

Review: Large Language Models

The *Language Modeling* task is to extend a sequence of words with the likely *next* word.

$$p(\mathbf{y}_{(t)} \mid \mathbf{y}_{(1:t-1)})$$

This process can be repeated, resulting in the generation of complete stories from a short "seed" of a few words.

Language Models are the basis for a great deal of the current revolution in Generative AI.

Language Models: the future (present ?) of NLP ?

The Language Model objective is a simple NLP training objective.

However, it seems to result in models that have the ability to easily adapt to solve *other* objectives.

This leads to a new paradigm called *Pre-training + Fine-Tuning*

- train a large model, on lots of data, using the Language Model objective
- Fine-tune this model on a small number of examples from a new Target Task

Let's learn about this objective.

- [Language Models \(NLP_Language_Models.ipynb\)](#)

Universal API/In-context Learning

Rather than having one model for every task:

- Is it possible to create a *single model* to solve every task ?

Text to text is a "Universal API" [Universal API \(LLM Universal API.ipynb\)](#).

In addition to a Large Language Model easily adapting to a new task via Fine-Tuning

- LLM's seem to have the ability to solve new Target tasks
- *without* further training (Fine-Tuning)
- just by being show instances of examples for the new task *at inference time*

This is called *In-Context Learning*.

- [In-Context Learning \(In Context Learning.ipynb\)](#).

Large Language Models (LLM) and Beyond

We give a brief history of the evolution (and rapid growth in size) of Large Language Models.

- [Large Language Models \(NLP Large Language Models.ipynb\)](#)

LLM's are powerful but are sensitive to the prompts presented to them.

We briefly introduce the idea of Prompt Engineering.

We then demonstrate how In Context Learning can be used as a "no training" approach to adapting an LLM to solve an unseen task.

- [PreTrain Prompt Predict \(NLP Beyond LLM.ipynb\)](#)
- [From LLM to Bing Search \(From GPT to BingSearch.ipynb\)](#)

In [2]: `print("Done")`

Done

