Pre-training, Prompting or Fine-tuning LLMs

Nicolas.Hernandez@univ-nantes.fr

M2 ATAL - 15 janvier 2025

Nantes Université

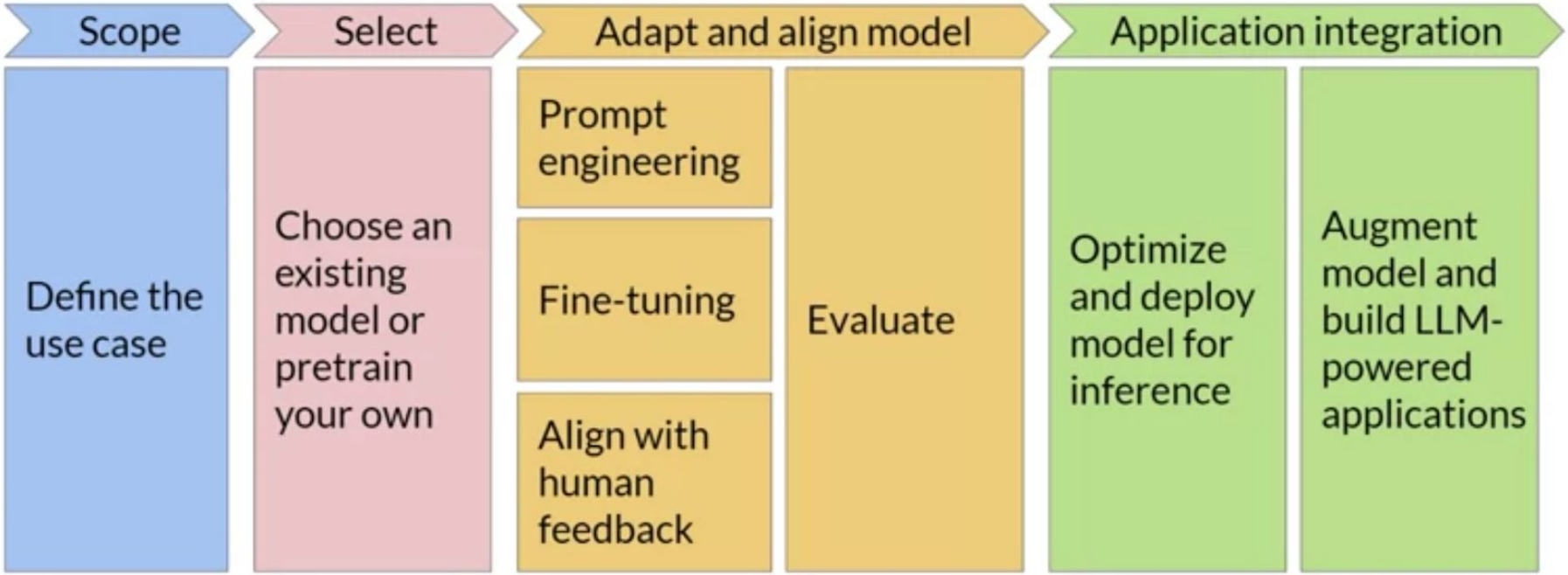
Laboratoire des Sciences du Numérique de Nantes

TALN

Avertissement

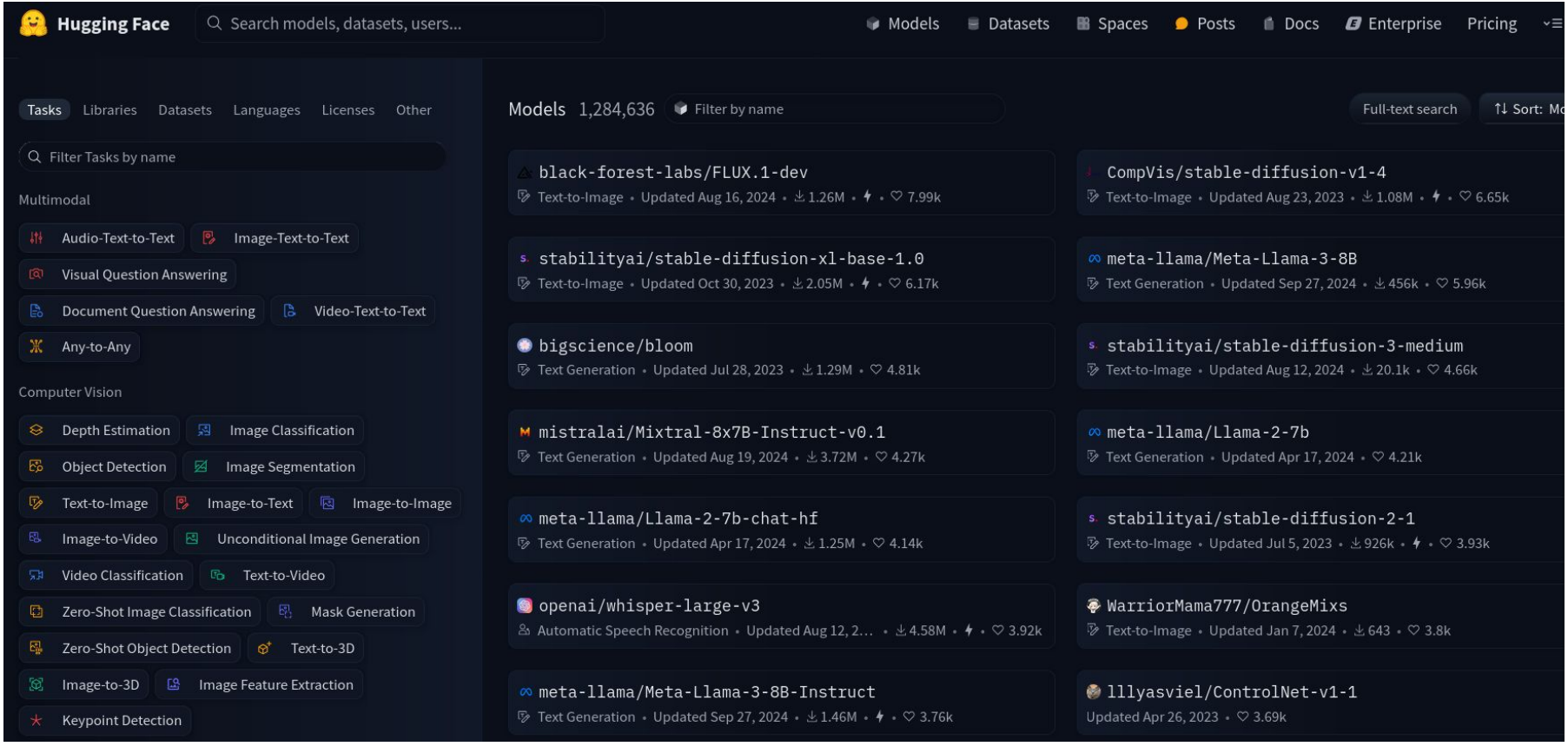
Ce support a été construit sur la base d'un matériel pédagogique mis à disposition dans le coursera: Generative AI with Large Language Models.

