - $y = w_1 x_1 + b_1$; $(y, w_1, x_1, b_1 \text{ are real numbers; })$

  - But···If you want to buy a school district house, then you need to rethink to decide the house. x_2 stands for the distance to the school, the function changes:

    - $y = w_1 x_1 + w_2 x_2 + b_1 + b_2$; $(y, w_1, x_1, b_1, w_2, x_2, b_2 \text{ are real numbers})$

  - But···If you want to rent to someone else after your child's graduation, then there is one more function to decide the **rental** price y_2 :

$$y_1 = \boldsymbol{w_{11}} x_1 + \boldsymbol{w_{21}} x_2 + b_1$$
$$y_2 = \boldsymbol{w_{12}} x_1 + \boldsymbol{w_{22}} x_2 + b_2$$

2-dim $\Rightarrow$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} w_{11}, & w_{11} \\ w_{12}, & w_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b$$

2-dimension Matrix

Expand to N-dim $\Rightarrow$ $Y = WX + b$ (W, X, Y are matrices)

N-dimension Matrix

Wrap up

- The current neural network is mainly regression type.

- Weight is the slope of $\frac{y_t - y_{t-1}}{x_t - x_{t-1}}$

- The learning process is forward process.

- The update process of weight and bias (both of them are matrices) is backward.

  - The process uses backpropagation to get the gradient descent (slope) of each learning step.

  - By calculating the loss function between prediction result and ground truth (GT) to make loss function as smaller as it can be.
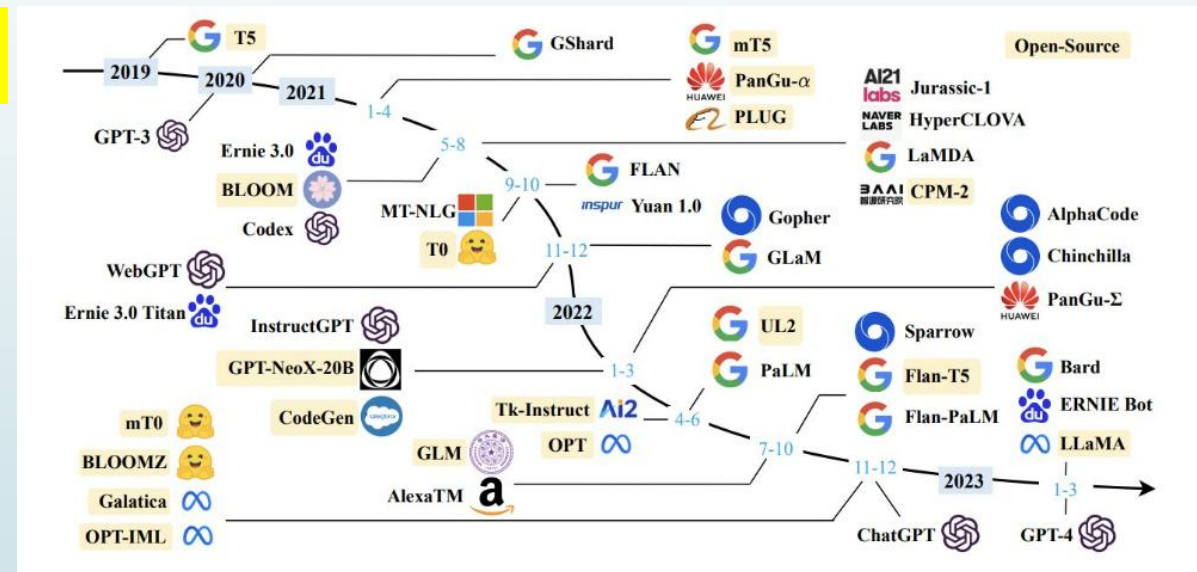
But if there is no GT?

What is LLM?

➡ A large language model (LLM) is a type of **machine learning** model designed for **natural language processing** tasks such as **language generation**. LLMs are language models with many parameters (weights, inputs, bias, outputs), and are trained with **self-supervised learning** on a vast amount of text.

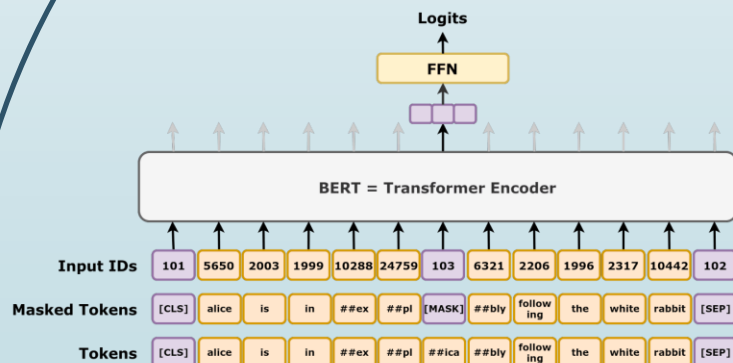BERT (2018) 340M)

Mamba, GPT-4o...

A timeline of existing large language models (having a size larger than 10B) until 2023/04

Self-supervised Learning

➡ Self-supervised learning is a form of **unsupervised learning where the system generates supervisory signals (supervisor)** from the data itself. This is typically achieved through the **creation of pretext tasks**, which involve manipulating the input data and then training the model to predict the original data from these modifications. Common pretext tasks include:

➡ Masked Language Modeling (MLM)

➡ Next Sentence Prediction (NSP)

In masked language modeling, 15% of tokens would be randomly selected for masked-prediction task, and the training objective was to predict the masked token given its context. In more detail, the selected token is
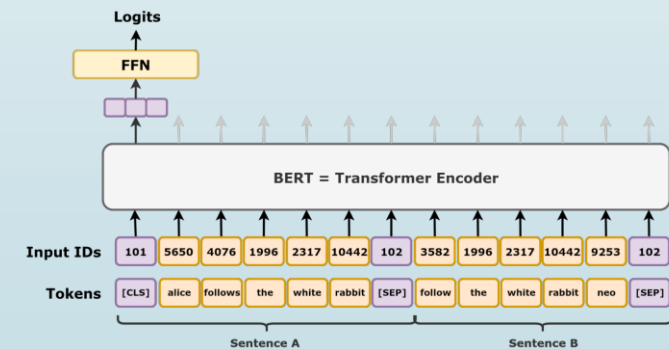
· replaced with a [MASK] token with probability 80%,
· replaced with a random word token with probability 10%,
· not replaced with probability 10%.

For example, given "[CLS] my dog is cute [SEP] he likes playing" the model should output token [IsNext].

Given "[CLS] my dog is cute [SEP] how do magnets work" the model should output token [NotNext].
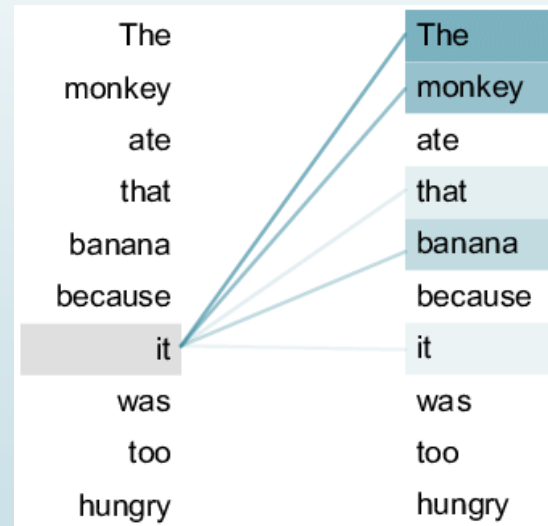

The masked language modeling task.

