

## Before we get started, feel free to drop in the chat:

1

Who are you? (Introduce yourself, feel free to drop your LinkedIn)

2

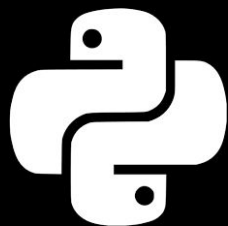
Where are you based and current role?

3

1 question you have about NLP, data science, or AI



# LLMs for me



**Introduction to LLMs &  
Generative Text**

**llmsfor.me**



Myles Harrison,  
AI Consultant & Trainer



**January 6th, 2025**

*NLP from scratch* 

# Agenda

**01**

**Welcome & Course Overview**

**02**

**Introduction to Large Language Models**

**03**

**Generative Text Models**

**04**

**Conclusion**

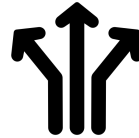
# Manifesto



Knowledge is only valuable if it is useful.



The best way to learn is by doing.



Learning is a non-linear process.



Learning is not a journey, it is guided exploration.



Teaching and learning are complementary.

# Course Overview

The course will run 6 weeks from Monday, January 6th to Monday, February 10th, 2025.

Course sessions are 7-10 PM EST on Monday evenings.



**COURSE  
SESSIONS**

Mondays 7-10 PM EST

January						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	●	7	8	9	10	11
12	●	14	15	16	17	18
19	●	21	22	23	24	25
26	●	28	29	30	31	
February						
						1
2	●	4	5	6	7	8
9	●	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

# Content & Delivery

The course will span 6 live sessions of 3 hours each, covering the topics in the curriculum show on the right.

Course sessions will be held online through Google Meet.

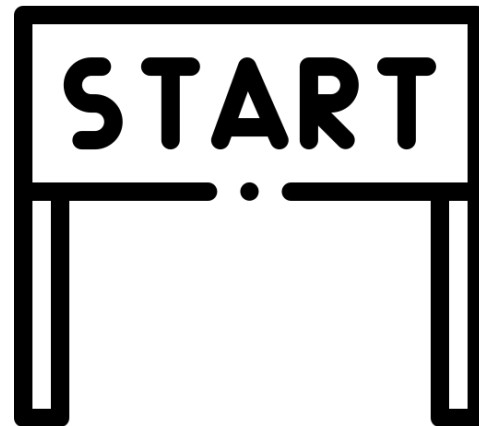
Slides will be provided in PDF format and code in Jupyter notebooks which can either be run locally or through Google Colab.

- 1** Introduction to LLMs and Generative Text
- 2** Fine-tuning LLMs, PEFT and Quantization
- 3** GPT and the OpenAI Ecosystem
- 4** Developing Large Language Model Applications Locally
- 5** Multimodal LLMs and Frameworks
- 6** Case Study in LLMs and Generative AI

# Introduction to LLMs & Generative Text

In Part 1, we will introduce the field of Large Language Models (LLMs) and Generative Text models.

We will also get started working with LLMs with our first steps with the Hugging Face library.



# Fine-tuning LLMs, PEFT and Quantization

In this continuation from Part 1, we will look at fine-tuning LLMs with our own datasets to modify their behavior.

We will also look at approaches for making this more computationally tractable, collectively known as Performance Efficient Fine-Tuning (PEFT) techniques, as well as model quantization.





# GPT and the OpenAI Ecosystem

In the third session of the course, we will get introduced to the ecosystem of models created by OpenAI as well as the OpenAI platform.

We'll make calls to the OpenAI API programmatically in Python as a precursor to building an LLM application backed by one of the GPT-series of models.



# Developing Large Language Model Applications Locally

In the fourth session of the course, we'll be looking at frameworks for working with LLMs locally.

This will build upon our work using Hugging Face, and we will also look at the Ollama framework and associated tooling.



# Multimodal LLMs and Frameworks

In the penultimate session of the program, we will dive into multimodal models by looking at image generation with models such as Stable Diffusion and Flux.

We will also look at frameworks for running image generation models locally such as ComfyUI.

