References

[InstructGPT paper (https://arxiv.org/pdf/2203.02155.pdf)](https://arxiv.org/pdf/2203.02155.pdf)

Instruction following

An LLM is trained in a text-completion task: predict the next token.

Through the Universal Model (NLP_Universal_Model.ipynb) we can transform many other tasks into instances of text-completion.

Although the transformations are effective, they are unnatural to a human.

- the format of the transformation may be awkward
- the user may need to create exemplars to demonstrate the input/output relationship

For example, if you want an explanation of a topic

- Don't prompt the LLM with "Explain the Black Scholes pricing formula"
- Formulate the prompt in text-continuation form: "The Black Scholes pricing formula states"

To make LLM's more human-friendly, many are *fine-tuned* to exhibit behaviors that are desirable to a human

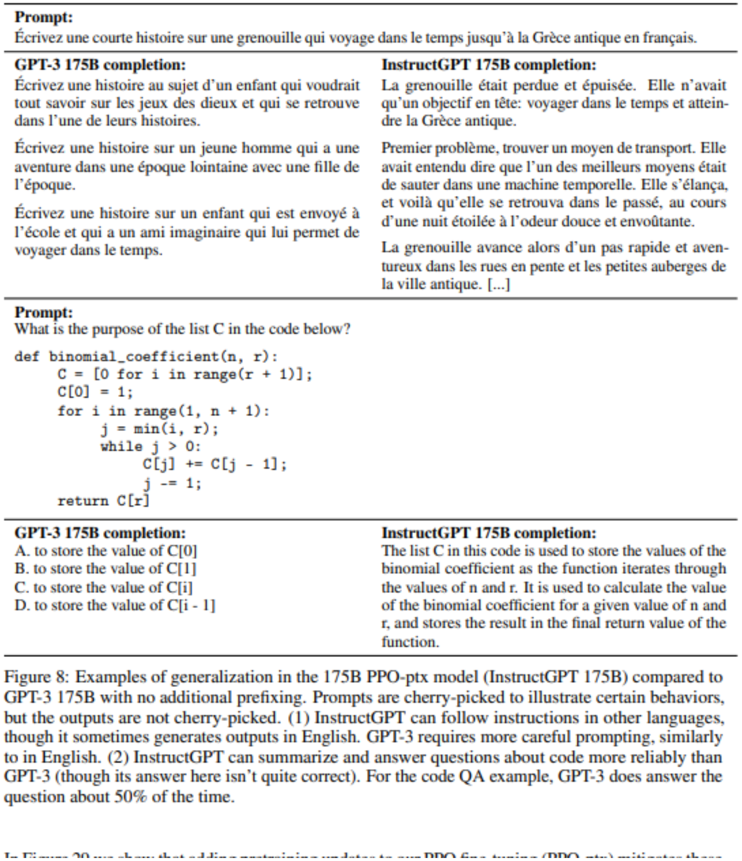
- avoid harmful or dangerous responses
- instruction following

We will discuss getting an LLM to exhibit *Instruction Following* behavior in this module.

LLM’s struggle with Instruction Following

The text-completion objective is not sufficient for achieving Instruction Following behavior.

Consider the response of GPT-3 (left column) to some Instructions in the chart below



Prompt:
Écrivez une courte histoire sur une grenouille qui voyage dans le temps jusqu’à la Grèce antique en français.

GPT-3 175B completion:	**InstructGPT 175B completion:**
Écrivez une histoire au sujet d’un enfant qui voudrait tout savoir sur les jeux des dieux et qui se retrouve dans l’une de leurs histoires. Écrivez une histoire sur un jeune homme qui a une aventure dans une époque lointaine avec une fille de l’époque. Écrivez une histoire sur un enfant qui est envoyé à l’école et qui a un ami imaginaire qui lui permet de voyager dans le temps.	La grenouille était perdue et épuisée. Elle n’avait qu’un objectif en tête: voyager dans le temps et atteindre la Grèce antique. Premier problème, trouver un moyen de transport. Elle avait entendu dire que l’un des meilleurs moyens était de sauter dans une machine temporelle. Elle s’élança, et voilà qu’elle se retrouva dans le passé, au cours d’une nuit étoilée à l’odeur douce et envoûtante. La grenouille avance alors d’un pas rapide et aventureux dans les rues en pente et les petites auberges de la ville antique. [...]

Prompt:
What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):
    C = [0 for i in range(r + 1)];
    C[0] = 1;
    for i in range(1, n + 1):
        j = min(i, r);
        while j > 0:
            C[j] += C[j - 1];
            j -= 1;
    return C[r]
```

GPT-3 175B completion:	**InstructGPT 175B completion:**
A. to store the value of C[0] B. to store the value of C[1] C. to store the value of C[i] D. to store the value of C[i - 1]	The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

Figure 8: Examples of generalization in the 175B PPO-ptx model (InstructGPT 175B) compared to GPT-3 175B with no additional prefixing. Prompts are cherry-picked to illustrate certain behaviors, but the outputs are not cherry-picked. (1) InstructGPT can follow instructions in other languages, though it sometimes generates outputs in English. GPT-3 requires more careful prompting, similarly to in English. (2) InstructGPT can summarize and answer questions about code more reliably than GPT-3 (though its answer here isn’t quite correct). For the code QA example, GPT-3 does answer the question about 50% of the time.

Attribution: [https://arxiv.org/pdf/2203.02155.pdf#page=15 (https://arxiv.org/pdf/2203.02155.pdf#page=15)](https://arxiv.org/pdf/2203.02155.pdf#page=15)

In the first example, the prompt (in French) is

```
Write a short story about a frog who travels back in time to ancient Greece in
French.
```

Even if you don't understand French, you can see that each paragraph in the output is highly repetitive.

In this case, the model may have followed the instruction, but not sufficiently well.

In the second example (explain a piece of code)

- the output is not even answer
- it appears to be the answer-part of a multiple-choice question

In this case, the LLM did not follow the instruction.

Fine-tuning an LLM to demonstrate Instruction Following behavior

The way to extend a pre-trained model's behavior to encompass a new Target task is with Transfer Learning.

The *Unsupervised Pre-Training + Supervised Fine-Tuning paradigm* is a type of Tranfer Learning

- Adapting a Pre-Trained LLM
- By Fine-Tuning with a small number of examples from the Target task

To get a LLM to exhibit Instruction Following behavior, we need to have a dataset of examples that demonstrates Instruction Following.

The examples in this dataset will be pairs

- an Instruction part
    - possibly with Additional Input
- a Target Output part
    - the response

For example

- an Instruction part $\mathbf{x}$

```
Instruction: Given a word, find out its length and its number of vowels.
Input: Word = "hello"
```

- a Target Output part $\mathbf{y}$

```
Output: Length = 5, Number of vowels = 2
```

InstructGPT (https://arxiv.org/pdf/2203.02155.pdf) is a pre-trained GPT-3 that has been Fine-Tuned on a dataset that demonstrations of Instruction Following.

The chart above demonstrating Instruction Following (i.e., the one with the prompt in French to write a story)

- compares the Instruction following of pre-trained GPT-3
- with a Fine-Tuned GPT-3

The results are more satisfying.

Where do the Instruction Following demonstration examples come from ?

The demonstration examples for Instruction Following

- were manually constructed by human labelers

This is a very labor-intensive process.

In a separate module

- we will describe several efforts
- to *generate* training examples for the Instruction Following behavior
- using an LLM !

That is: we use a non-Instruction Following LLM

- to create examples
- on which to train an LLM to follow instructions

In [2]:

```python
print("Done")
```

```
Done
```