Generative AI project lifecycle

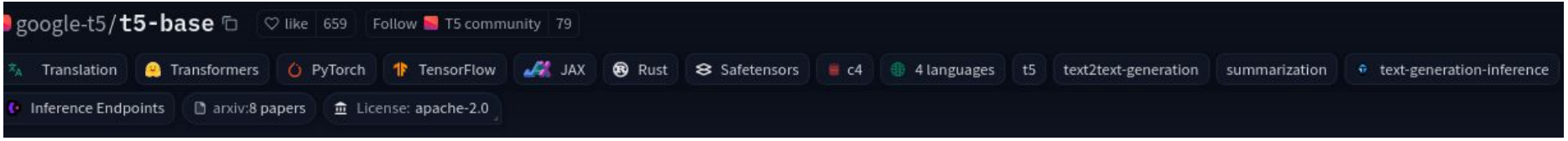


Choosing an existing foundation model

E.g. hugging face hub



Model card and more e.g. for T5 base



Model Card for T5 Base

model image

Table of Contents

1. Model Details
2. Uses
3. Bias, Risks, and Limitations
4. Training Details
5. Evaluation
6. Environmental Impact
7. Citation
8. Model Card Authors
9. How To Get Started With the Model

Dataset used to train `google-t5/t5-base`

`legacy-datasets/c4`

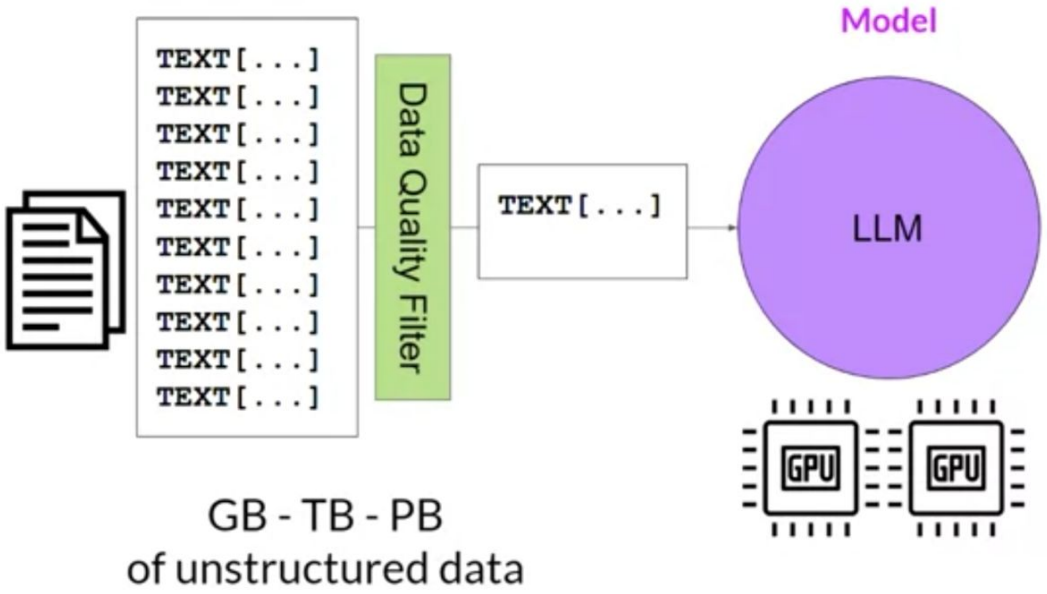
Updated Mar 5, 2024 • ↓ 17.4k • ♡ 240

Model tree for `google-t5/t5-base`

Adapters	38 models
Finetunes	435 models
Merges	1 model
Quantizations	6 models

Pretraining a custom LLM

LLM pre-training at a high level



GB - TB - PB
of unstructured data

Token String	Token ID	Embedding / Vector Representation
'_The'	37	[-0.0513, -0.0584, 0.0230, ...]
'_teacher'	3145	[-0.0335, 0.0167, 0.0484, ...]
'_teaches'	11749	[-0.0151, -0.0516, 0.0309, ...]
'_the'	8	[-0.0498, -0.0428, 0.0275, ...]
'_student'	1236	[-0.0460, 0.0031, 0.0545, ...]
...	...	...

Vocabulary

LLM pre-training at a high level

Requires vast amount of data (giga to petabytes of text), of GPUs, and compute

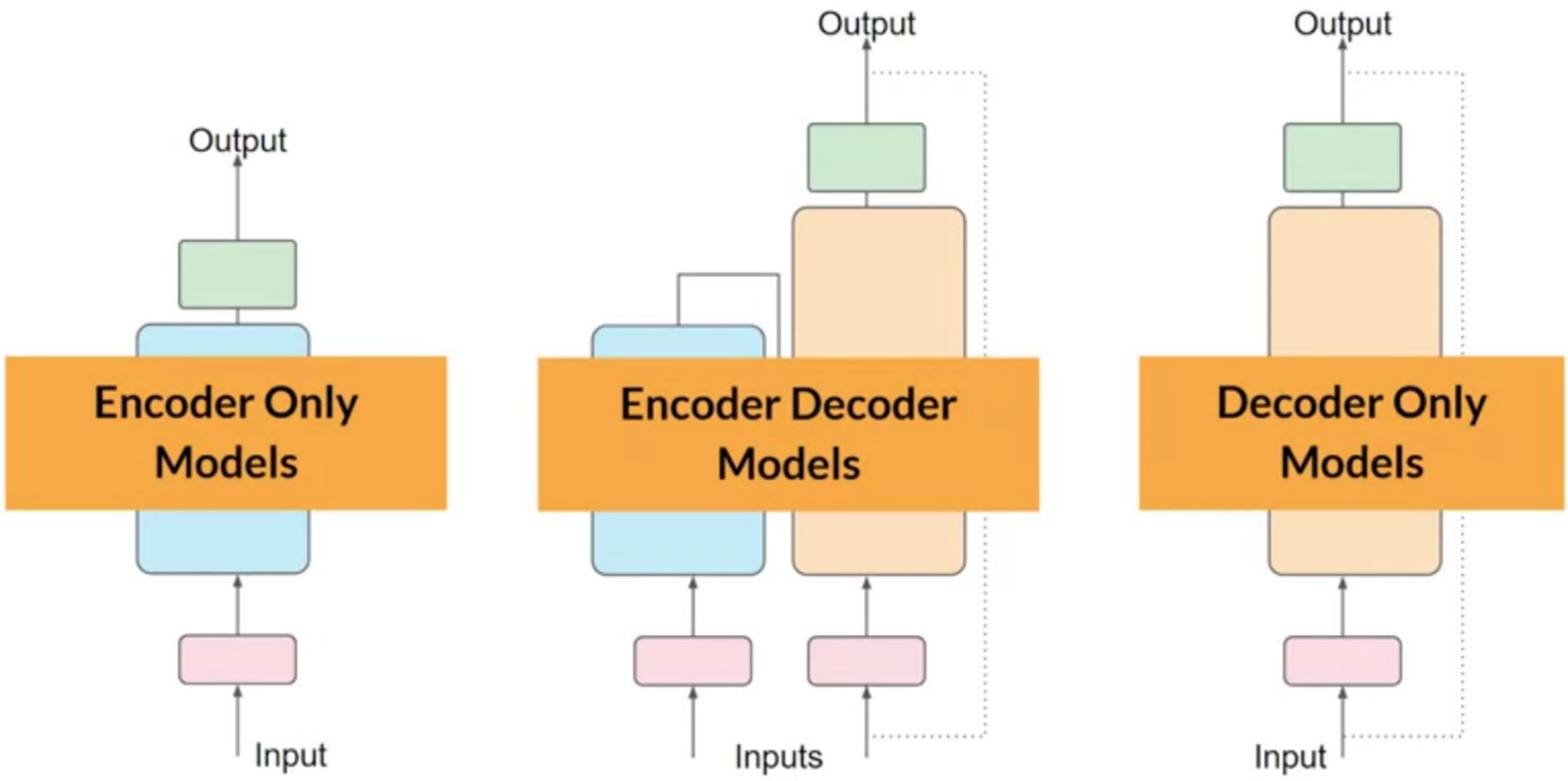
Many sources (web scraping, texts corpora)