The next sentence prediction task.

Source:
https://www.goml.io/the-role-of-self-supervised-learning-in-llm-development/
https://en.wikipedia.org/wiki/BERT_(language_model)

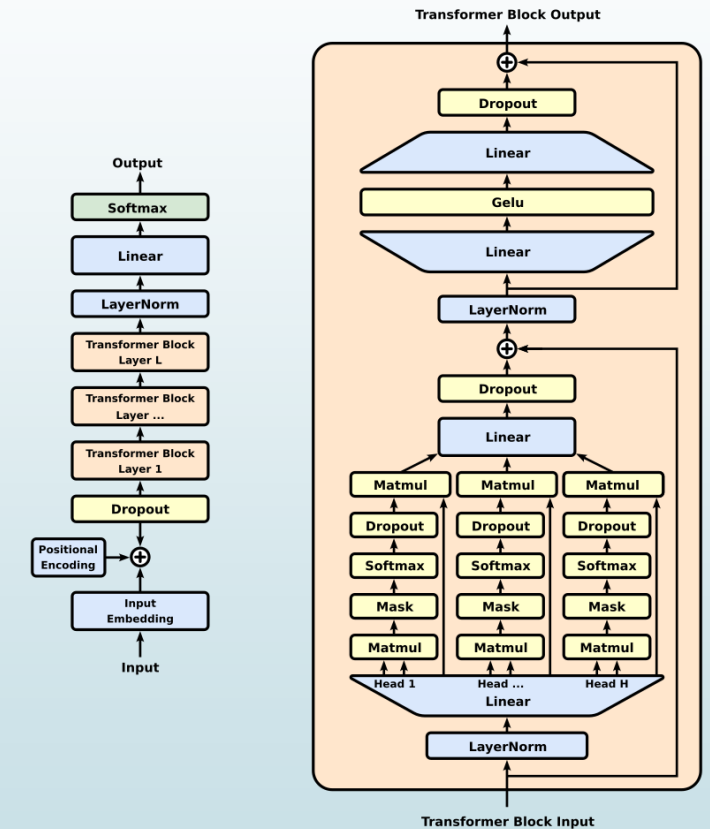# Transformer (Self-Attention), Generative pre-trained transformer (GPT)

The GPT-1 architecture was a twelve-layer **decoder-only transformer**, using **twelve masked self-attention heads**, with 64-dimensional states each (for a total of 768). Rather than simple stochastic gradient descent, the Adam optimization algorithm was used; the learning rate was increased linearly from zero over the first 2,000 updates to a maximum of 2.5×10−4
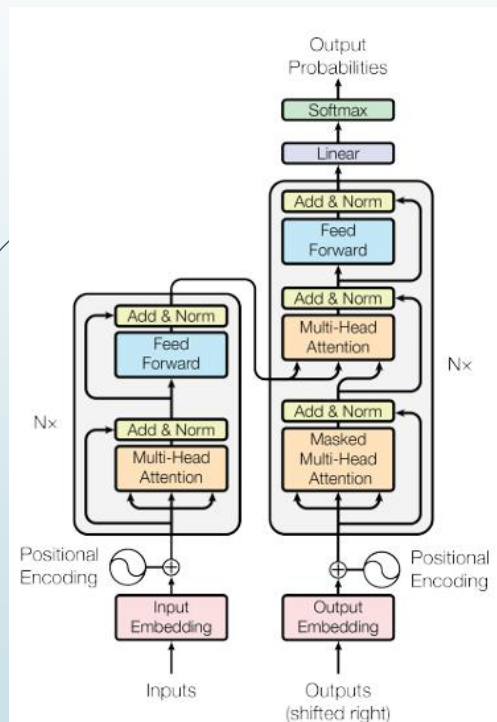
A standard Transformer architecture, showing on the left an encoder, and on the right a decoder.

**Self-attention sentence example**

Original GPT architecture

Wrap up

- Even without GT, LLM still can use the self-supervised learning to predict the next output.

- Self-attention can help find inter-sentence words/tokens how close they are.

- GPT uses the decoder part of Transformer, by adding attention to generate the next word/token.

LLM Key Features - Scaling Laws

- OpenAI in 2020, published a **Scaling laws** show Language modeling performance improves smoothly as the increasing of the model size, dataset size, and amount of compute used for training.

- However, Google Research proposed **Chain-of-Thought** in 2022, showing that LLMs have challenges on solving more complex tasks, which need more reasoning.
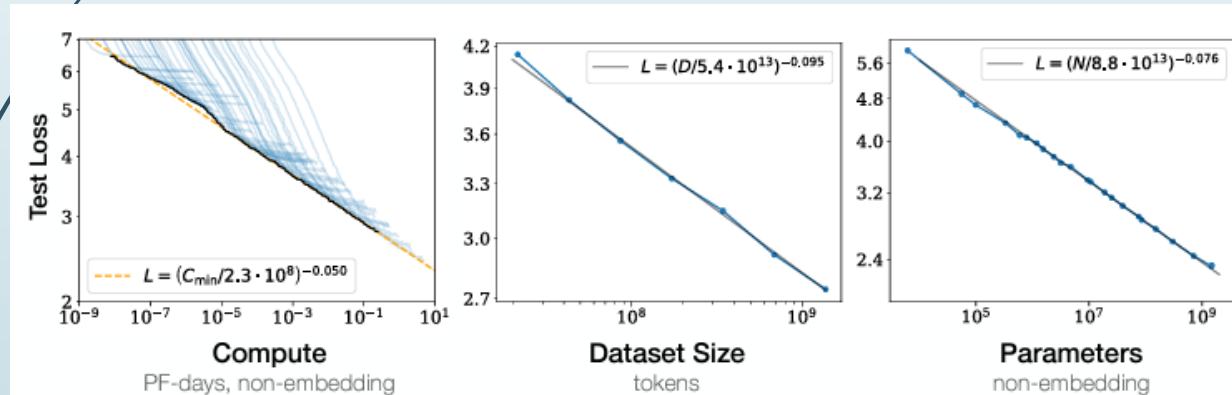
OpenAI published a paper in 2020 by showing that Language modeling performance improves smoothly as the increasing of the model size, datasetset size, and amount of compute used for training.
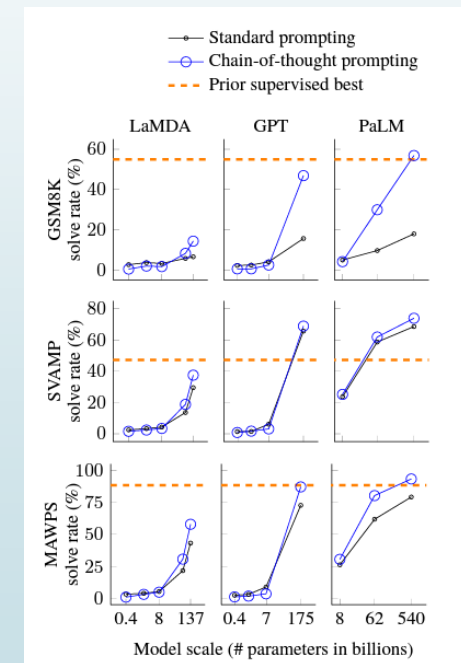
Chain-of-thought prompting enables large language models to solve challenging math problems.