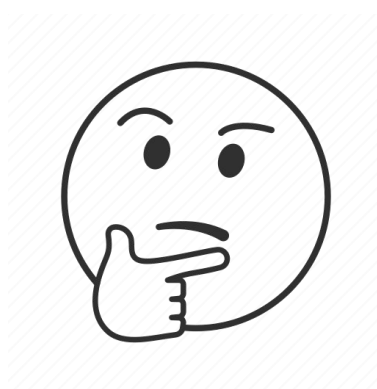
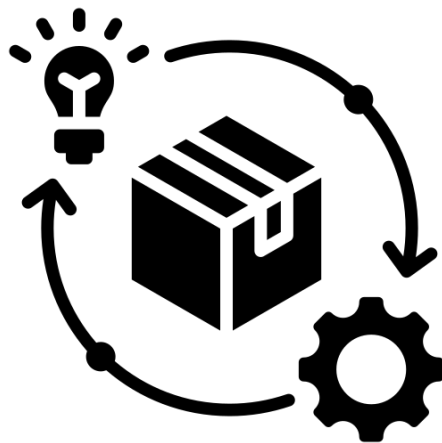
# stability.ai



# Case Study in LLMs and Generative AI

In the final session of the program, we will apply all we've learned together to build a simple MVP LLM application for a case study.

We'll also review everything we've learned and look at potential next steps on your learning journey into GenAI as the course concludes.




# Reminder Pricing & Payment

This course is offered on a Pay-What-You-Can (PWYC) basis.

You may pay any amount for the course (including \$0), based on what you are able to comfortably afford and that you feel the course is worth.

I would appreciate your support in my developing the course and future content.

You may pay at any time during the course or after the course concludes.

*NLP from scratch* 

Pay NLP from scratch

CA\$0.00

+ tax ⓘ



NLP for me - PWYC

CA\$0.00

NLPfor.me is an online Pay-What-You-Can course in NLP.

You may pay as much as you're comfortably able and... ▼

---

Subtotal

CA\$0.00

Tax ⓘ

CA\$0.00

---

Total due

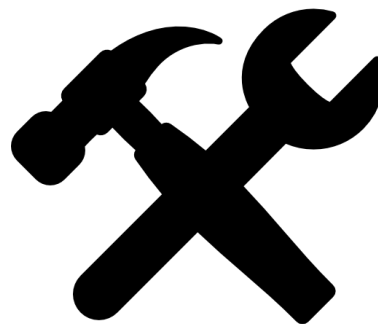
[nlpfromscratch.com/pwyc](https://nlpfromscratch.com/pwyc)



# Required Tooling

In the first part of the course, we'll be working exclusively in Google Colab, which will make the technical requirements much easier.

For sessions which require local LLM development or other software and tools, you will be notified well in advance in order to allow you time to install what is required and familiarize yourself with it in advance of each weekly session.





# **Intro to LLMs**

# What the Heck is an LLM?

A *large language model* (LLM) is a type of machine learning model.

More specifically, LLMs are a kind of *neural network* or *deep learning* model, a type of model based upon imitating the structure of neurons in the brain.

The “large” in large language models refers to both the size of the models - most modern LLMs being composed of hundreds of millions, billions, or now even trillions (!) of parameters - as well as the data they are trained upon, which is typically very large bodies of text (trillions of words).

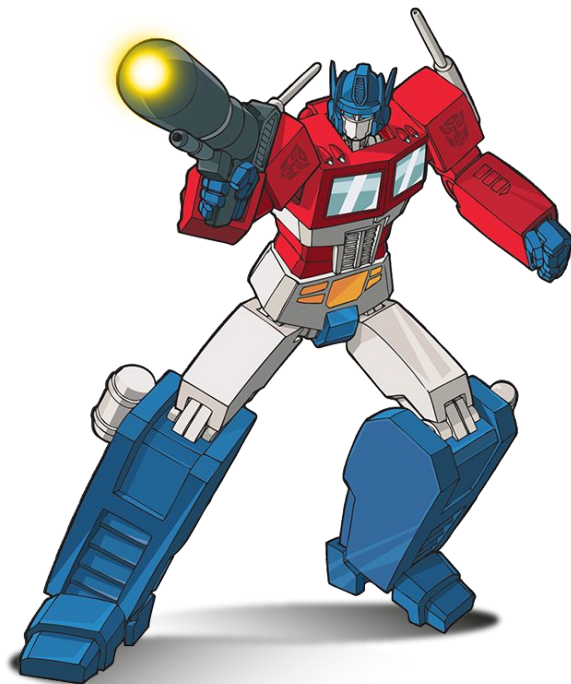
Large language models currently represent the state of the art in natural language processing (NLP) applications and the vast majority are based upon the *transformer architecture*.