training objective depends on the model architecture

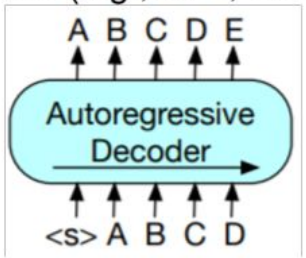
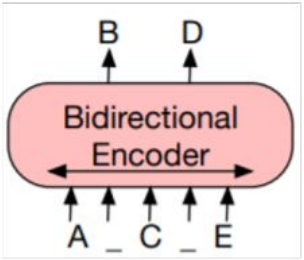
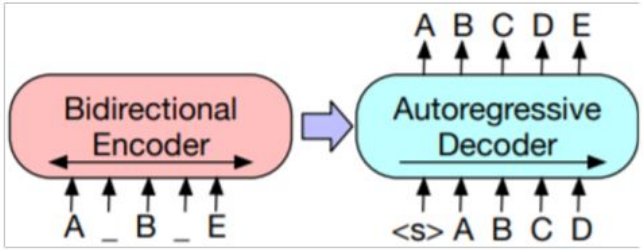
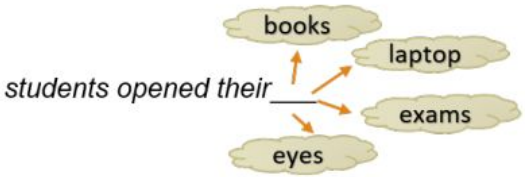
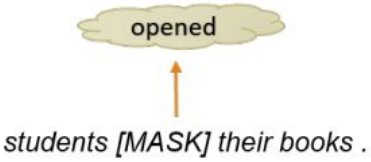
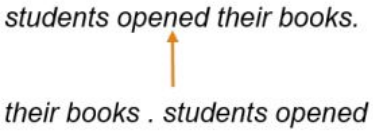
Updates the model weights to minimize the loss of the training objective

When scraping from (web) public sites, data quality curation required to address bias and remove other harmful content: 1-3% of tokens used

Transformer-based architectures



3 types of pre-training

	Encoder	Decoder	Encoder-Decoder
	Autoregressive language model (e.g., GPT, GPT-2/3)	**Masked** language model (e.g., BERT, RoBERTa, XLM-R)	**Encoder-Decoder** (e.g., BART, T5)
Model & illustration			
Training objective	Predicting what word comes next given previous words	Predicting masked words given other words in the sequence	Corrupting a sequence and then predicting the original sequence
Example			

(Min et al, 2021;Lewis et al., 2020)

3 types of pre-training

Encoder: **Autoencoding/Masked Language** models

- Objective: reconstruct text (“denoising”) by predicting masked words
- Good for token or sentence classification tasks
- E.g. ALBERT, BERT, DistilBERT, ELECTRA, RoBERTa

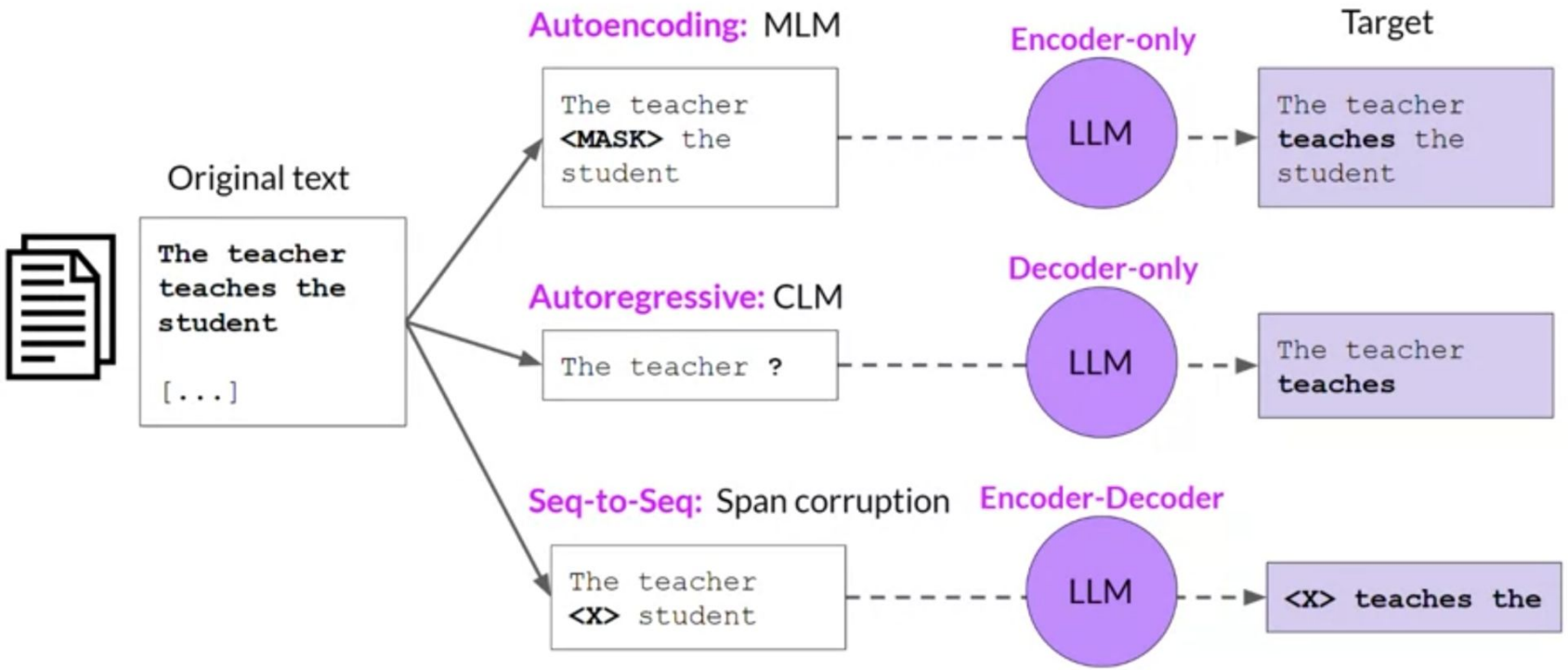
Decoder: **Autoregressive** models/**Causal** language models

- Objective: Predict the next word (iteratively, for each word, « masks » the following words to force the attention head to pay attention to the preceding context)
- Good use cases: generation tasks
- E.g. CTRL, GPT, GPT-2, Transformer XL

Encoder-Decoder: **Sequence-to-sequence** models

- Objective: predict a target sequence given a corrupted source one (word permutation, missing words, masked words, span corruption…)
- Good use cases: translation, summarization, question/answering
- E.g. BART, mBART, Marian, T5

Model architectures and pre-training objectives



Pre-training LLaMA 1

Trained on **1T words** (1,000 milliards) during **21 days** on **2,048 GPUs**

	GPU Type	GPU Power consumption	GPU-hours	Total power consumption	Carbon emitted (tCO_2eq)
OPT-175B	A100-80GB	400W	809,472	356 MWh	137
BLOOM-175B	A100-80GB	400W	1,082,880	475 MWh	183
LLaMA-7B	A100-80GB	400W	82,432	36 MWh	14
LLaMA-13B	A100-80GB	400W	135,168	59 MWh	23
LLaMA-33B	A100-80GB	400W	530,432	233 MWh	90
LLaMA-65B	A100-80GB	400W	1,022,362	449 MWh	173

1 tCO2eq (or 1,000kg CO2eq) corresponds to emissions of (source: educlimat):

1 lamp lit for 54 years ⇔ **1 house heated with gas for 1 year** ⇔ 12 km / day by car for a year ⇔ 2 beef dishes per week for 1 year

…LLaMA2, released 5 months later, was trained on **2T words**

Source: [LLaMA: Open and Efficient Foundation Language Models](), (Touvron et al.@Meta, 27 Feb, 2023)

Pre-training for domain adaptation

Specialized domains

New words, uncommon words, common words with a different meaning, idiosyncrasy…

Legal language

The prosecutor had difficulty proving mens rea, as the defendant seemed unaware that his actions were illegal.