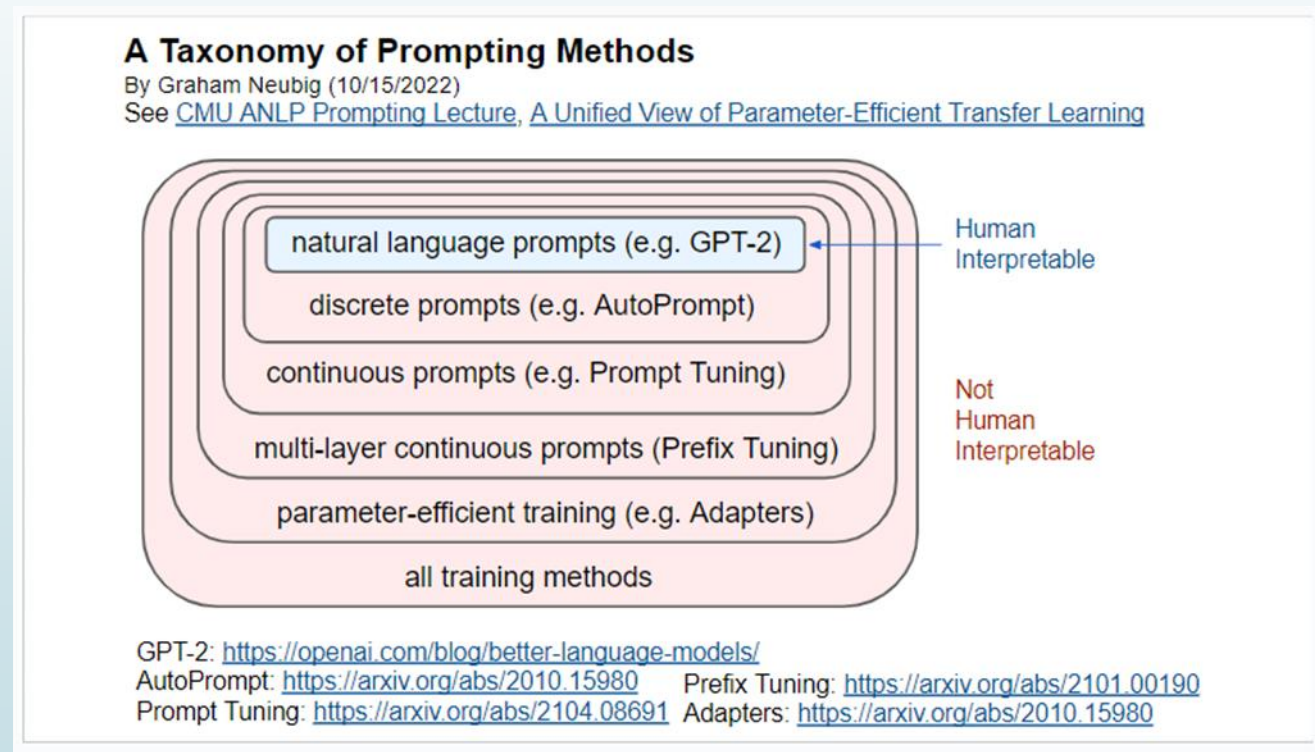# LLM Key Features -Prompt Engineering

- Chain-of-Thought is a type of prompt engineering. Actually, there are many types of prompt engineering.

- There re many prompting methods.

A Taxonomy of Prompting Methods
By Graham Neubig (10/15/2022)
See CMU ANLP Prompting Lecture, A Unified View of Parameter-Efficient Transfer Learning

natural language prompts (e.g. GPT-2) — Human Interpretable

discrete prompts (e.g. AutoPrompt)

continuous prompts (e.g. Prompt Tuning) — Not Human Interpretable

multi-layer continuous prompts (Prefix Tuning)

parameter-efficient training (e.g. Adapters)

all training methods

GPT-2: https://openai.com/blog/better-language-models/
AutoPrompt: https://arxiv.org/abs/2010.15980   Prefix Tuning: https://arxiv.org/abs/2101.00190
Prompt Tuning: https://arxiv.org/abs/2104.08691   Adapters: https://arxiv.org/abs/2010.15980

A taxonomy of prompting methods

- Full fine-tuning (2018)

  - Fine-tune all the model parameters to adapt general-purpose PLMs to downstream tasks.

    - Per task, it is access to all the tasks datasets simultaneously.

    - Co-train the LM weights and top-layer weights.

    - Train 100% parameters efficient.

- Parameter-efficient fine-tuning/training: Adapter tuning/Adapters (2019)

  - Add only a few trainable parameters per task, and new tasks can be added without revisiting previous ones. Train on tasks sequentially.

    - Inject new layers (only a small number of additional parameters per task) into the original network.

    - The weights of the original network are untouched, whilst the new adapter layers are initialized at random.

    - Train only about 3.6% parameters per task.

- Prompting/prompt engineering (2020)

  - Human interpretable (textual) prompting/prompt engineering/prompt design/priming

    - In-context learning (GPT-3)

    - Instruction-tuning (is also a type of prompt-tuning)

    - Normal zero-shot, one-shot, few-shot prompting, etc.

  - Human uninterpretable (discrete, continuous, etc.) prompting/prompt engineering

Elements of a Prompt

➡ A prompt can contain information like the ***instruction*** or ***question*** you are passing to the model and include other details such as ***context***, ***input (sometimes with examples)***, and ***output indicator*** .

➡ **Instruction** - a specific task or instruction you want the model to perform

➡ **Context** - external information or additional context that can steer the model to better responses

➡ **Input Data** - the input or question that we are interested to find a response for

➡ **Output Indicator** - the type or format of the output.

```
Classify the text into neutral, negative, or positive
Text: I think the food was okay.
Sentiment:
```

