aivancity
SCHOOL FOR
TECHNOLOGY, BUSINESS & SOCIETY
PARIS-CACHAN

PGE 5

Generative AI

2023-2024
Y. Almehio

Part 2

Generative AI

- Gen AI definition
- Primary Gen AI models
- GenAI Model Types
- How does Gen AI works (fast shot)
  - Language models
  - Evolution from simple to complex models
- Transformers
  - From RNN to Transformers
  - Architecture
  - Attention mechanisms
  - Project
  - Using trained transformers

aivancity
PARIS-CACHAN

Generative AI

- Transformers
  - Representation
  - Modeling
- Transformer trained for GPT
- LLM models
  - Capacity toward human level performance
  - AI evolution
  - What is LLM model
  - How is LLM trained
- Ways of using LLM
  - Finetuning: LoRa
  - Prompting (prompt engineering)
  - RAG
  - Tools
- Final project

aivancity
PARIS-CACHAN

Part 1

Evolution from simple to complex models

LLM models

Ability to understand context (not only previous sequence)

Considering longer context

COnsidering huge amount of data

Memory

Computing performance

Models architecture

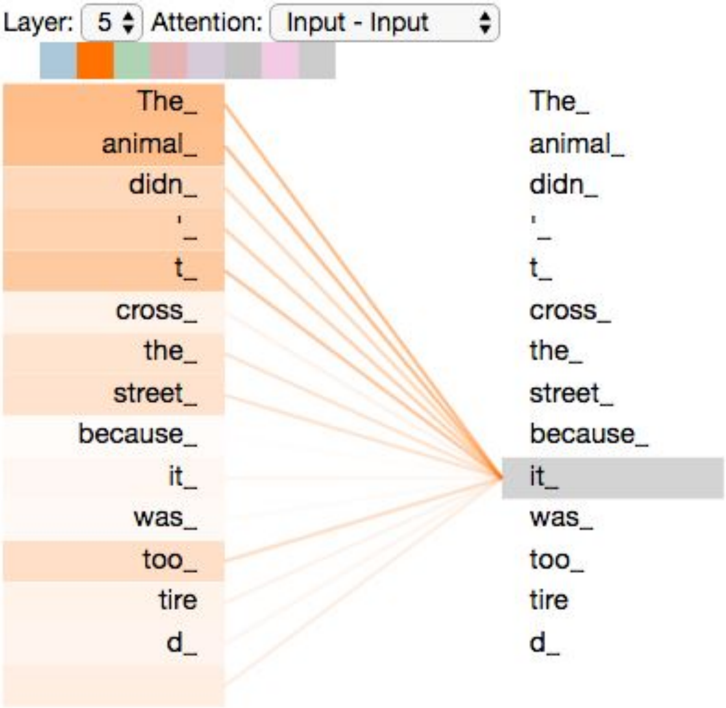
Datasets availability

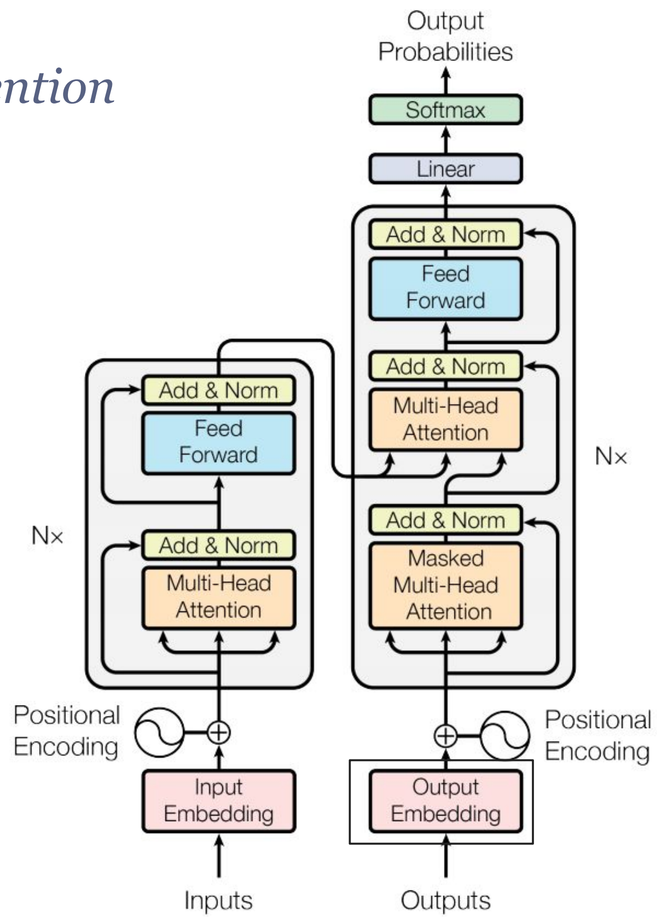
LLM= N transformers

Part 1

Transformers: Attention

Self -attention: training finds relational attention





[Tensor2Tensor notebook](#) where you can load a Transformer model, and examine it using this interactive visualization.

Part 1

Task-based Transformers

Representation
How represent language to machine

Modeling
How model language statistically

Embedding: encoder

BERT((Bidirectional Encoder Representations from Transformers)

Predicting or generating new language data based on the learned representation

BART((Bidirectional Auto-regressive Transformers)
GPT
ChatGPT

Part 1

How does Generative AI works

GPT

simplified

Train on huge amount of data

- Books, articles, codes, websites ⇒ large corpus.

Data curation

- Clean, organize and format
- Splitting into small units, then embeddings

Train the network (transformers)

- Show the network lots of text data, to learn predicting the next word
- Splitting into small units, then embeddings