The judge dismissed the case, citing the principle of res judicata as the issue had already been decided in a previous trial.

Despite the signed agreement, the contract was invalid as there was no consideration exchanged between the parties.

Medical language

After a strenuous workout, the patient experienced severe myalgia that lasted for several days.

After the biopsy, the doctor confirmed that the tumor was malignant and recommended immediate treatment.

Sig: 1 tab po qid pc & hs

Take one tablet by mouth four times a day, after meals, and at bedtime.

Pre-training from scratch result in better models for highly specialized domains (legal, medical, finance, science)

BloombergGPT: A Large Language Model for Finance (Wu et al. at Bloomberg, 2023)

Best model on financial tasks without sacrificing perf on general LLM benchmarks thanks to a mixed of specialized ~51% and general ~49% data

Follow roughly the Chinchilla approach for a fixed compute budget of 1.3 M GPU hours (~230 M petaflops): optimal for 50 Billions parameters and 1.4 Trillions tokens

Actually: a bit more param and less tokens than the recommendation

Challenging to collect financial data: 700 B tokens were collected : less than compute-optimal value

And due to early stopping, the training process terminated after processing 569 B

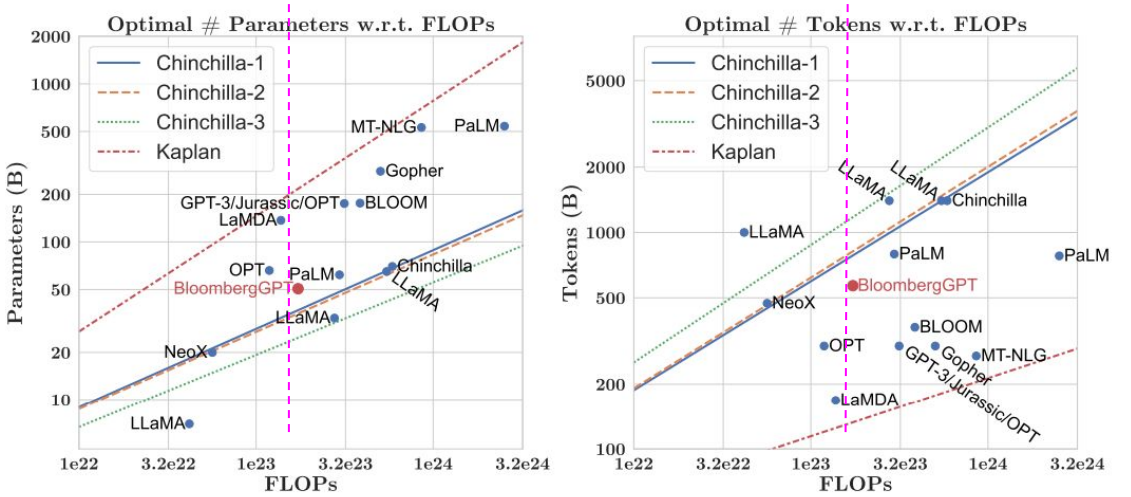


Figure 1: Kaplan et al. (2020) and Chinchilla scaling laws with prior large language model and BLOOMBERGGPT parameter and data sizes. We adopt the style from Hoffmann et al. (2022).

Real world constraints may force you to make trade offs when pretraining your own models

Prompting, prompt engineering and in-context learning

Prompting and prompt engineering

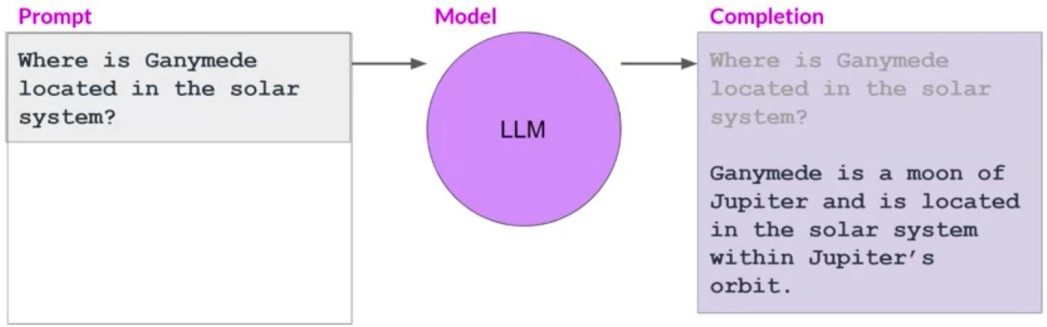
Prompt : données (e.g. texte) introduites en entrée du modèle to encourage the pre-trained model to use specific encoded information in solving specific task

Inference : acte de production du modèle (prédiction ou génération)

Completion: données produites et en sortie du modèle

Context window : la taille maximale du prompt en nombre de token

Prompt engineering : art d'écrire des prompts (en fonction du comportement du modèle)



In-context learning (ICL) : Ajout de données au prompt

Zéro-démonstration (*zero-shot*)

Texte : L'histoire du film m'a captivé.
Sentiment :

Sentiment : positif

Prompt (écrit par un humain)

Complétion (générée par la machine)

Quelques-démonstrations (*few-shot*)

Texte : Marie-Claude bondit sur toute la scène, dansant, courant, transpirant, s'essuyant le visage et faisant généralement preuve du talent unique qui lui a valu la célébrité.
Sentiment : positif

Texte : Malgré toutes les preuves du contraire, ce navet a réussi à se faire passer pour un vrai long métrage, le genre de film dont l'entrée est payante, qui fait l'objet d'un battage médiatique à la télévision et qui prétend amuser les petits enfants et les jeunes adultes.
Sentiment : négatif

Texte : L'histoire du film m'a captivé.
Sentiment :

Sentiment : positif

Language Models are Few-Shot Learners (Brown et al. @OpenAI, May 2020)

In-context learning avec une instruction, un raisonnement

Instruction

Stp répond à la question suivante : Quel est le point d'ébullition de l'azote ?

L'azote liquide a une température d'ébullition de -196 °C.

Traduire de l'anglais au français : cheese =>

Fromage

[Finetuned Language Models Are Zero-Shot Learners](#) (Wei et al. @Google, 2022)

Raisonnement (*chain-of-thought*)

Q : Roger a 5 balles de tennis. Il en achète deux tubes de balles de tennis. Chacun des tubes compte 3 balles. Combien a-t-il de balles maintenant ?

R : Rocher a démarré avec 5 balles. 2 tubes de 3 balles de tennis correspond à 6 balles de tennis. 5+6 = 11. La réponse est 11.

Q : La cafétéria avait 23 pommes. Elle en a utilisé 20 pour le déjeuner et en a acheté 6 de plus, combien de pommes reste-il ?

La cafétéria avait 23 pommes à l'origine. Elle en a utilisé 20 pour préparer le déjeuner. Elle avait donc 23 - 20 = 3. Elle a acheté 6 pommes supplémentaires, elle a donc 3+6 = 9.

[Chain-of-Thought Prompting Elicits Reasoning in Large Language Models](#) (Wei et al. @Google, NeurIPS 2022)

Inviter à raisonner

Réponds à la question suivante en raisonnant étape par étape :
La cafétéria avait 23 pommes. Elle en a utilisé 20 pour le déjeuner et en a acheté 6 de plus, combien de pommes reste-il ?