An example of prompt

Zero-shot prompting example

- E.g., there is no prompt example given to the model while doing classifications, however, the LLM already understands "sentiment" -- that's the zero-shot capabilities at work.
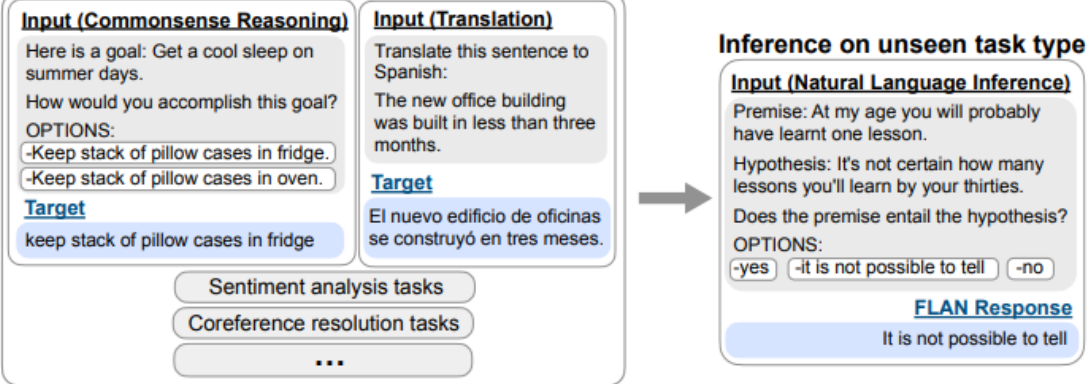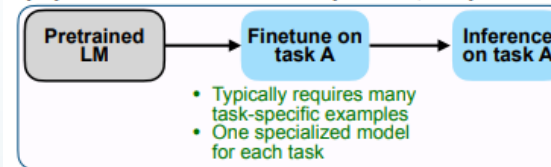


```
Prompt:

    Classify the text into neutral, negative or positive.
    Text: I think the vacation is okay.
    Sentiment:

Output:

    Neutral
```
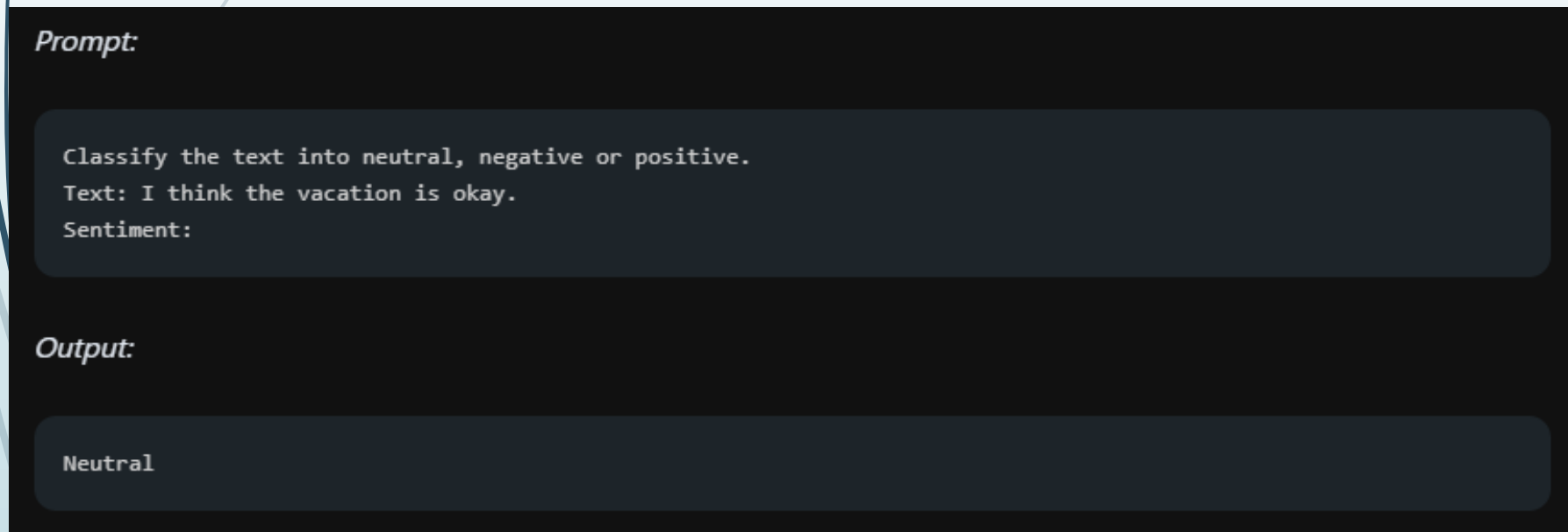
There is no prompt example

Figure 01: An example of zero-shot prompting

Instruction tuning for zero-shot prompting experiment

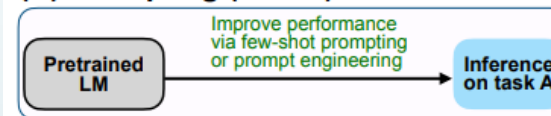Overview of instruction tuning and FLAN
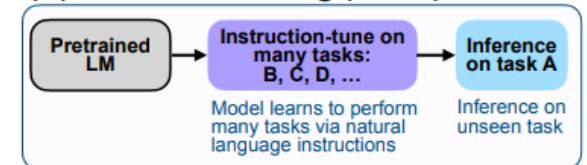
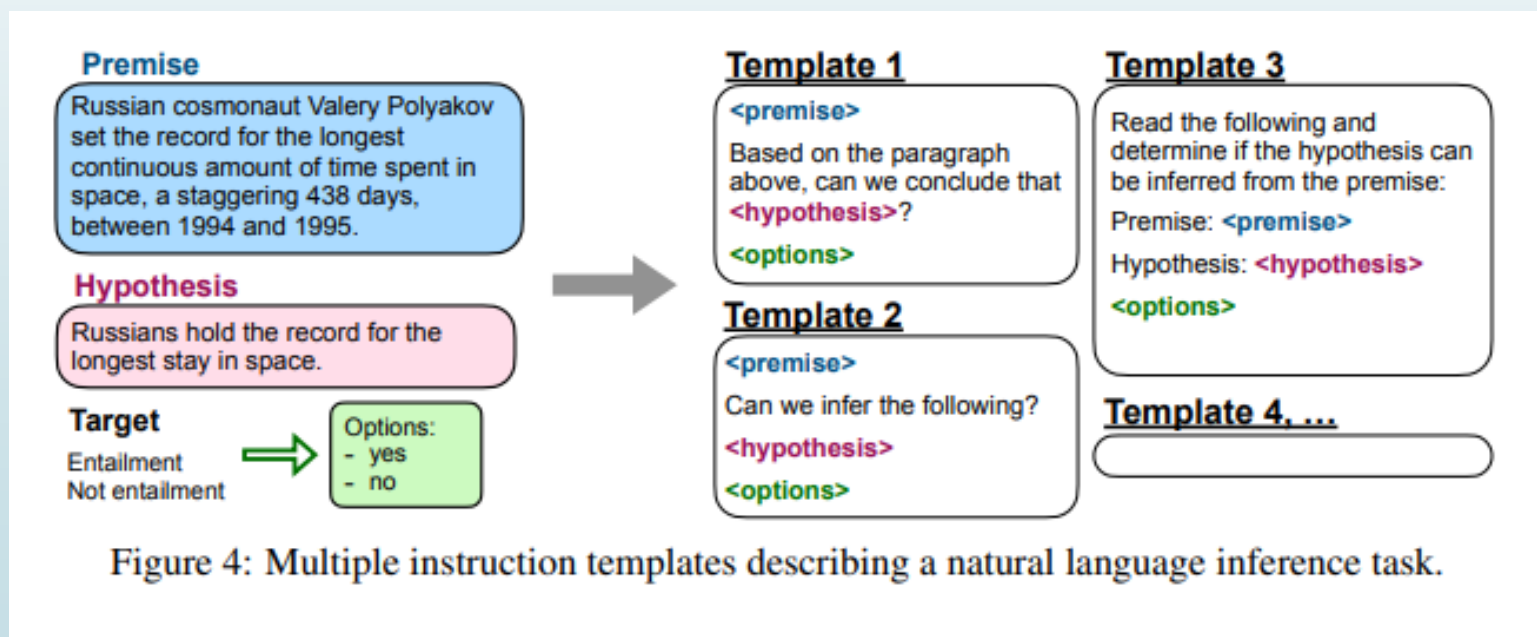Comparing instruction tuning with pretrain–finetune and prompting
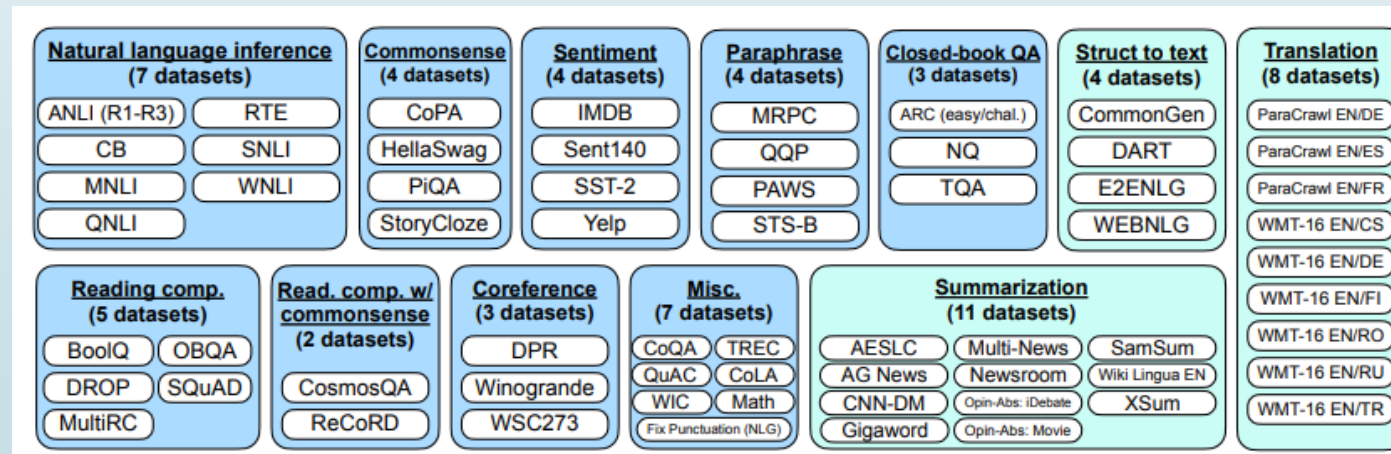
- Dataset