# The Transformer Architecture

- Groundbreaking paper "Attention is All You Need" from Google researchers (Vaswani et al, 2017) introduced Transformer architecture
- Original application in machine translation but now general purpose and applied to a myriad of other tasks
- Represents the state of the art for LLMs and also applied in domains outside of language (image generation) - virtually all new models based on this architecture
- Popularized by OpenAI and the Generative Pretrained Transformer (GPT) series of models



# Types of Transformers (not Decepticons)

**Encoder Only**

autoencoding models

**Decoder Only**

autoregressive models


**Encoder-Decoder**

seq2seq models

## TASKS

- Classification
- Named entity recognition
- Extractive QA
- Masked language modeling
- Text generation (Causal language modeling)
- Translation
- Summarization
- Generative QA

Credit: [Abby Morgan](#)

*NLP from scratch* 

# Language Modeling Tasks - Two Examples

The rain in [MASK] falls mainly in the plain.  
The rain in **Spain** falls mainly in the plain.

Masked Language  
Modeling (MLM)

The rain in Spain ? ? ? ? ? ?  
The rain in Spain **falls** ? ? ? ? ?  
The rain in Spain falls **gracefully** ? ? ?  
The rain in Spain falls gracefully **from** ? ?  
The rain in Spain falls gracefully from **the** ?  
The rain in Spain falls gracefully from the **sky**.

Causal Language  
Modeling (CLM)

# Foundation Models

Encoder Only



**BERT**

- Bi-directional stacked encoders
- Trained using masked token and next sentence prediction
- Highly generalizable by adding heads for different tasks
- “Foundation of foundation”
- Google Research, October 2018

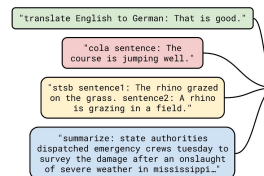
Decoder Only



**GPT**

- Stacked decoders
- Generative text model
- Innovation and improved performance with RLHF
- Size follows Moore’s Law, proprietary after GPT-2
- OpenAI, June 2018

Encoder-Decoder



**T5**

- Encoder and decoder
- Text-To-Text Transfer Transformer
- Multiple different tasks in training and objectives
- Text as input, text as output
- Google Research, June 2020

# Use Cases for Generative Text



**Code autocompletion and AI-assisted coding:** Microsoft's Github Copilot was launched in June 2022. Initially, more than  $\frac{1}{4}$  of developers' code files on average were generated by GitHub Copilot, and today with widespread adoption this is close to nearly half (~46%) and has been used by over 1M developers. In October 2023, Copilot surpassed \$100M in annually recurring revenue.



**Writing Assistants for creativity and copywriting:** AI writing assistants have arisen for improved productivity and content creation for marketing, sales, creative, and numerous other areas. For example, Google has made this a part of their core offerings with their announcement of Duet AI and Canva has introduced MagicWrite based upon OpenAI's offerings.



**Entertainment and Social:** Training generative language models on specific datasets has allowed to give them "personality". Character.ai was created by developers who previously worked on Google's LaMDA model, offers chatbots based upon fictional characters and famous individuals. It is #2 on Anderssen- Horowitz's list of top 50 most popular GenAI web products (Sept 2023).

# GPT - The Household name of LLMs



**GPT-1**

115M parameters



Toronto Book Corpus  
~800M words



**GPT-2**

1.5B parameters



WebText (8M docs, 40GB)



**GPT-3**

175B parameters



CommonCrawl, Books 1+2,  
WebText, Wikipedia (~45 TB?)



**GPT-4**

1.7T parameters?