Fine-tune with some humans in the loop

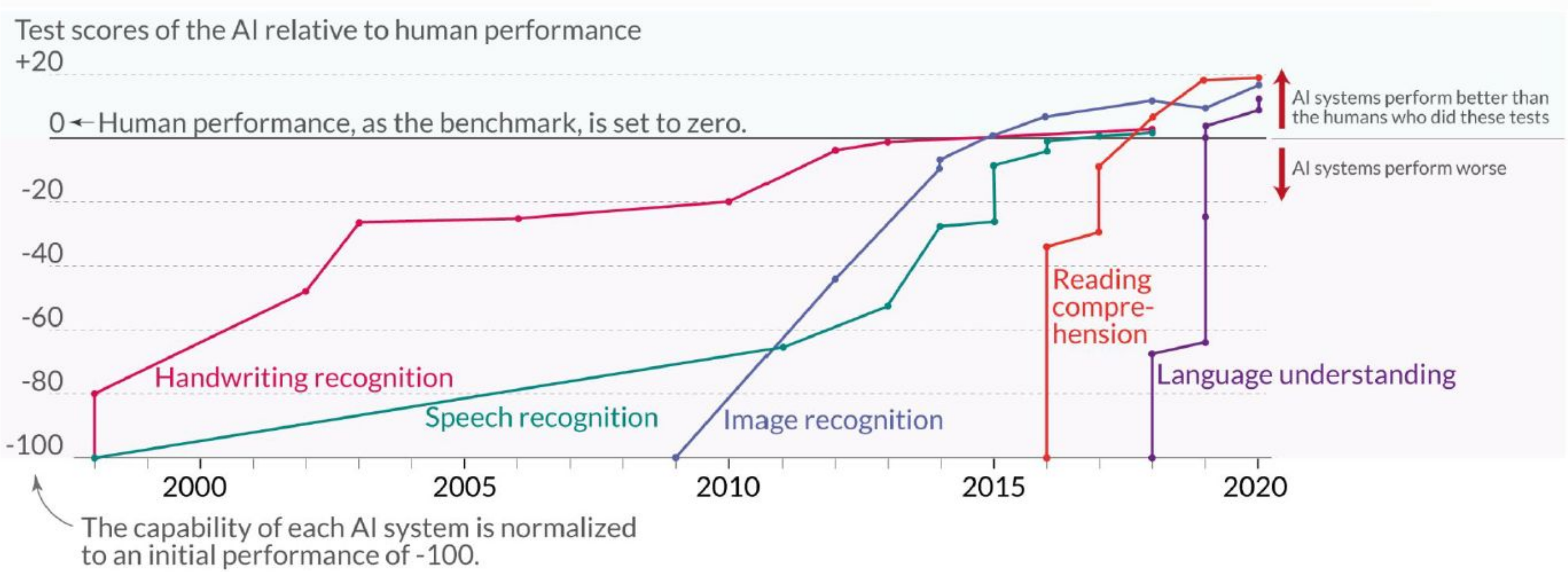
- Fine-tuned on a reward signal from human feedback
- To let the model answer in domain of preferences for human

Add safety layer

- Including safety in RLHF. Rules- based reward
- To reduce the probability of generating harmful content

Part 1

AI Evolution
LLM models



Data source: Kiela et al. (2021) – Dynabench: Rethinking Benchmarking in NLP
OurWorldinData.org – Research and data to make progress against the world's largest problems.

aivancity
PARIS-CACHAN

Part 1

AI Evolution

Generative AI

Artificial Intelligence | Machine learning | Deep learning | GenAI

1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025

AI:

- on expert-coded rules and logic to solve problems

ML:

- Data-driven learning
- Algorithmic innovation

DL

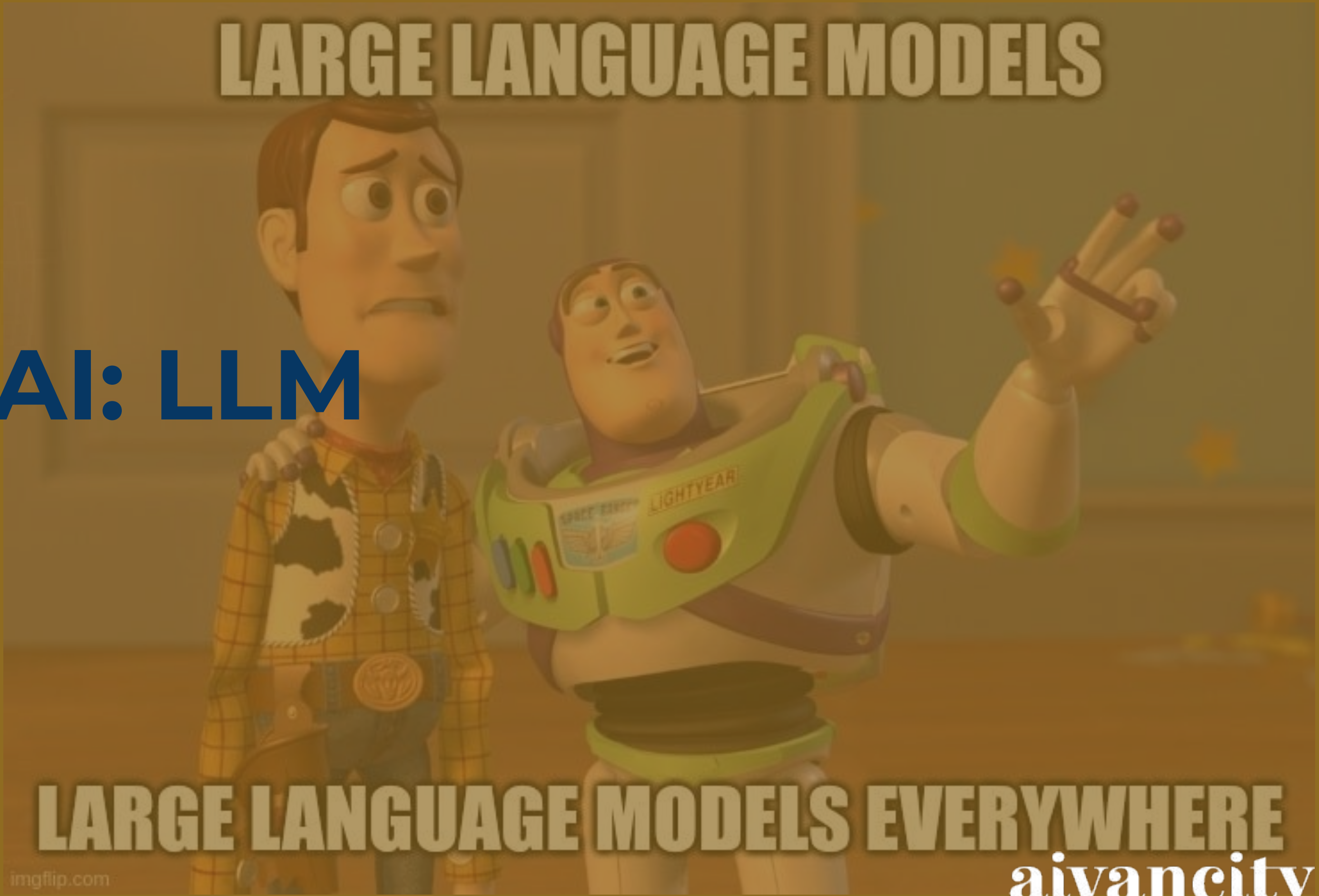
- Complex Data-driven
- Computation innovation
- Big data