Datasets and task clusters used in this paper (Natural Language Understanding tasks in blue; Natural Language Generation tasks in teal)

Unique templates as instruction

- Unique template

  - For each dataset, the authors manually compose ten unique templates that use natural language instructions to increase diversity.

  - For each dataset they also include up to three templates that "turned the task around" (to increase the diversity of the task).

Premise

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

Hypothesis

Russians hold the record for the longest stay in space.

Target

Entailment
Not entailment

Options:
- yes
- no

Template 1

<premise>

Based on the paragraph above, can we conclude that <hypothesis>?

<options>

Template 2

<premise>

Can we infer the following?

<hypothesis>

<options>

Template 3

Read the following and determine if the hypothesis can be inferred from the premise:

Premise: <premise>

Hypothesis: <hypothesis>

<options>

Template 4, ...

Figure 4: Multiple instruction templates describing a natural language inference task.

Few-shot prompting example

- In the example, the task is to correctly use a new word in a sentence by providing a prompt example.

Prompt:

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.   → Prompt example

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is: → task/question

Output:

When we won the game, we all started to farduddle in celebration.

An example of few-shot prompting

CoT prompting example

Standard prompting

Input: Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

...

Q: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?
A:

Model output: The answer is 50. ✖

Chain of thought prompting

Input: Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

...

Q: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?
A:

Model output: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. So that is 10 x .5 = 5 hours a day. 5 hours a day x 7 days a week = 35 hours a week. The answer is 35 hours a week. ✔

Few-shot prompt learning:
given a few exemplar prompts of Q and A sets directly.

⇩

(Not just the answer, but also please tell me the logic/reason/inference why you give this answer!)

⇩

Few-shot manual-CoT prompt learning:
Not just providing the exemplar prompts of Q and A sets, but also the CoT of A.

Chain of thought (highlighted) facilitates multistep reasoning in large language models. The output here is from a 137B parameter LaMDA language model.
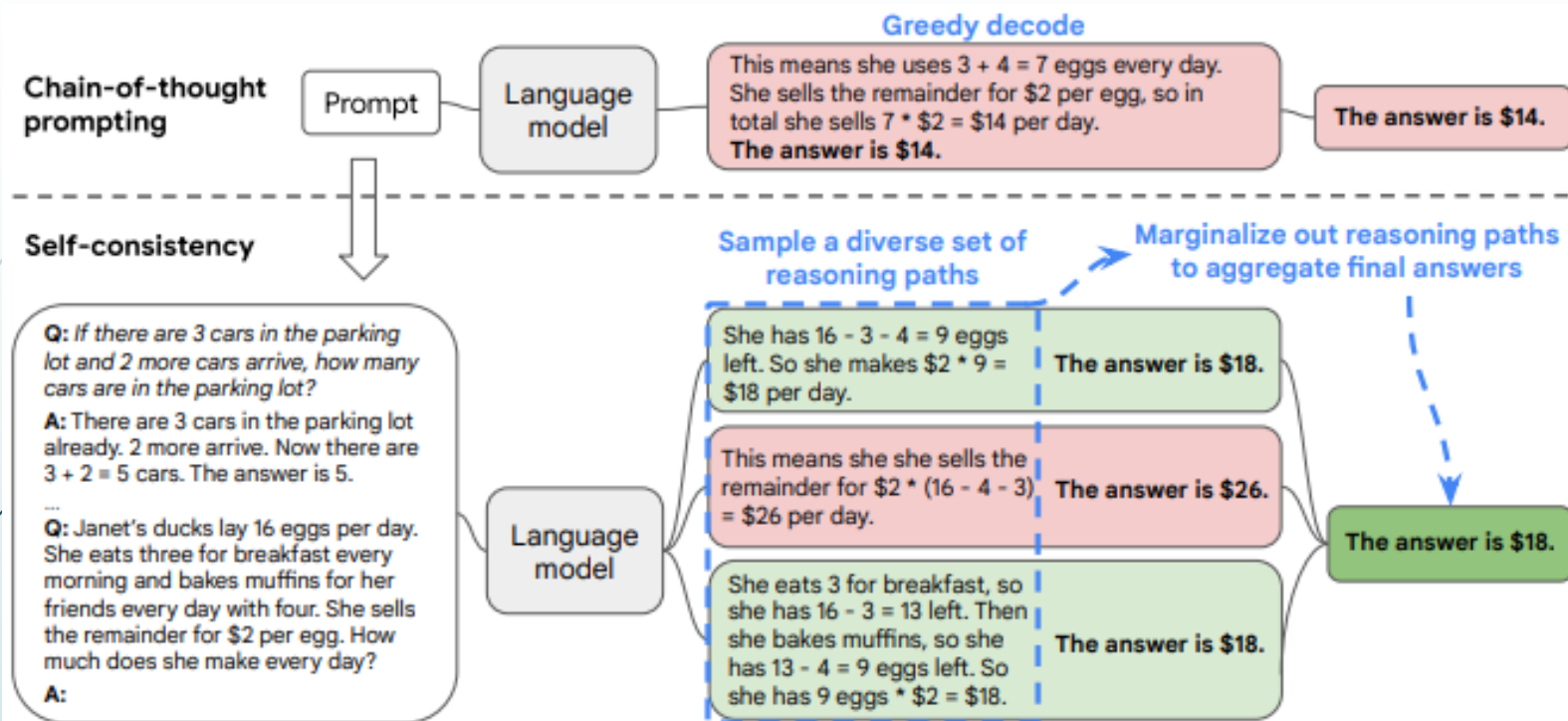
Self-consistency example

Figure 1: The self-consistency method contains three steps: (1) prompt a language model using chain-of-thought (CoT) prompting; (2) replace the "greedy decode" in CoT prompting by sampling from the language model's decoder to generate a diverse set of reasoning paths; and (3) marginalize out the reasoning paths and aggregate by choosing the most consistent answer in the final answer set.

The comparison between CoT prompting and Self-consistency

Tips on Prompt Engineering

- Use prompt engineering technical introduced in this slides in your prompt.
  - Zero-shot, Few-shot, CoT, self-consistency prompting
- Adding a role into your prompt.
  - You are a Mathematical expert…
- Decompose the question into small pieces for LLM to understand.
- How to write ChatGPT prompt:
  - https://www.coursera.org/articles/how-to-write-chatgpt-prompts

Wrap up

- Scaling law is not really the truth.

- The prompt engineering commonly known is hard prompting.

- Zero-shot prompting, zero-shot prompting, CoT prompting are the most commonly used hard prompt engineering methods.