La cafétéria avait 23 pommes à l'origine. Elle en a utilisé 20 pour préparer le déjeuner. Elle avait donc 23 - 20 = 3. Elle a acheté 6 pommes supplémentaires, elle a donc 3+6 = 9.

« Let's think step by step »
[Large Language Models are Zero-Shot Reasoners](#) (Kojima et al., 2022)

In-context learning (ICL)

Ajout de données au prompt e.g. a few-shot, chain-of-thought

Pas réellement de l'apprentissage (pas d'ajustement de poids du modèle)

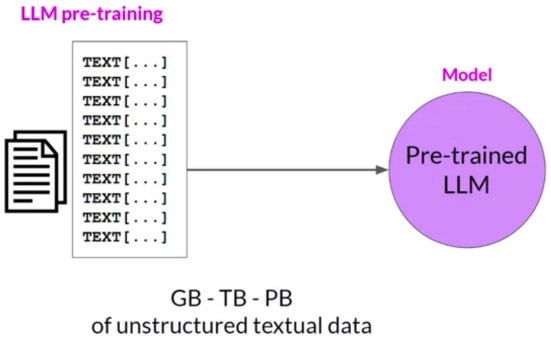
Performance et les capacités liées à la taille du modèle : ICL may not work for smaller LLM

Limitation due à la taille de la fenêtre contextuelle

Si les performances ne sont pas au rdvs alors envisager le fine-tuning de modèle

Fine-tuning an LLM

LLM fine-tuning at a high level

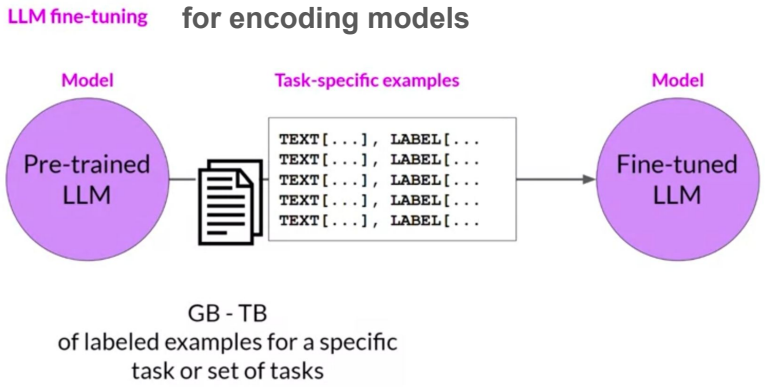


Self-supervised process

Full fine-tuning updates all parameters

Supervised process

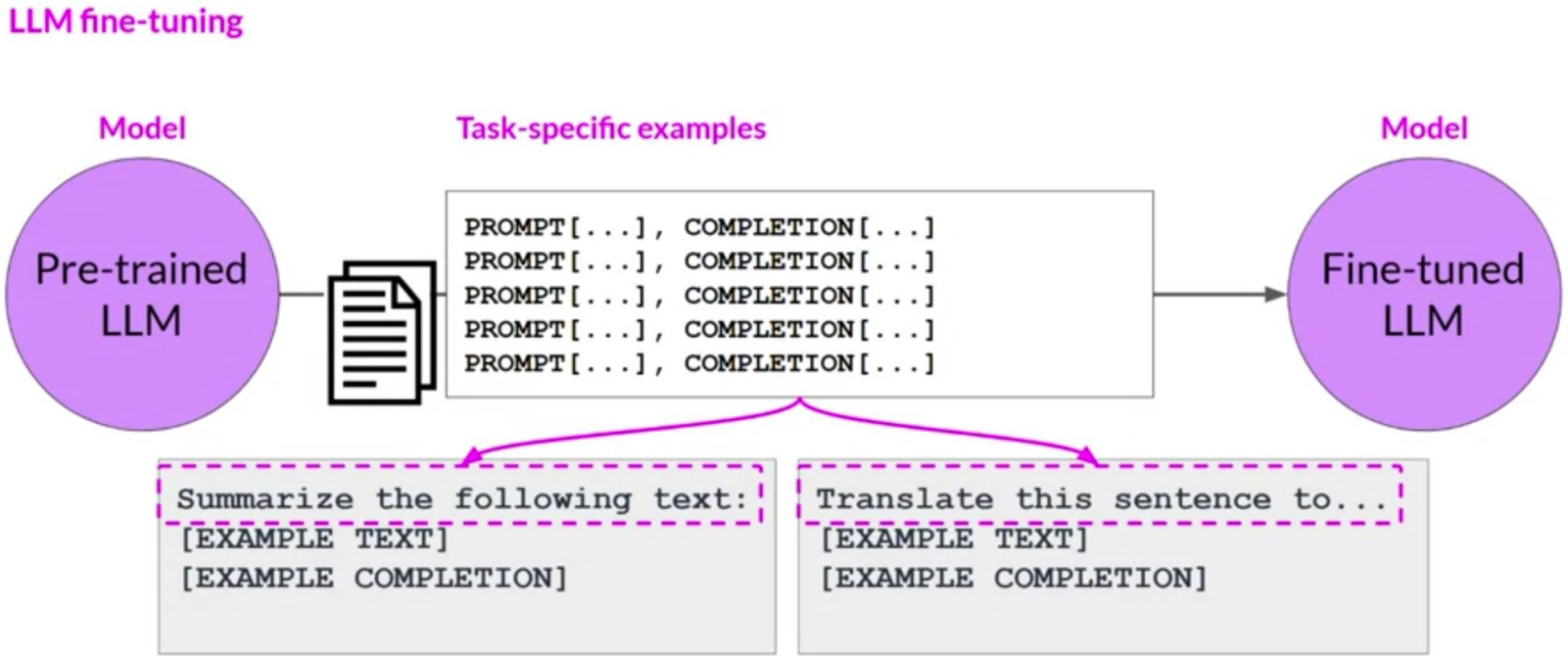
Full fine-tuning updates all parameters : requires enough memory and compute budget to store and process all the gradients, optimizers and other components that are being updated during training



Fine-tuning an LLM with instruction prompts

Instruction fine-tuning

Using prompts to fine-tune LLMs with instruction



Prompt template libraries to turn existing datasets into instruction prompt datasets for fine tuning

Many available dataset not formatted as instructions

Luckily, developers have assembled prompt template libraries that can be used to take existing datasets, and turn them into instruction prompt datasets for fine-tuning

For example, the large data set of Amazon product reviews

Prompt template libraries to turn existing datasets into instruction prompt datasets for fine tuning

Sample prompt instruction templates

Classification / sentiment analysis

```
jinja: "Given the following review:\n{{review_body}}\npredict the associated rating\
  \ from the following choices (1 being lowest and 5 being highest)\n- {{ answer_choices\
  \ | join('\\n- ') }} \n|||\n{{answer_choices[star_rating-1]}}"
```

Text generation

```
jinja: Generate a {{star_rating}}-star review (1 being lowest and 5 being highest)
  about this product {{product_title}}.  |||  {{review_body}}
```

Text summarization

```
jinja: Give a short sentence describing the following product review \n{{review_body}}\
  \ \n|||\n{{review_headline}}
```

Source: https://github.com/bigscience-workshop/promptsource/blob/main/promptsource/templates/amazon_polarity/templates.yaml