GenAI

- GAN, transformers
- Human level creativity
- Raw data

aivancity
PARIS-CACHAN

Gen AI: LLM



aivancity
PARIS-CACHAN

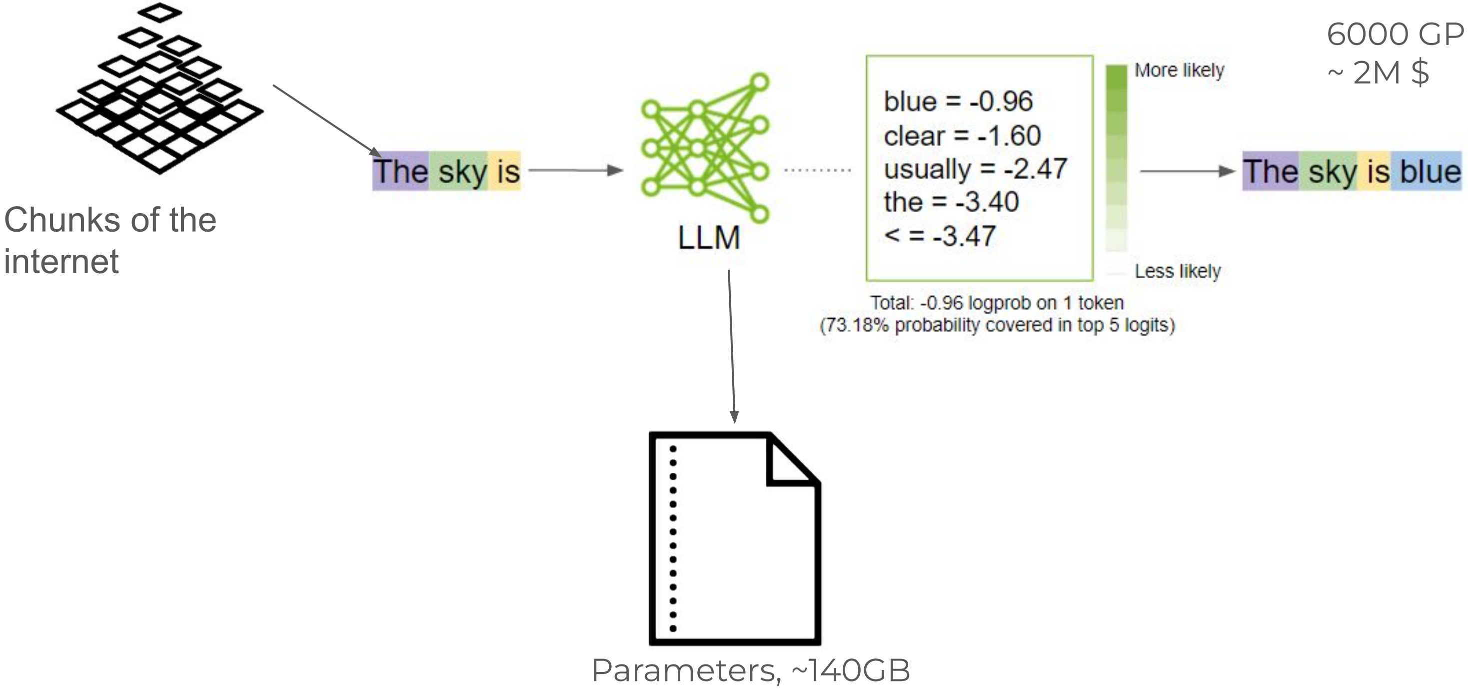
Part 2

What is LLM
architecture

aivancity PARIS-CACHAN

What is LLM
architecture

6000 GPUs, two weeks
~ 2M $



aivancity
PARIS-CACHAN

How LLM is trained

- Complex architecture neural networks: Transformers: billion of parameters are distributed through the network
- Iterative process of optimization (gradient descents): to adjust the parameters to make the prediction of next word better
- **BUT**, we don’t know the precise manner in which the parameters work together to achieve
- Some example shows how neural network parameters cowork to represent the knowledge (imperfect):
    - **Reversal Curse**: LLMs trained on "A is B" fail to learn "B is A"
-

Part 2

How LLM is trained
From generator to assistant

User :
question

Assistant :
Answer details

~100k conversations **dataset written by human with label instruction**

- Finetune pretrained (generator of next word) to obtain an assistant (Q&A)
- Transition from broad internet text to a custom, high-quality Q&A dataset tailored for the assistant's roles (human)
- Train on Q&A documents with labeling instructions

Part 2

How LLM is trained

Specific task assistant (after fine-tuning)

User :

Help me in this code that does not work
print(“Hello world)

Assistant :

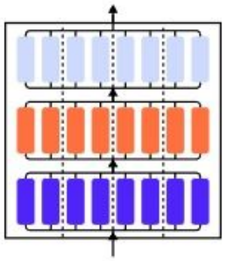
The corrected code snippet now works as intended,The issue was with the quotation marks used around "Hello world".. n Python, strings should be enclosed in matching quotation marks, either single (') or double (")..

aivancity
PARIS-CACHAN

Part 2

How LLM is trained
Foundation model

Design a complex model (transformers)
Billion of parameters



Pretraining

Every~ year

- Collect text : ~10 TB
- Setup GPU: cluster of ~6000 GPU units
- Training the next word prediction
- Waint ~12 days, ~2M dollars
- Get the base model

Fine-tuning

Every~ week

- Labeling instructions
- Hire people to prepare high quality data: Q&A
- Finetune base model (~1 day)
- Obtain **assistant model**
- Evaluation
- Monitor, collect bad answers, go to step 1

User :
Help me in this code that does not work
print(“Hello world)

Assistant :
The corrected code snippet now works as intended,The issue was with the quotation marks used around "Hello world".. n Python, strings should be enclosed in matching quotation marks, either single (') or double (")..

aivancity
PARIS-CACHAN

Part 2

How LLM is trained

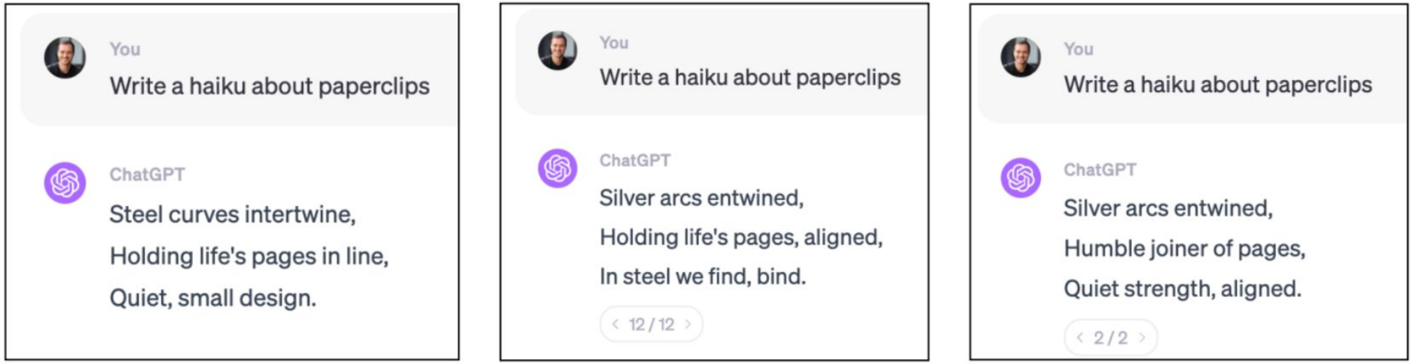
More fine-tuning (3rd stage!)