  - All there of them are also called In-context learning (ICL).

    - ICL is a model's ability to adapt to input without changing its internal parameters, while few-shot prompting is a technique within ICL that uses examples to guide the model's behavior.

- Self-consistency prompting aims to increase the trustworthiness and robustness of an LLM.

RAG

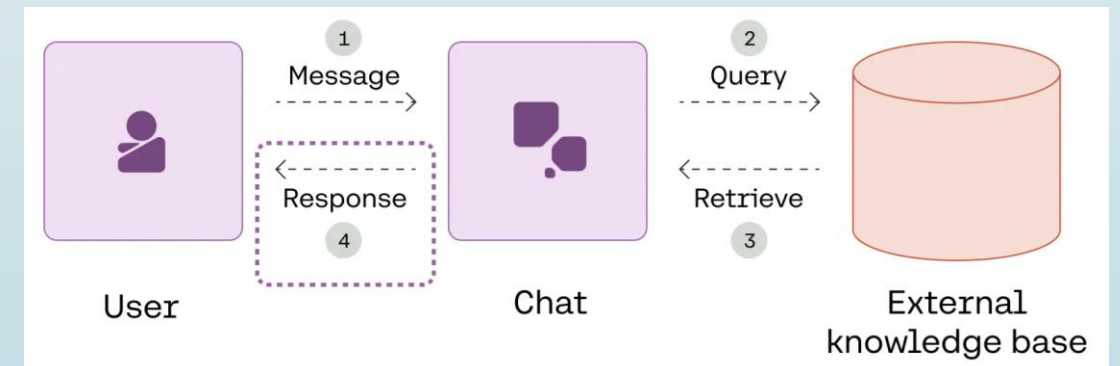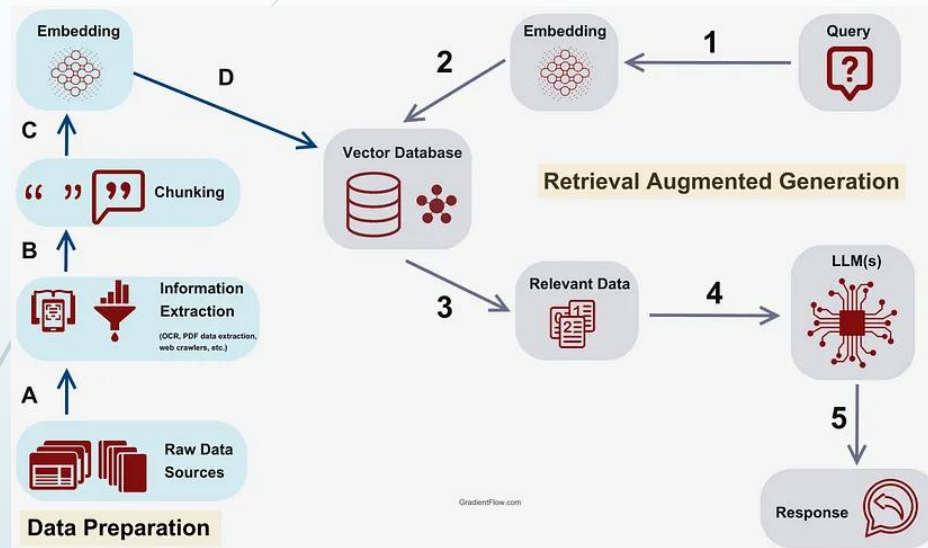| RAG |
|---|
| Reason: Normal LLM can not answer internal domain questions (e.g., company related information). |
| Technical: Relies on in-context learning: source material is included as part of the prompt when generating. |
| No model training (no parameter (weight, bias) update) |
| Use case: add knowledge to the model |

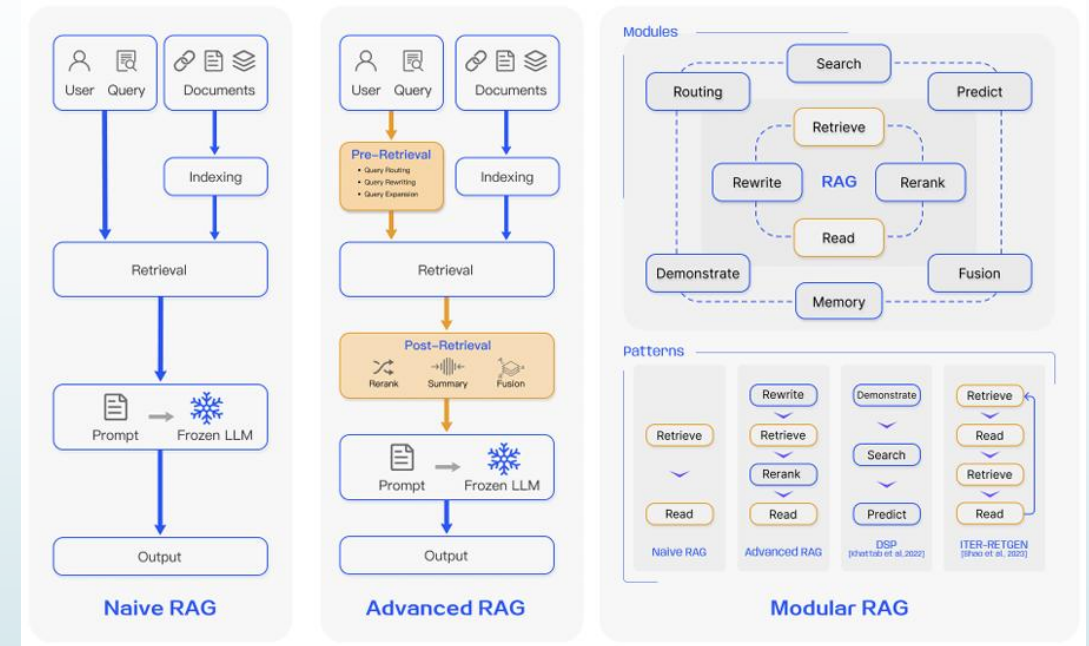## What RAG Adds

How RAG works

Source: Introduction to RAG
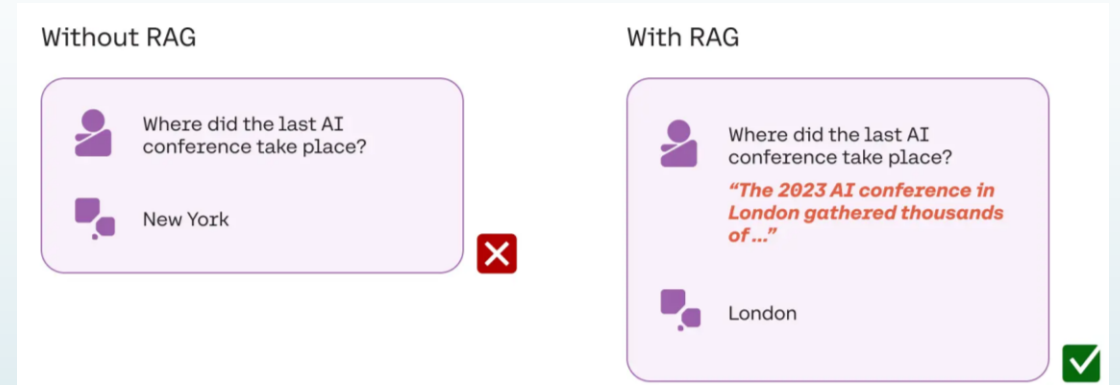
Details of RAG