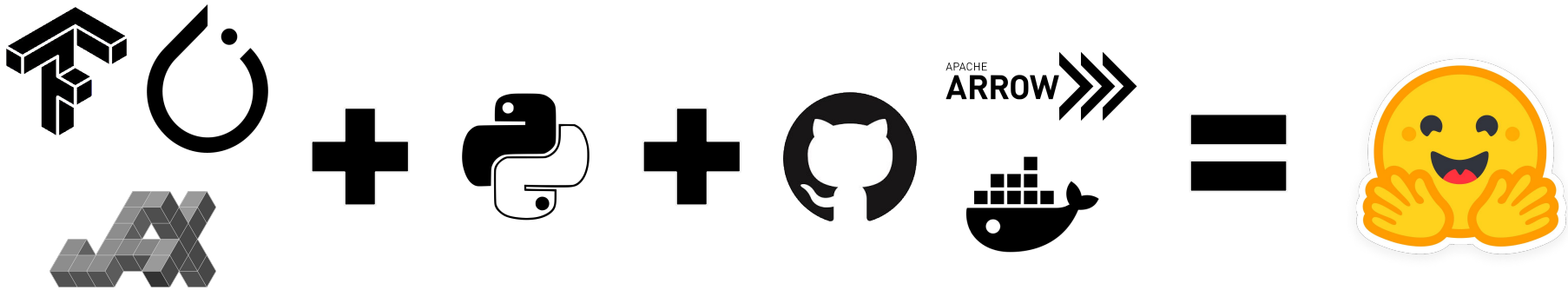
# Hugging Face


Hugging Face is a software company founded in 2013 and based in New York city. As of August 2023, the company is in Series 'D' funding with a valuation of \$4.5B and backing from companies such as Salesforce, Google, Amazon, IBM, Nvidia, AMD, and Intel.

While this name refers to the company, it also refers to the software and platform they develop for working with large language models and data in the natural language processing and other domains.

The datasets library allows working with data hosted on the platform, and the transformers library for working with models of this type. There are also other libraries for working with specialized types of models (e.g. diffusers for diffusion models) and data processing and model optimization.





*NLP from scratch* 



# Creating a Hugging Face account



Hugging Face

Search models, datasets, users...