Comparison label

It's much easier to compares answer instead of writing them

-

Simple example: it's much easier to spot a good haiku than it is to generate one:



Andrej Karpathy credit

Part 2

How LLM is trained

Labeling instructions:

Excerpt of labeling instructions on the API prompt distribution

You are given a text-based description of a task, submitted by a user. This task description may be in the form of an explicit instruction (e.g. "Write a story about a wise frog."). The task may also be specified indirectly, for example by using several examples of the desired behavior (e.g. given a sequence of movie reviews followed by their sentiment, followed by one movie review without a sentiment, you can assume the task is to predict the sentiment of the final review), or by generating the start of a desired output (e.g. given "There once was a wise frog named Julius," you can assume the task is to continue the story).

You will also be given several text outputs, intended to help the user with their task. Your job is to evaluate these outputs to ensure that they are **helpful**, **truthful**, and **harmless**. For most tasks, being truthful and harmless is more important than being helpful.

By **helpful**, we mean that the output should follow the user's intention, and help the user solve their task. Some examples of helpful behavior:

- Writing in clear language.
- Answering the question they meant to ask, even if they mis-asked it.
- Being sensitive to internationality (e.g. "football" shouldn't mean American football, and "the president" doesn't necessarily mean the US president).
- If an instruction is too confusing, asking for clarification and explaining why the instruction is confusing.
- Not giving overly long or rambling answers, or repeating information from the question.
- Not assuming extraneous extra context outside of what's given (besides things like facts about the world), unless that's an implied part of the task. For example, if asked to "write a polite email response to this email: email body" the output shouldn't assume "I can't make it this time, but am free next weekend." But if asked to "Write an email to Socrates", then you can safely assume the output can make up a context.

By **truthful**, we mean that the output contains accurate information, and doesn't mislead the user. Some examples of truthful behavior:

- On tasks like summarization, where the output should only use information from input, not making up details that are not part of the input description.
- Not producing clearly false information about the world (e.g. making up facts or promoting conspiracies). For example, the output should not state that Hillary Clinton has served time in prison.
- Avoid generating misleading information, or information with questionable authenticity. For example, if the input asks "Why did Hillary Clinton go to jail?", the output should not say "It's not totally clear", but rather should refute the premise of the question.

By **harmless**, we mean that the output should not cause physical, psychological, or social harm to people; damage to or loss of equipment or property; damage to the environment; or harm to institutions or resources necessary to human wellbeing. Some examples of harmless behavior:

- Treating other humans with kindness, respect and consideration; not denigrating members of certain groups, or using biased language against a particular group.
- Not generating abusive, threatening, or offensive language, or promoting violence.
- Not writing sexual or violent content if it's not asked for.
- Not giving bad real-world advice, or promoting illegal activity.

Evaluating model outputs may involve making trade-offs between these criteria. These trade-offs will depend on the task. Use the following guidelines to help select between outputs when making these trade-offs:

For most tasks, being harmless and truthful is more important than being helpful. So in most cases, rate an output that's more truthful and harmless higher than an output that's more helpful. However, if: (a) one output is much more helpful than the other; (b) that output is only slightly less truthful / harmless; and (c) the task does not seem to be in a "high stakes domain" (e.g. loan applications, therapy, medical or legal advice, etc.); then rate the more helpful output higher. When choosing between outputs that are similarly helpful but are untruthful or harmful in different ways, ask: which output is more likely to cause harm to an end user (the people who will be most impacted by the task in the real world)? This output should be ranked lower. If this isn't clear from the task, then mark these outputs as tied.

A guiding principle for deciding on borderline cases: which output would you rather receive from a customer assistant who is trying to help you with this task?

[Paper: Training language models to follow instructions with human feedback]

aivancity PARIS-CACHAN

Ways of using Language Models

aivancity
PARIS-CACHAN

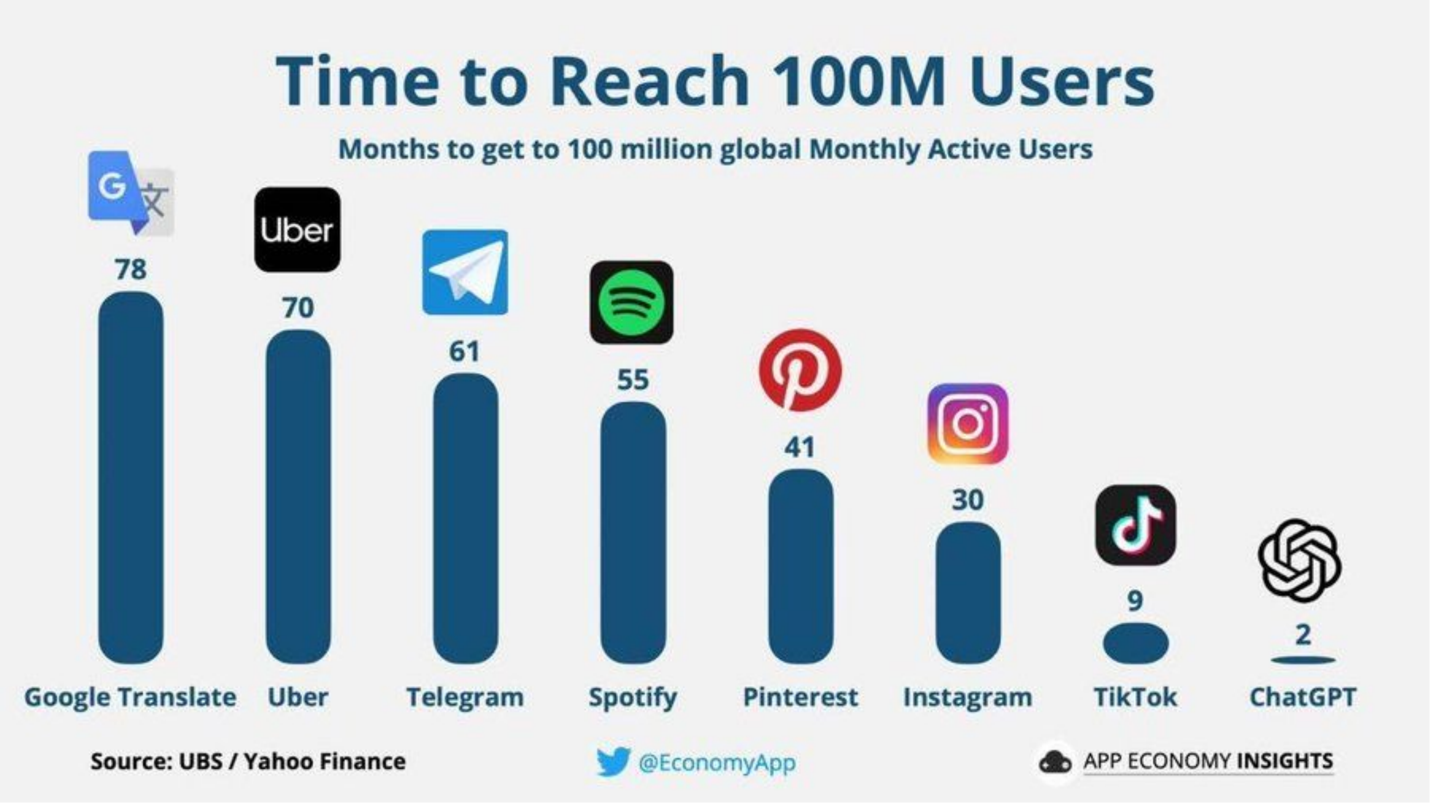
Part 21

LLM usage techniques

Fine-tuning	Prompting
Fine-tuning Gradient descent: optimize the performance on one task	**Prompting** Design special query to cue the model into specific mode to answer
Change the model itself	Change the way to use it
Update some parts of the model	No parameters change, only refining the input
Become specialist	Stay generalist