The process of Naïve RAG

Comparison between the three paradigms of RAG. (Left) Naive RAG mainly consists of three parts: indexing, retrieval and generation. (Middle) Advanced RAG proposes multiple optimization strategies around pre-retrieval and post-retrieval, with a process similar to the Naive RAG, still following a chain-like structure. (Right) Modular RAG inherits and develops from the previous paradigm, showcasing greater flexibility overall. This is evident in the introduction of multiple specific functional modules and the replacement of existing modules. The overall process is not limited to sequential retrieval and generation; it includes methods such as iterative and adaptive retrieval.
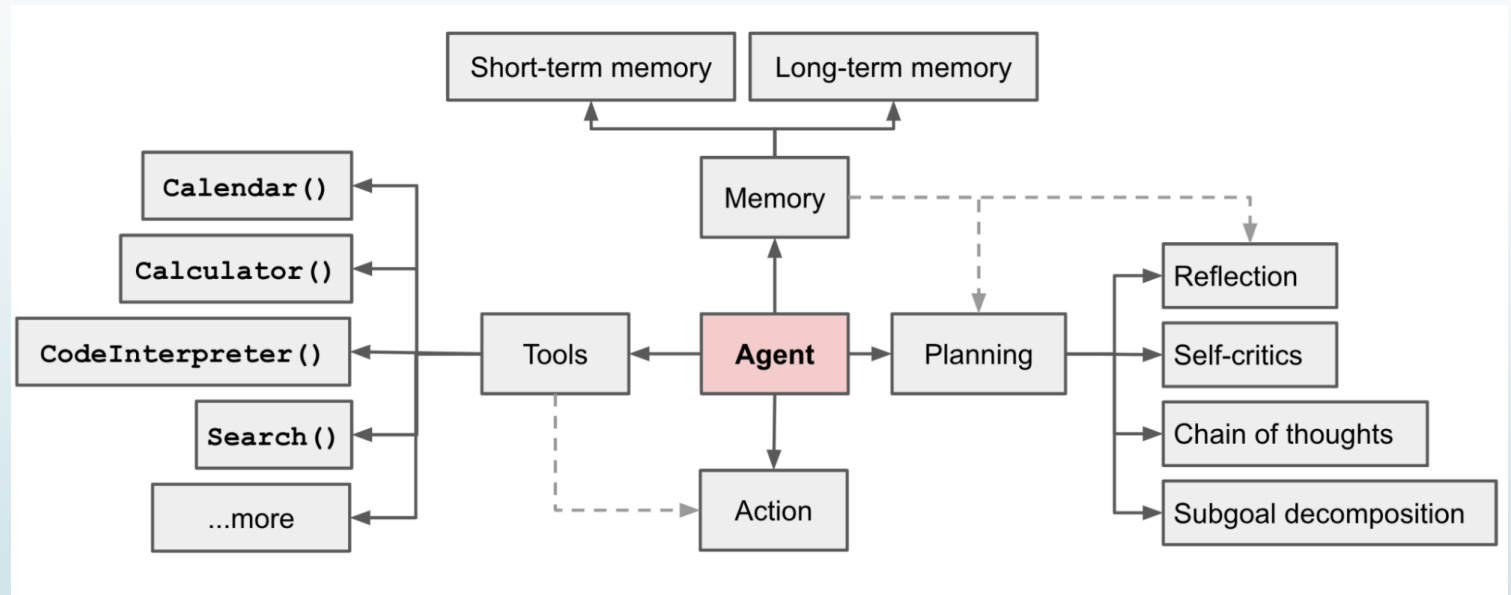
AI Agent

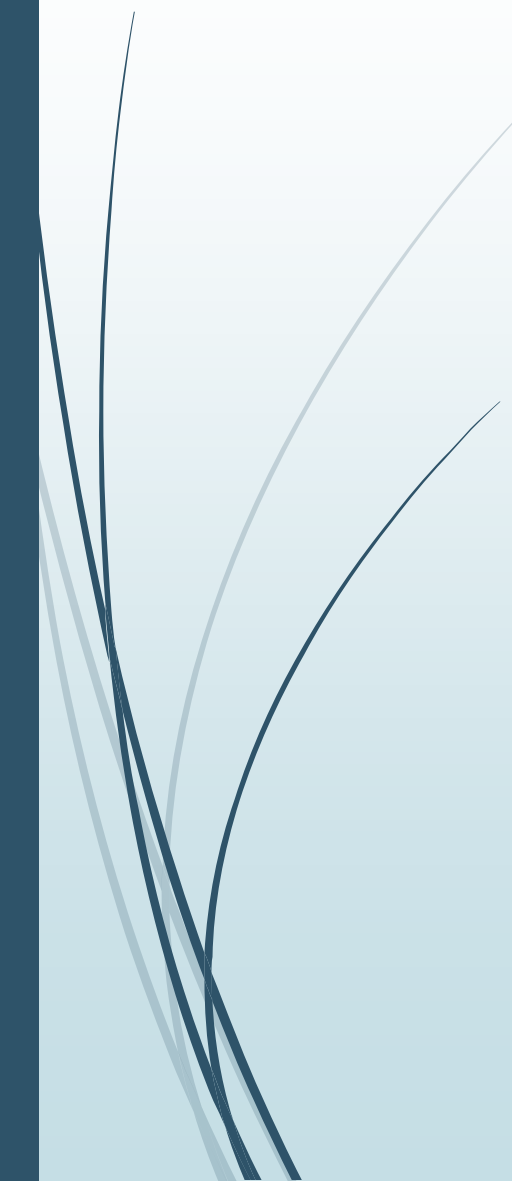| AI Agent |
| --- |
| Reason: The question involves a complex problem, thus normal LLM can not solve the problem. |
| Technical: Relies on the tool-use of RAG: Let LLM to planning, use tools and take action by it self. |
| No model training (no parameter (weight, bias) update) |
| Use case: Plan and Use different tools to solve more complex problems |

Overview of an LLM-powered autonomous agent system.

Wrap-up

- RAG is a special prompt engineering use LLM to retrieve internal-domain data.

- AI Agent uses LLM to analysis the question, and decompose the question into several smaller problems. And each problem uses different tools. Each tool takes different action to help solve the problem. Memory is used to remember the previous actions and results to better help LLM make plans.
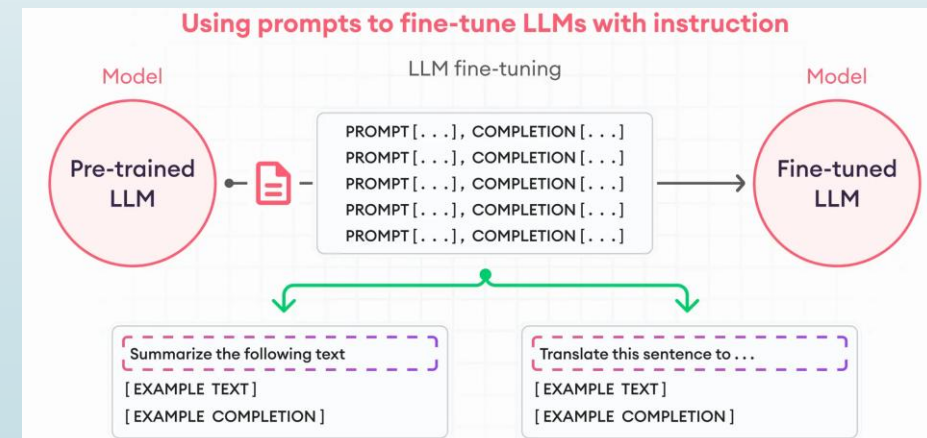
Fine-tuning