Models

Datasets

Spaces

Docs

Solutions

Pricing

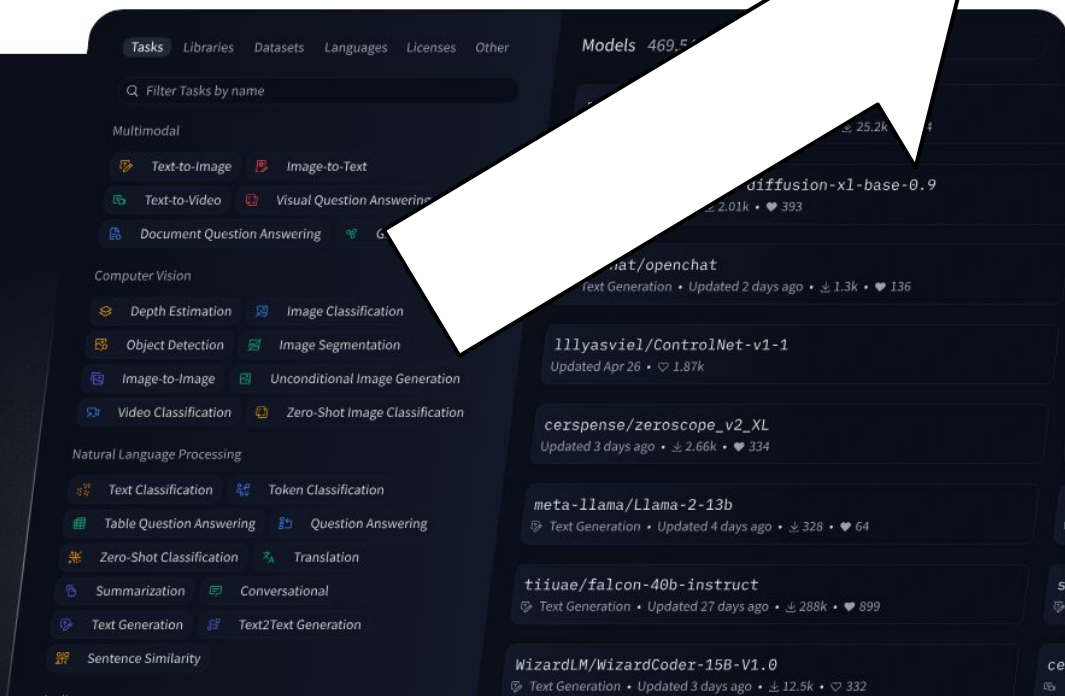
Log In

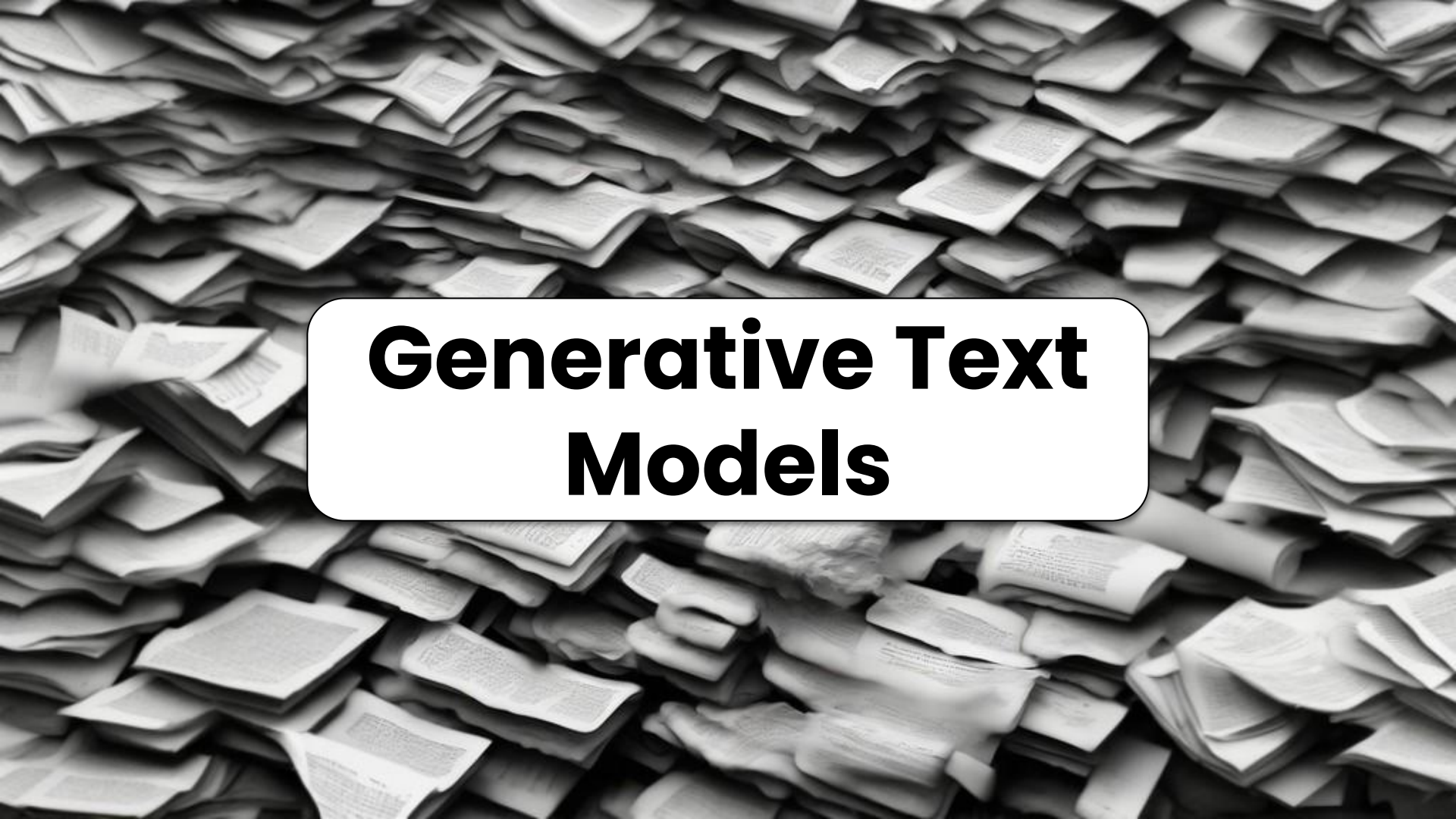
Sign Up



The AI community  
building the future.