Part 2

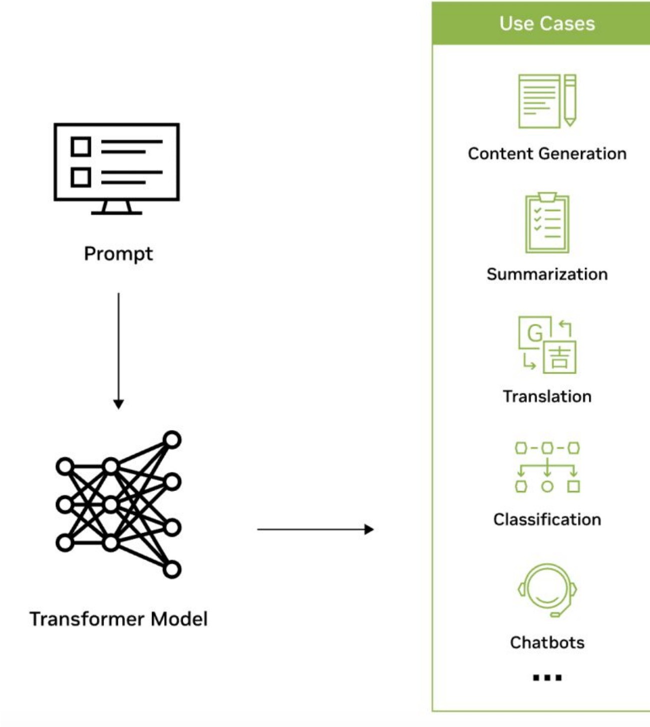
Using chatGPT



Part 2

Example Using chatGPT by API

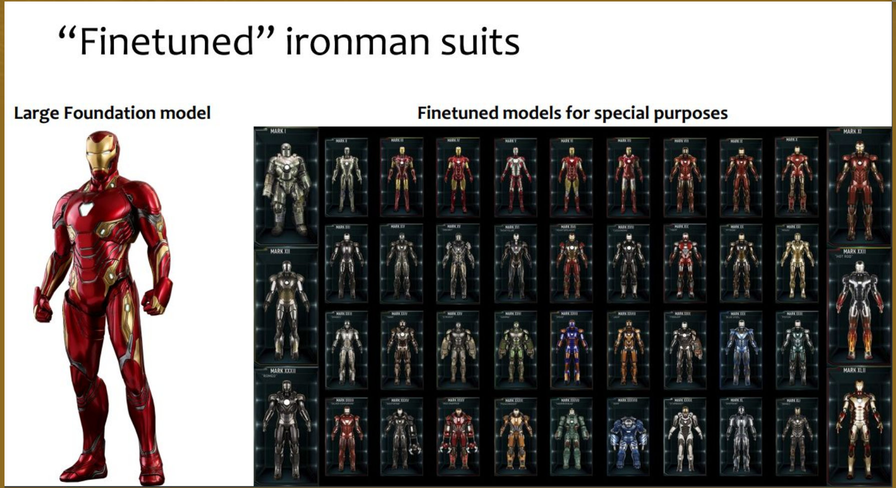
- The ChatGPT API allows for easy integration of chatbots into workflows for both businesses and individuals
- without requiring deep tech skills or significant resources.
- It supports the creation of virtual assistants and personal tutors, among others.
- OpenAI's documentation offers detailed guidance on using the API effectively.



project

LLM usage techniques

LLM Fine-tuning



Part 2

What is Fine tuning ?

taking a pre-trained language model (GPT-3) and adjusting at least one internal parameter for a specific Use case

GPT 3

ChatGPT 3.5, fine tuned .

Fine-tuning transforms a base language model, like GPT-3, into a specialized tool tailored for specific tasks, enhancing its accuracy and practicality.

Part 2

What is Fine tuning ?

A smaller fine-tuned model can outperform a larger base model

GPT 3. 175 b.

InsructGPT 1.3b

Part 2

Fine-tuning

- **Self-Supervised Learning:**
  - Train the model using a raw & huge dataset,
  - **Process**: Predict the next word given a sequence of words, allowing customization for specific tasks.
-

Part 21

Fine-tuning

- **Self-Supervised Learning:**
    - Train the model using a raw & huge dataset,
    - **Process**: Predict the next word given a sequence of words, allowing customization for specific tasks.
- **Supervised Learning:**
    - Use question-answer pairs or other labeled data to train the model for improved task-specific responses.
    - Method: Translate input-output pairs into prompts, enabling the model to learn how to answer questions effectively.

Part 2

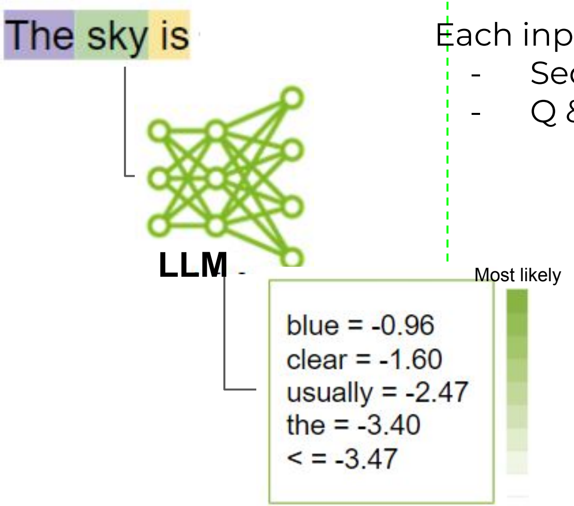
Fine-tuning

- **Self-Supervised Learning:**
  - Train the model using a raw & huge dataset,
  - **Process**: Predict the next word given a sequence of words, allowing customization for specific tasks.
- **Supervised Learning:**
  - Use question-answer pairs or other labeled data to train the model for improved task-specific responses.
  - Method: Translate input-output pairs into prompts, enabling the model to learn how to answer questions effectively.
- **Reinforcement Learning:**
  - Train the model using rewards for generating desired completions.
  - Steps: Supervised fine-tuning, creating a reward model, and reinforcement learning through feedback loops.