➡ Usually, **instruction tuning** is used for fine-tuning an LLM.
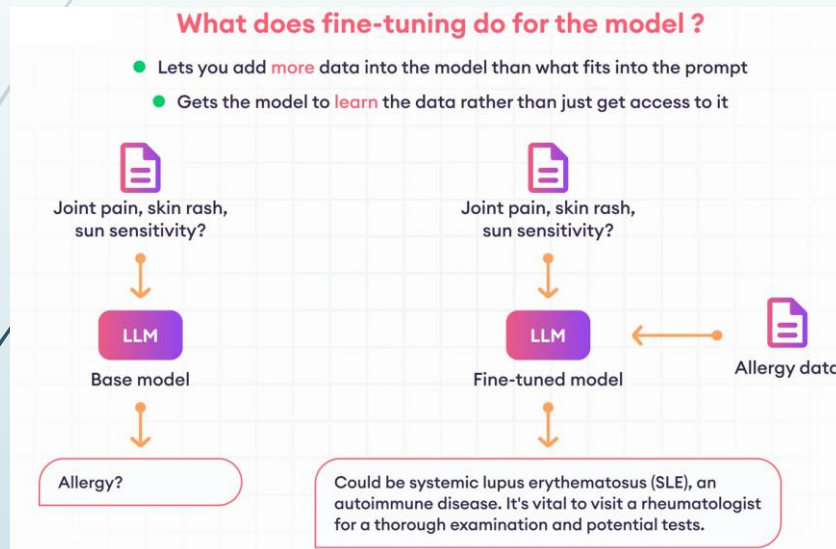
For the same question, the results comparison of using normal LLM and fine-tuned LLM.
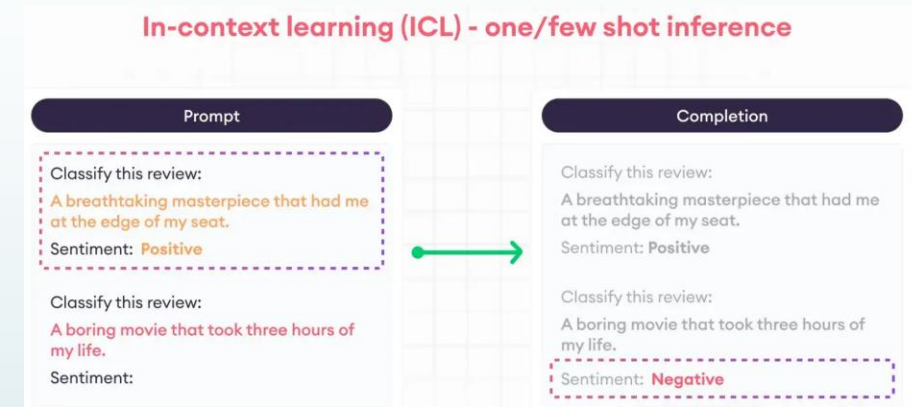

An instruction data format

The process of instruction tuning

PEFT

- The high cost of full-fine tuning has inspired researchers to develop cheaper approaches

- LORA (Low Rank Adaptation) is representative of PEFT approaches: Train 2 matrices with lower rank than the original, but whose product yields the same rank as the original matrix

- Fewer parameters means it's cheaper to train, and cheaper to serve multiple PEFT models simultaneously

PEFT
Reason: Pretraining or fine-tuning an LLM is time and cost consuming.
Technical: Rank decomposition. Let $W = AB$, the rank of A and B is far lower than W.
A smaller number of parameters (weight, bias) are updated
Use case: The downstream task dataset is not big.
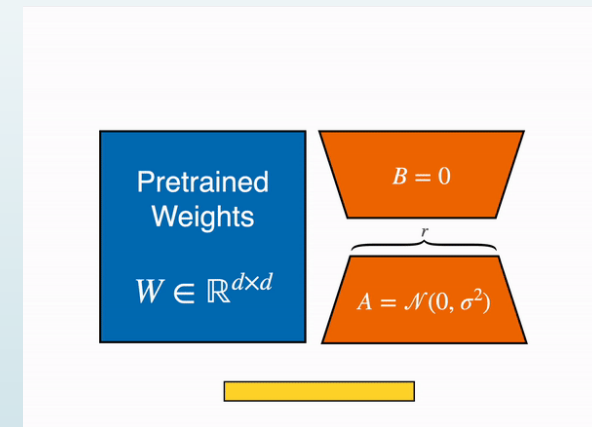
The technique constrains the rank of the update matrix ΔW using its rank decomposition. It represents ΔW_{nk} as the product of 2 low-rank matrices B_{nr} and A_{rk} where $r << min(n, k)$. This implies that the forward pass of the layer, originally Wx, is modified to $Wx + BAx$ (as shown in the figure below). A random Gaussian initialization is used for A and B is initially to 0, so $BA=0$ at the start of training. The update BA is additionally scaled with a factor α/r.

Wrap up

Increasing complexity and cost but increasing customizability

RAG	PEFT	SFT
• Retrieval Augmented Generation • No training • In-context learning	• Parameter Efficient Fine Tuning • Minimal training • Minimal data	• Supervised Fine Tuning (especially, instruction tuning) • High level behavior: for specific domain or function (e.g., chat, healthcare...) • More data

* Approaches are not mutually exclusive

LLM Evaluation

Key Metrics for LLM Evaluation

Accuracy and Performance
- Perplexity: Predicts next word accuracy
- Accuracy: Proportion of correct predictions
- BLEU: Precision-based text comparison
- ROUGE: Recall-oriented text evaluation

Bias and Fairness
- Demographic parity: Consistent across groups
- Equal opportunity: Even error distribution
- Counterfactual fairness: Sensitivity to attributes

Language Quality
- Fluency: Naturalness and grammar
- Coherence: Logical flow and consistency

Content Quality
- Factuality: Accuracy of information
- Relevance: Appropriateness to context
- Diversity: Variety in generated content

The key points to evaluate an LLM

LLM Evaluation Methodologies

Benchmark Datasets
- Existing benchmarks (GLUE, SuperGLUE, SQuAD)
- Custom datasets for domain-specific evaluation
- Enables standardized comparative analysis
- Establishes baseline performance

Human Evaluation
- Direct assessment through surveys and ratings
- Comparative judgment (e.g., pairwise comparison)
- Captures nuanced aspects of text quality
- Provides qualitative insights

Automated Evaluation
- Metric-based (e.g., perplexity, BLEU)
- Quick and objective assessment
- Quantifies various aspects of model outputs

Adversarial Evaluation
- Tests model robustness against attacks
- Reveals vulnerabilities and biases
- Important for reliability and security
- Helps identify and mitigate potential risks

The key methods to evaluate an LLM

Attention: LLM evaluation is not RAG evaluation. They are different.

Wrap-up

- LLM belongs to the IR4.0
- Deep learning is a subfield of machine learning which uses artificial neural networks to enable artificial intelligence systems to learn complex patterns/regularities from data.
- LLM's learning method is self-supervised learning. Since it is regression machine learning, so it is also called as self-regression learning or auto-regression.
- The current main architecture of LLMs is still transformer (self-attention).
- Scaling law is not correct, when reaching some point of scaler, combining prompt engineering is necessary.
- Zero-shot, few-shot, CoT, self-consistency, Role Prompting are commonly used prompting methods in our daily life.
- Learning how to interact with LLM and decompose the question for LLM to understand are important.
- RAG, AI Agent are the future main use-cases to use LLMs.
- PEFT is more efficient while the training data size is small.
- An LLM is evaluated through different perspectives using different metrics.

Thank you