The platform where the machine learning community



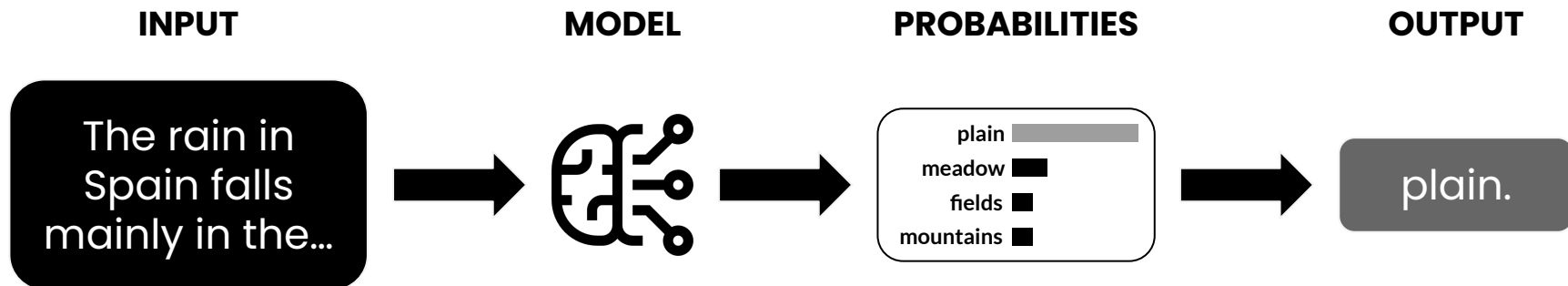


# **Generative Text Models**

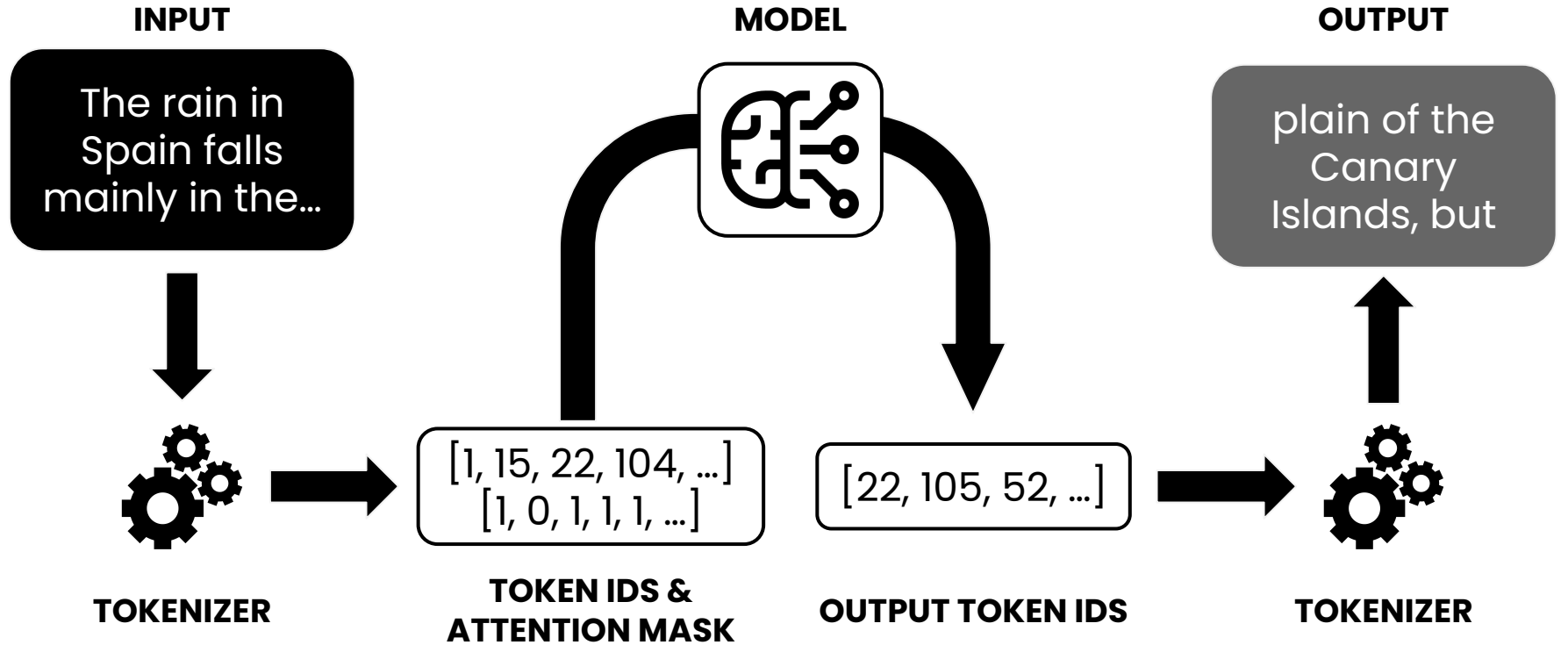
# Generating Text with a Model

When generating text, the model assigns probabilities to all possible tokens based on its understanding of the entire context. It then selects the next token in the output based on these probabilities.