LLM fine-tuning process

LLM fine-tuning

Prepared instruction dataset

Training splits

```
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
```

Training

```
PROMPT[...], COMPLETION[...]
...
```

Validation

```
PROMPT[...], COMPLETION[...]
...
```

Test

LLM fine-tuning process

LLM fine-tuning

Prepared instruction dataset

Training splits

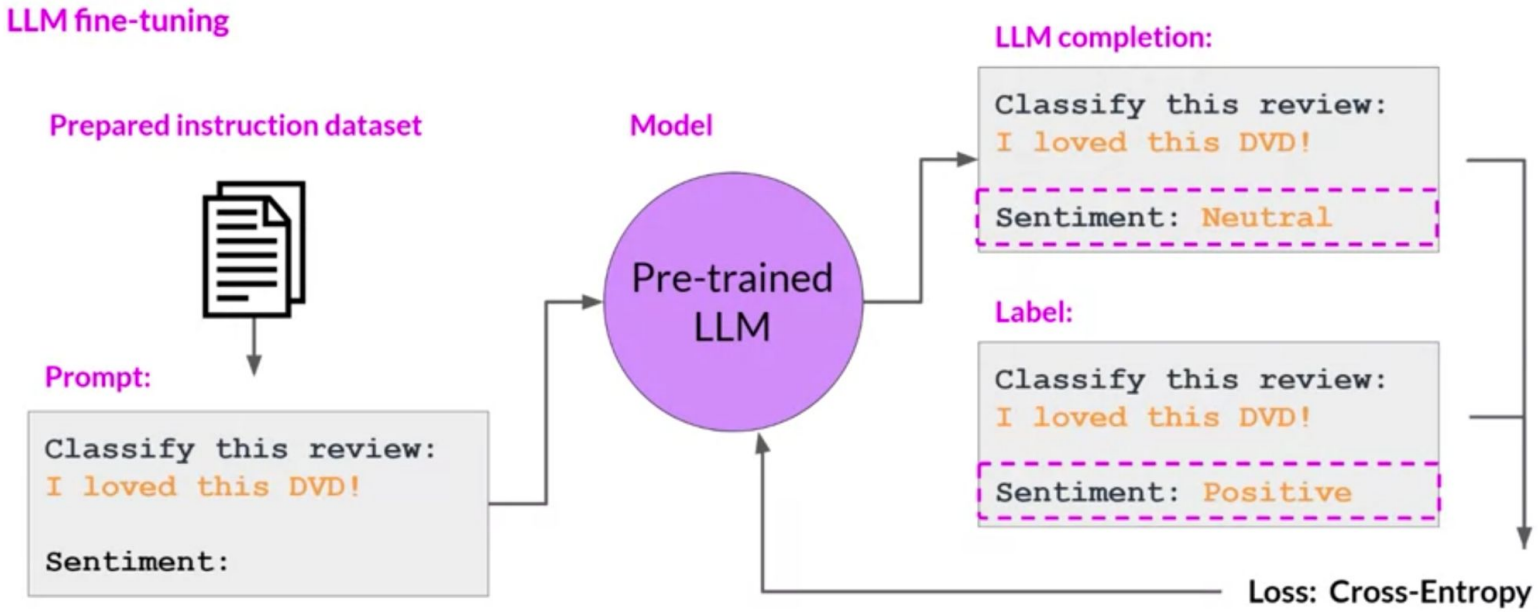
LLM fine-tuning process

LLM fine-tuning

Prepared instruction dataset

Training splits

LLM fine-tuning process

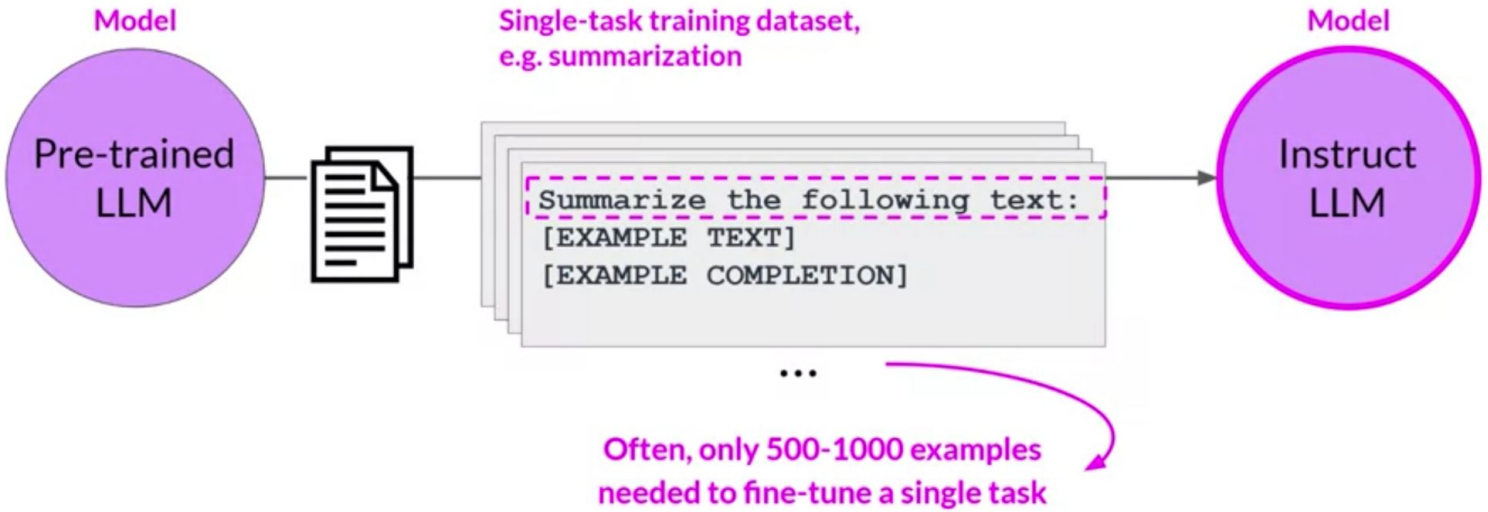


Result in a fined-tuned model also called an instructed model

Task-specific fine-tuning, general term or more for encoding model while Instruction fine-tuning more for generative models

Fine-tuning on a single task

Fine-tuning on a single task



Catastrophic forgetting and how to avoid it

Catastrophic forgetting

- Fine-tuning can significantly increase the performance of a model on a specific task
- But can lead to reduction in ability on other tasks

How to avoid

- First note that you might not have to (depending on your use case)
- If need the model to maintain its multitask generalized capabilities then perform fine-tuning on examples of multiple tasks at one time
- Consider Parameter Efficient Fine-tuning (PEFT) instead of full fine-tuning: preserve the original LLM weights and train only a few specific task adapter layers and parameters (works pretty well)

Good multitask fine-tuning may require 50-100,000 examples across many tasks, and so will require more data and compute to train.