Part 1

Fine-tuning-PEFT

Unlabeled corpus
Huge amount row data

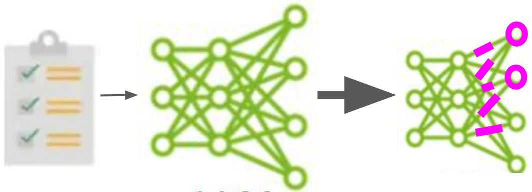


1. Self-supervised

labeled corpus

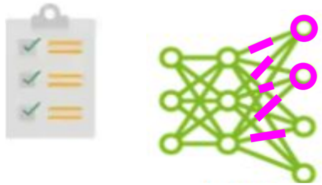
Each input is related to output
- Seq classification (sentiment analysis)
- Q & A pairs

Template: "Generate a [**specific output**] based on the following input: [**Input text**]."

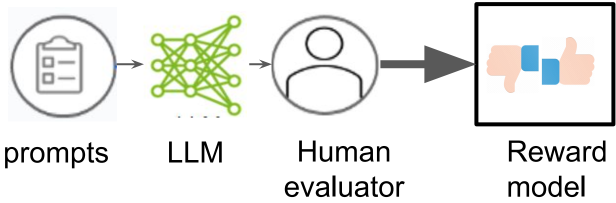


2. Supervised

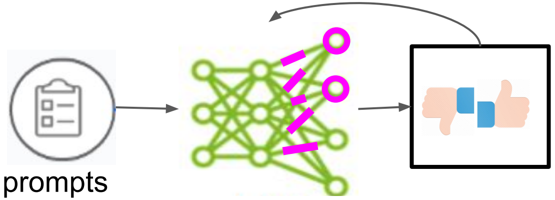
1) Supervised FT



2) Train reward model



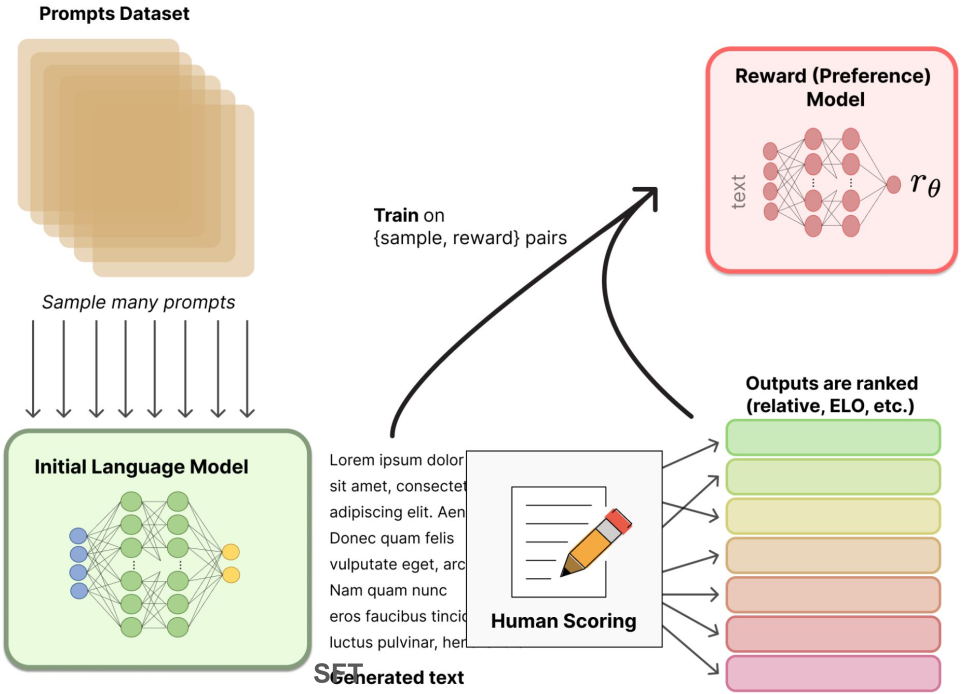
3) RL with PPO



3. Reinforcement learning from human feedback RLHF

Part 2

Fine-tuning with RL



Next word prediction, doesn't guarantee it willll align with human values helpful, truthful, or harmless. To address this, "human alignment" is to ensure LLMs act according to what humans expect.

Supervised Fine-tuning

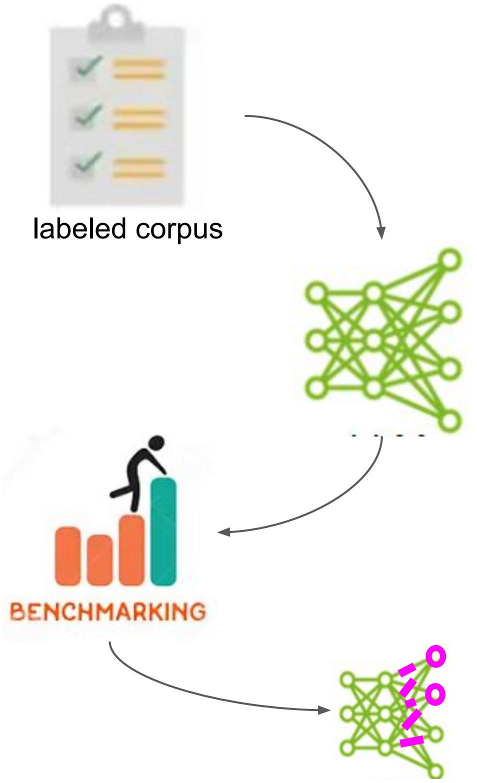
1. **Choose Your Fine-Tuning Task:** Define the specific task you want the model to perform: text summarization, classification.

2. **Prepare Your Training Dataset**: Curate a dataset containing input-output pairs (specific task).i.e training for summarization, pair original texts with their corresponding summaries.

3. **Choose Your Base Model:** Select a pre-trained LLM as your starting point. It'll be fine-tuned for your specific task.

4. **Fine-Tune the Model via Supervised Learning**: Use the curated dataset to train the selected base model.