There are different parameters we can specify when generating text from a model to vary the outputs thereof.

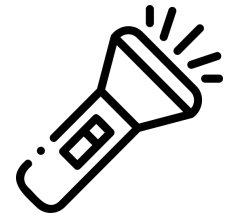


# Generating Text in Hugging Face 🙌

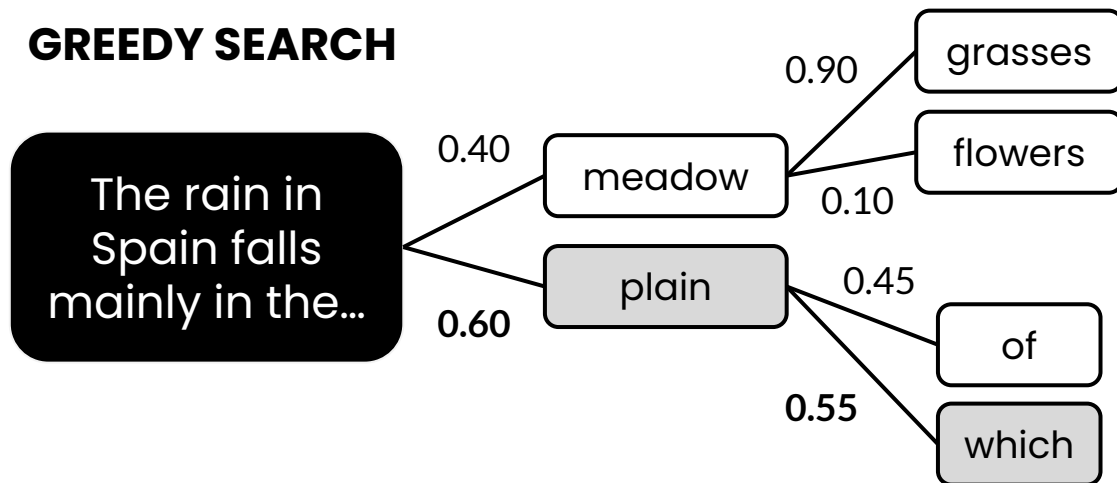


# Greedy Search vs. Beam Search

- *Greedy search*, is the simplest decoding strategy, and chooses the token with the highest probability at each step. However, this may not always lead to the most coherent outputs since it prioritizes the most probable token at each step without considering the overall context.
- *Beam search*, on the other hand, keeps track of a fixed number (the *beam width*) of the most probable tokens at each step, and chooses the combination of multiple tokens with the highest overall probability over the beam width.
- In general, beam search tends to work well with tasks such as translation or summarization, where the output length is predictable, but less so in open-ended generation, where its results can be repetitive or predictable



