

## Review: Large Language Models

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

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 and some of the models that have been trained using it.

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

# Beyond the LLM

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

## Universal API/In-context Learning

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

Here is a very crude notebook that uses the HuggingFace inference API to experiment with in-context learning.

- [Experiment in In context learning: Colab](https://colab.research.google.com/github/kenperry-public/ML_Advanced_Fall_2023/blob/master/HF_inference_play.ipynb)  
([https://colab.research.google.com/github/kenperry-public/ML\\_Advanced\\_Fall\\_2023/blob/master/HF\\_inference\\_play.ipynb](https://colab.research.google.com/github/kenperry-public/ML_Advanced_Fall_2023/blob/master/HF_inference_play.ipynb))

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

Done