5. **Evaluate Model Performance:** Assess the fine-tuned model's performance on relevant metrics to ensure it meets the desired

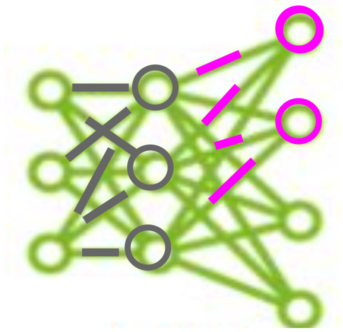


Part 2

Options for parameters tuning

1. **Retrain all parameters**

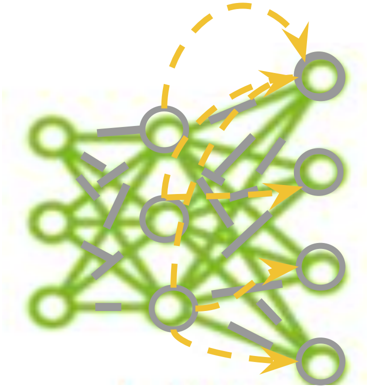
2. **Transfer learning**



3. **Parameter Efficient Fine-tuning (PEFT)**

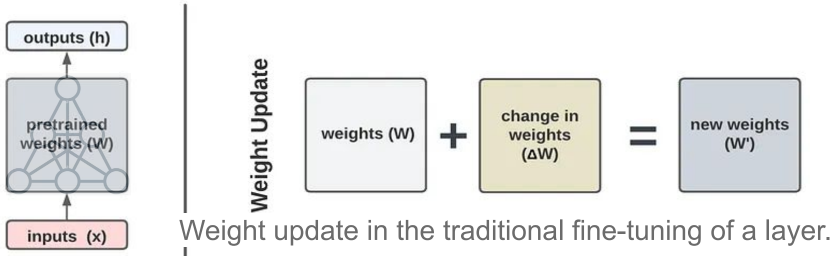
    Low Rank Adaptation (LoRA)

    A smaller set of new parameters is added to the model

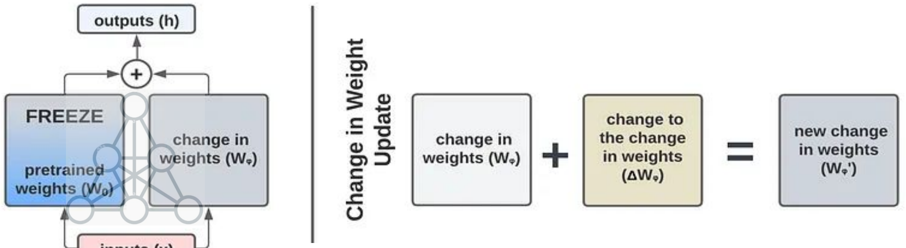


Part 2

Parameters Efficient Fine-tuning-LoRA

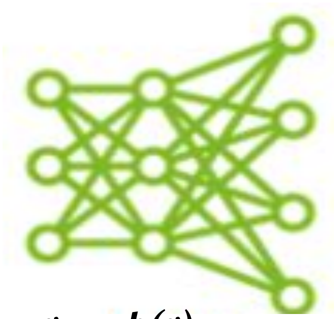


Weight update in the traditional fine-tuning of a layer.

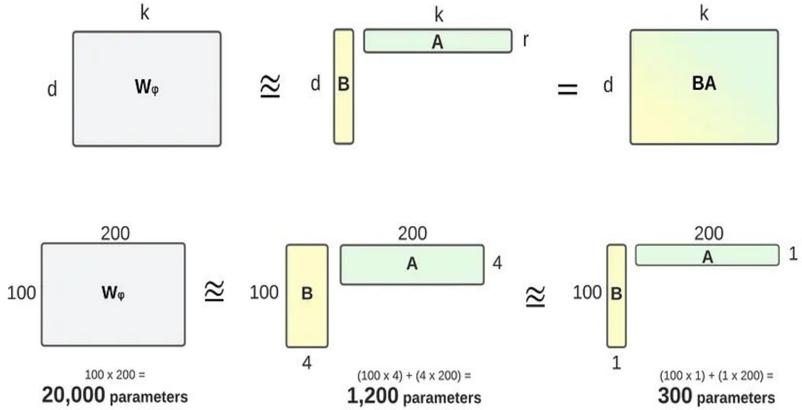


Reformulated forward pass and added weights

$$h(x) = W_0x + W_\varphi x = W_0x + BAx$$



$x \rightarrow h(x)$



Reminder: SVD

LORA : it's a low rank approximation of the weight matrix determined by picking the n largest eigenvalues

Part 2

Fine-tuning

Project 1

Project 2

Part 2

What is Hugging face

🤗

- ML & data platform, community helps AI users build , deploy and train model
- Provide infrastructure to run and demo
  - Get started
- LLM leaderboard: to track, rank and evaluate open LLMs and chatbots

Part 2

Fine-tuning: Project to Evaluate

Here you have an explained project about DOP fine tuning (Llama on stack exchange preference dataset). Which will be your base to for further.

- Generalize this process for any model and given dataset.
- Develop a **Gradio** interface for users to input fine-tuning parameters.

Different datasets for fine tuning

Generalizing the Fine-Tuning Process:

- Adapt fine-tuning for different models and datasets.
- Automate preprocessing and fine-tuning steps.
- Adapt your code to be able accept different datasets

Validation:

- Validate the platform by demonstrating its application across different models and datasets, showcasing its flexibility.

Deliverables:

- A Gradio interface that enables flexible model and dataset fine-tuning.
- A demonstration of the platform's application on various use cases.
- Code: a notebook explaining all steps

Part 2

LLM usage techniques

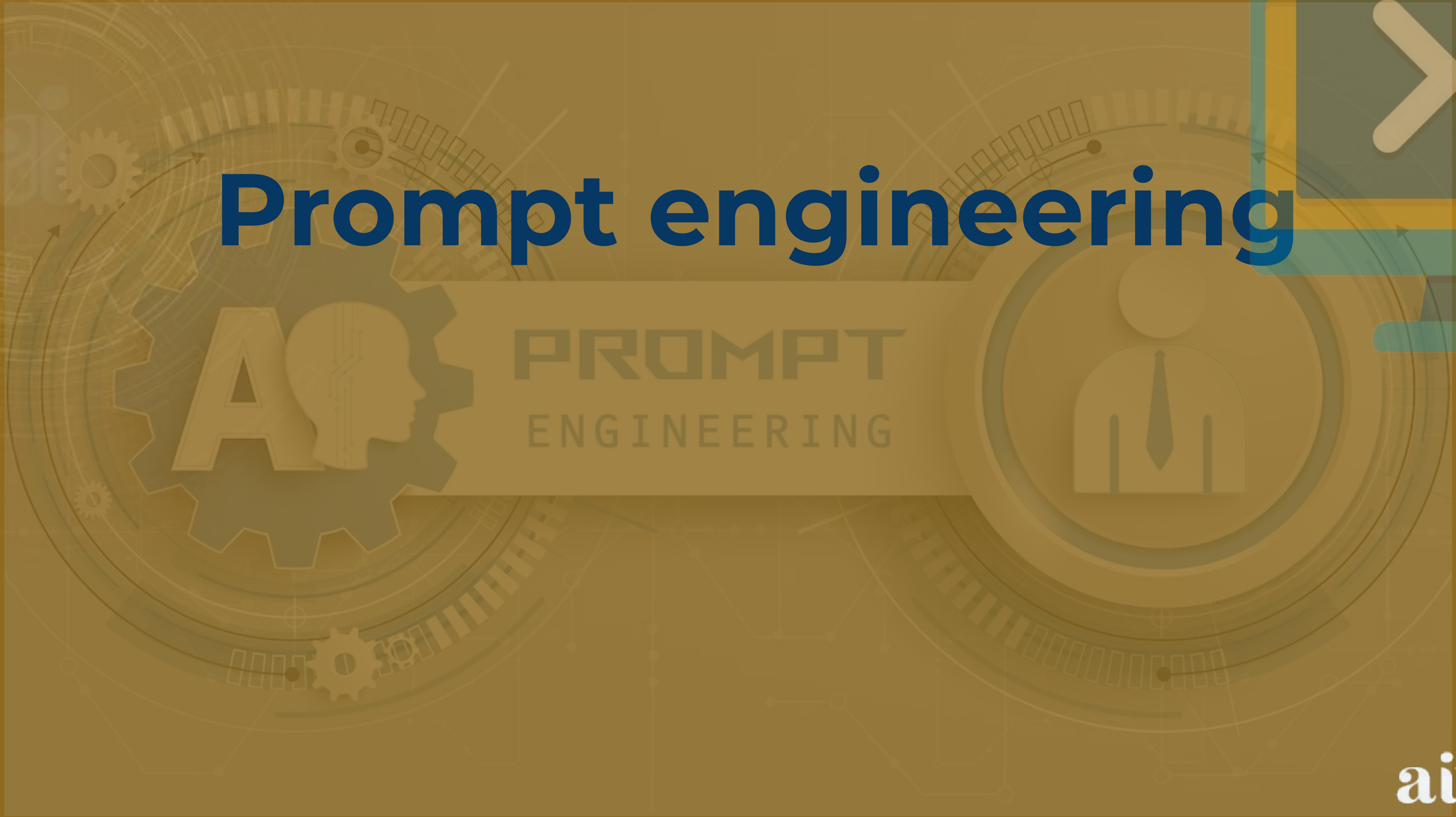
Fine-tuning
Gradient descent: optimize the performance on one task