Multi-task, Instruction Fine-tuning

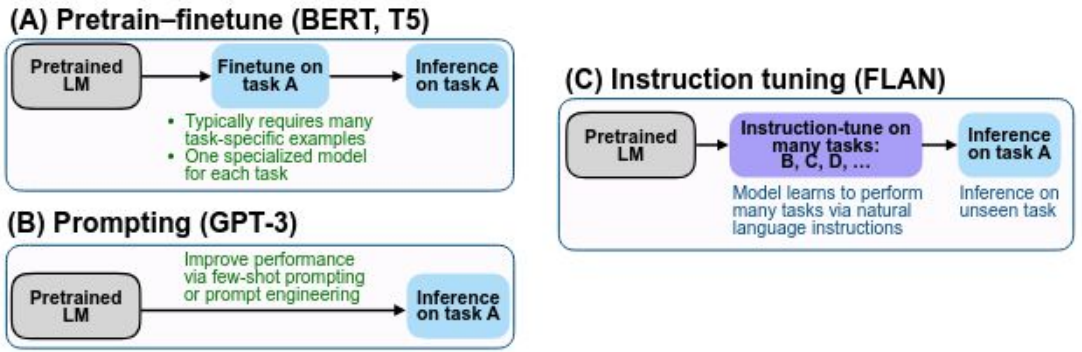
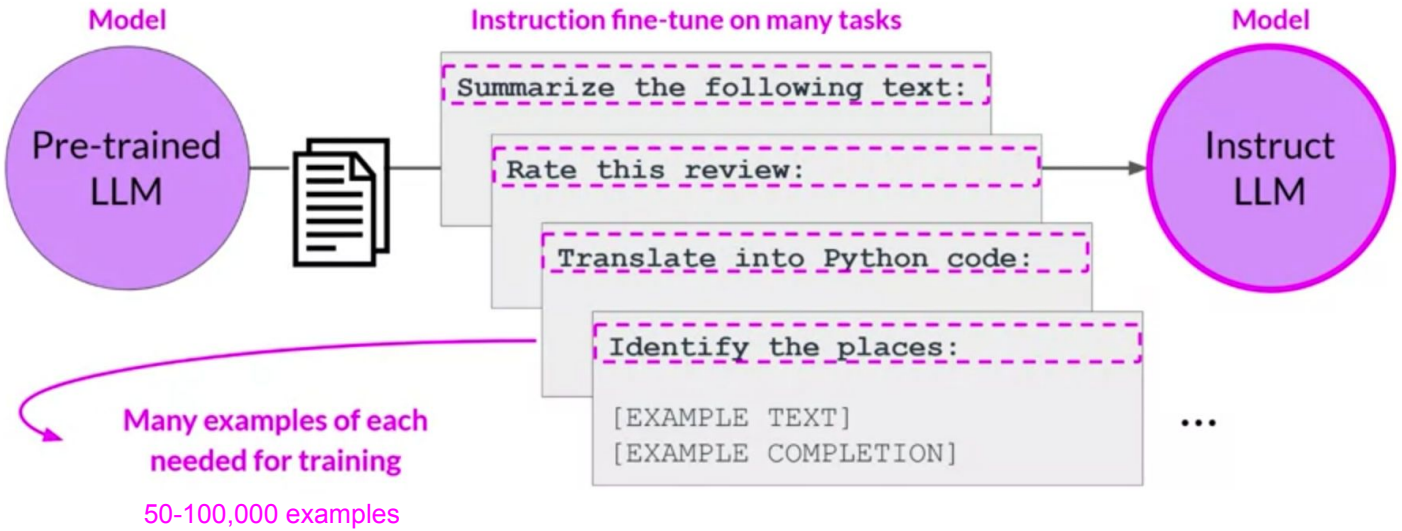


Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

Multitask, instruction fine-tuning

Multitask fine-tuning: extension of single task fine-tuning, where training dataset comprised of mixture of example inputs and outputs for multiple tasks



Instruction fine-tuning with FLAN

Instruction tuning initially proposed in Finetuned Language Models Are Zero-Shot Learners (Wei et al @Google Fev 2022)

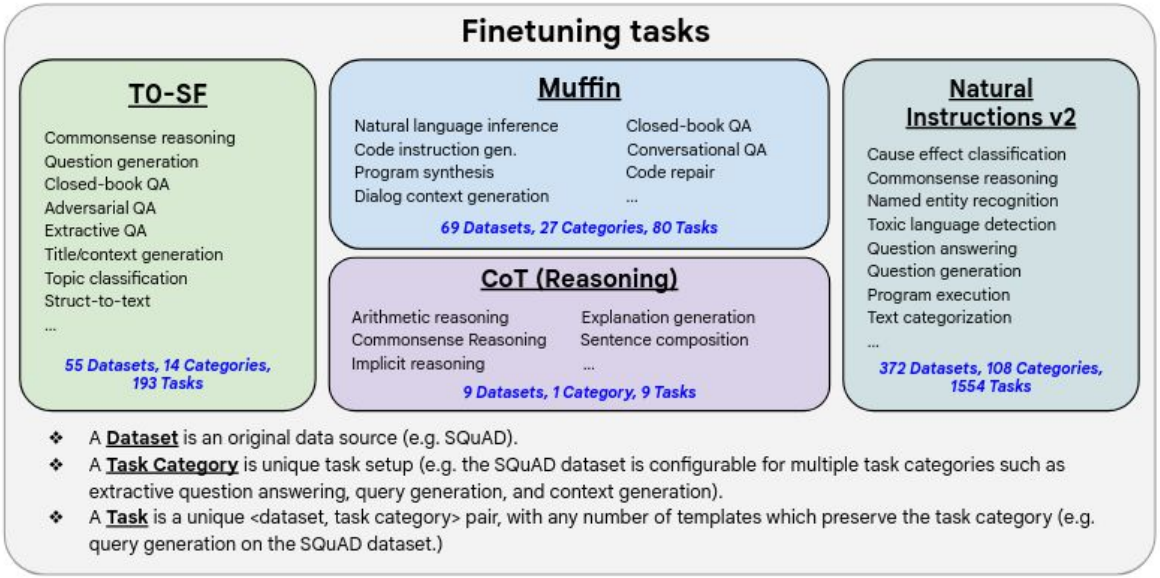
- To improve the zero-shot learning
- Main idea: Verbalize NLP tasks via natural language instruction templates
- Resulting models are called Fine-tuned LAnguage Net (FLAN) models ; as the "dessert" because FLAN fine-tuning is the last step of the training process after the "main course" of pre-training
- Improves model performance and generalization to unseen tasks

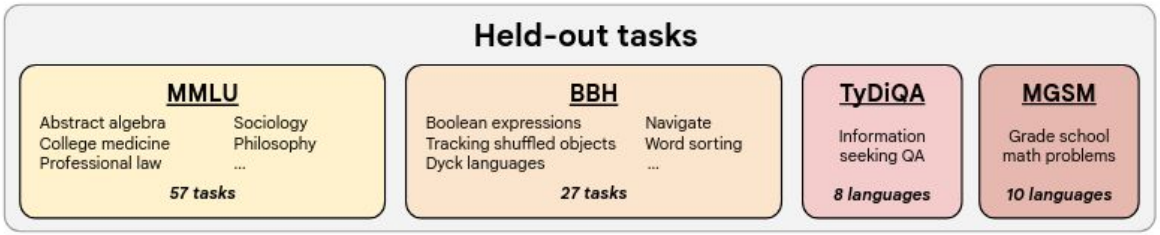
Scaling Instruction-Finetuned Language Models (Chung et al. @Google, Oct 2022)

- Explored aspects: 1) scaling number of tasks, 2) scaling model size, and (3) fine-tuning on chain-of-thought data
- Show the performance improvement on a variety of model classes (PaLM, T5, U-PaLM), prompting setups (zero-shot, few-shot, CoT), and evaluation benchmarks (MMLU, BBH, TyDiQA, MGSM, open-ended generation)

The Flan Collection: Designing Data and Methods for Effective Instruction Tuning (Longpre et al. @Google 2023) used to produce Flan-T5 and Flan-PaLM ; see the repository

FLAN-T5: Fined-tuned version of pre-trained T5 model





a great, general purpose, instruct model fined tuned on 1836 tasks

SAMSum: A dialogue dataset

Included in Muffin datasets, collection of 16,000 messenger conversations with summaries crafted by linguists

Datasets: **samsum** | Tasks: Summarization | Languages: English

dialogue (string)	summary (string)
"Amanda: I baked cookies. Do you want some? Jerry: Sure! Amanda: I'll bring you tomorrow :-)"	"Amanda baked cookies and will bring Jerry some tomorrow."
"Olivia: Who are you voting for in this election? Oliver: Liberals as always. Olivia: Me too!! Oliver: Great"	"Olivia and Olivier are voting for liberals in this election. "
"Tim: Hi, what's up? Kim: Bad mood tbh, I was going to do lots of stuff but ended up procrastinating Tim: What did…	"Kim may try the pomodoro technique recommended by Tim to get more stuff done."