Prompting
Design special query to cue the model into specific mode to answer

Fine-tuning	Prompting
Change the model itself	Change the way to use it
Update some parts of the model	No parameters change, only refining the input
Become specialist	Stay generalist

LLM usage techniques

Prompt engineering



aivancity
PARIS-CACHAN

Part 2

Prompting LLM

Zero shot approach: a direct question (good for analysis sentiment)

i.e.: I was on hold for 30 minutes, their customer support service is a nightmare.
Sentiment in one word:

The lukewarm food arrived long after overzealous waiters convinced us to skip appetizers
Sentiment in on word:
Write a neutral review:

i.e.: i.e.: I was on hold for 30 minutes, their customer support service is a nightmare.
Sentiment in one word:
Category:

Few shot approach: a model is presented with a small set of high-quality examples consisting of an input and desired output

- *Analyze the following sentence: "Deforestation rates in the Amazon rainforest have reached a 10-year high."*
- *Briefly summarize the environmental impact of deforestation.*
- *Write a short social media post raising awareness about the dangers of deforestation. (Bonus: Include a call to action)*

Text: (lawrence bounces) all over the stage, dancing, running, sweating, mopping his face and generally displaying the wacky talent that brought him fame in the first place.
Sentiment: positive
Text: despite all evidence to the contrary, this clunker has somehow managed to pose as an actual feature movie, the kind that charges full admission and gets hyped on tv and purports to amuse small children and ostensible adults.
Sentiment: negative
Text: for the first time in years, de niro digs deep emotionally, perhaps because he's been stirred by the powerful work of his co-stars.
Sentiment: positive
Text: i'll bet the video game is a lot more fun than the film.
***Sentiment**: ?*

Part 2

Prompting LLM

Involves crafting better user inputs, for a precise generated output

Write an official letter $\Rightarrow$ specify length, style, the main points of the topic

Art of guiding generative model to answer/generate on specific task

User provides an article $\Rightarrow$ Summarize, extract argument, answering form it

Split complex tasks into simpler subtasks

Solve math problem $\Rightarrow$ split into subtasks

Tutorial1, Tutorial 2, *Studies looked into how to construct in-context examples to maximize the performance and observed that choice of prompt format, training examples, and the order of the examples can lead to dramatically different performance, from near random guess to near SoTA*. — *link*

Part 2

Prompting LLM: project

Test LLM's ability to label data

Here you find two codes that can be considered a good bases for a project that you can develop. Take one of them, understand the flow, and adapt it to classify your own dataset (annotation)

You may consider this dataset,

The idea is that you show your model some examples with their label, and ask it to find the labels of the remaining data

Part 2

Prompting LLM
disadvantages

What is the solution ?

Iterative process

User has to experiment with different prompts for the desired output
Time and resource intensive process

Some expertise

Refining prompts requires some skills

LLM models has limited tokens size, and costs money

Providing context can by expensive/and limited

Part 2

Prompting LLM
Evaluation

ROUGE or BLEU is limited to specific tasks. They fail if understanding of nuance, style, cultural context, or idiomatic expressions is required

Task-specific metrics:

Summarization: ROUGE (recall-oriented Understudy for Gisting Evaluation)

- ROUGE-1 Recall r:the number of unigram matches divided by the total number of unigrams in the reference text
- ROUGE-1 Precision P: the number of unigram matches divided by the total unigrams in the model's summary
- F1score 2*r*p/r+p

calculate

Translation: BLEU (Bilingual Evaluation Understudy): quantifies how close the machine translation is to a set of high-quality human translations

- For unigrams, we check what percentage of predicted words are present in the golden translation.
- ```
  machine = "Szybki lis skacze nad leniwym psem".split(" ")
  golden = "Szybki brązowy lis przeskakuje nad leniwym psem".split(" ")
  ```

```
Unigram: 5/6, bigram: 2/5, 3-gram : ?
Then calculate geometric mean of there three n-grams:geom avg precision = (0.83 * 0.4 * 0.25)**(1/3)
Then calculate Brevity penalty (if translation is shorter than the golden
```

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$
```

Part 2

Prompting LLM

Evaluation

- Research benchmarks:
  - Massive Multitask Language Understanding, GSM8k,
- LLM-self Evaluation :
  - You query your model and get a response back. Then, you feed the query and the response to another LLM, with a crafted prompt if the response is in the context of query

Good to read

Appendix

Type of attentions, how it works

SVD

- The Singular Value Decomposition (SVD) provides a way to factorize a matrix, into singular vectors and singular values
- much like how an integer is broken down into its prime factors for deeper understanding. This decomposition allows us to dissect a matrix into its fundamental components (pertinent features!),
    - **i.e 60 (integer), factorize 60 into its prime factors**
- SVD of any matrix A is given by: $A = UDV^T$, The matrix U and V are orthogonal matrices, D is a diagonal matrix (not necessarily square).
- Elements along diagonal D are known as Singular values. The columns of U are known as the left-singular vectors. The columns of V are known as right-singular vectors.
- Read this
- Or prompt it ;)