Sample FLAN-T5 prompt templates

```
"samsum": [
        ("{dialogue}\n\Briefly summarize that dialogue.", "{summary}"),
        ("Here is a dialogue:\n{dialogue}\n\nWrite a short summary!",
         "{summary}"),
        ("Dialogue:\n{dialogue}\n\nWhat is a summary of this dialogue?",
         "{summary}"),
        ("{dialogue}\n\nWhat was that dialogue about, in two sentences or less?",
         "{summary}"),
        ("Here is a dialogue:\n{dialogue}\n\nWhat were they talking about?",
         "{summary}"),
        ("Dialogue:\n{dialogue}\nWhat were the main points in that "
         "conversation?", "{summary}"),
        ("Dialogue:\n{dialogue}\nWhat was going on in that conversation?",
         "{summary}"),
]
```

Source: https://github.com/google-research/FLAN/blob/main/flan/templates.py

Improving FLAN-T5 capabilities

to domain specific or proprietary private data

Will be explored in a laboratory session with the dialogsum dataset

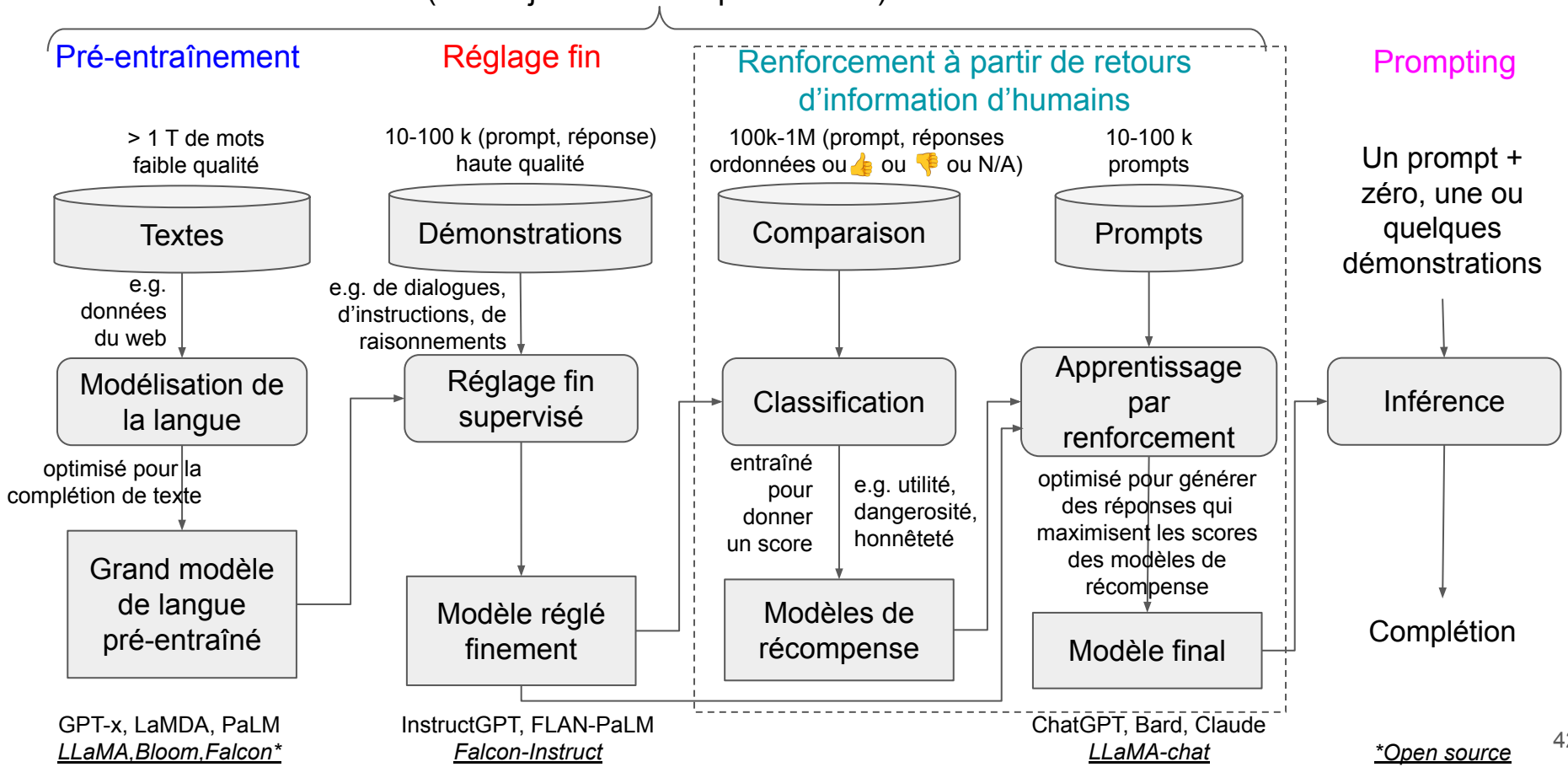
We will observe summary before fine-tuning FLAN-T5 with the dataset

Then we will observe it after fine-tuning FLAN-T5

Constructing ChatGPT

Source: (Chip Huyen, May'23), (llama2@Google), (Ouyang@OpenAI, 22)

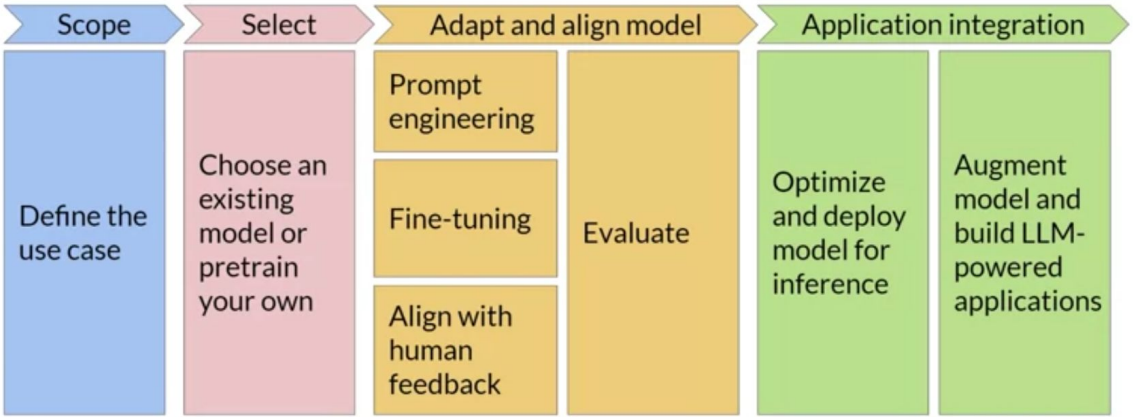
Apprentissage de modèles
(avec ajustement de paramètres)



Conclusion

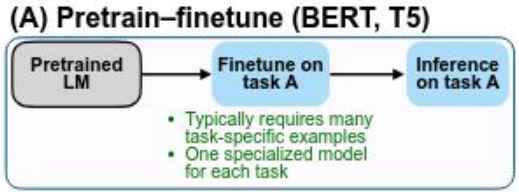
Generative AI project lifecycle

build a large generative model, define any problem as a generation problem, use in-context learning to prompt the model, optionally instruction tuning and human alignment



V. non generative ones

pre-train a prediction/masked language model, then fine-tune on specific task with little annotated data



TODO

Trainer configuration