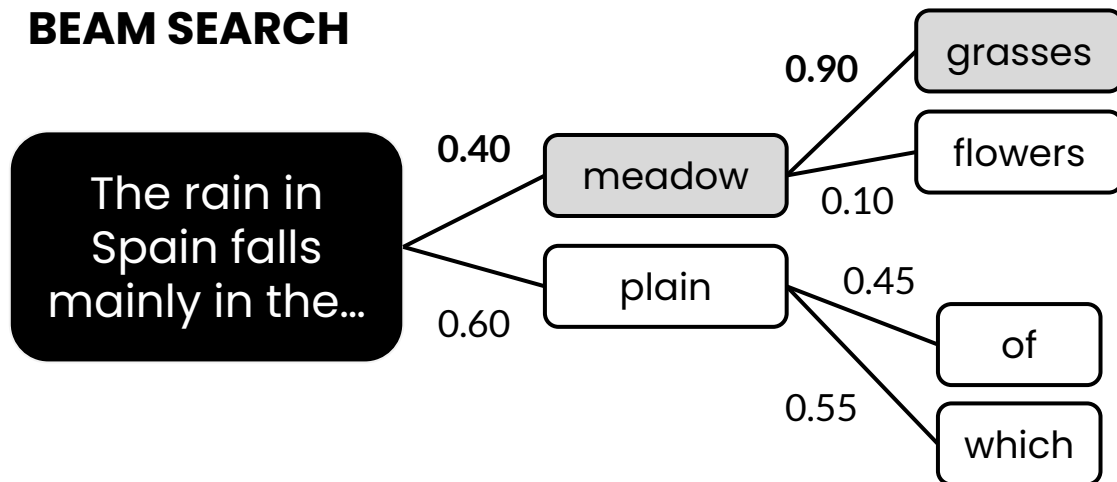
## GREEDY SEARCH



In Greedy search, the most probable next token is always selected at each point in the predicted sequence.

'Plain' is the most probable next token, followed by 'which'.

## BEAM SEARCH

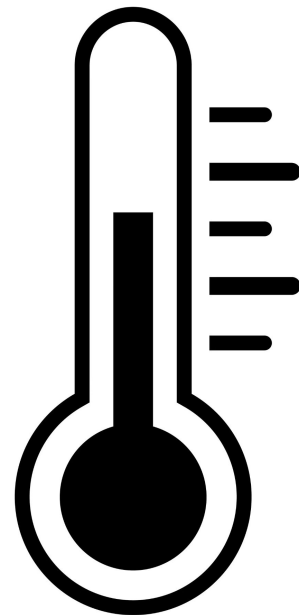


Here, for a beam width of 2,  $0.4 \times 0.9 = 0.36$  which is greater than  $0.6 \times 0.55 = 0.33$ , so these tokens are used.

The probability over the beam width is greater, even though the first token, 'meadow', has a lower probability than 'plain'.

# Temperature

- When generating text, the *temperature* refers to determines the variability of the output generated by the model
- A higher temperature value leads to more diverse and varied outputs, whereas a lower value results in more focused and deterministic results
- Setting a temperature value of 0 will result in 100% deterministic outputs (same output for a given input)
- Setting a temperature value higher will give the model too much freedom and can result in random or nonsensical outputs (gibberish)
- Lower temperatures more appropriate when performing tasks that have a "correct" answer (e.g. Q&A or summarization)



# More technically speaking

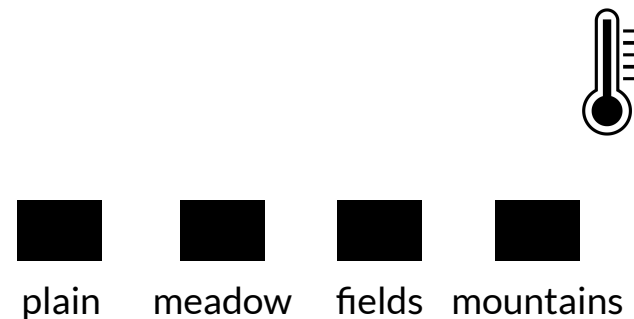
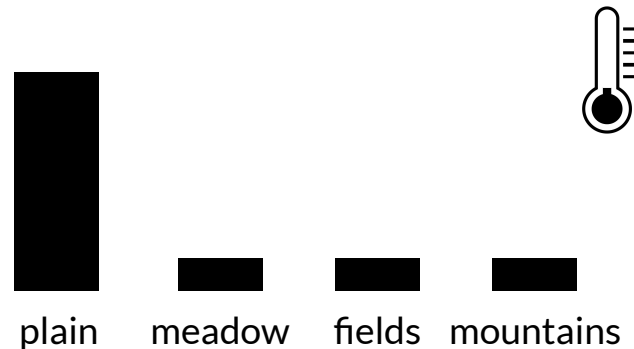
- The probability distribution of next tokens for a given input is modeled by the softmax function:

$$\text{softmax}(x_i/T) \quad i = 1, \dots, N,$$

where here,  $T$  represents the temperature and can be any number from 0 to infinity

- Therefore, as  $T$  approaches infinity, all tokens in vocabulary become equally likely
- “Reasonable” values for temperature will therefore vary by dataset model trained on and associated distribution of probabilities, vocabulary size, etc.
- In practice,  $T$  is never set to zero, but some very small number

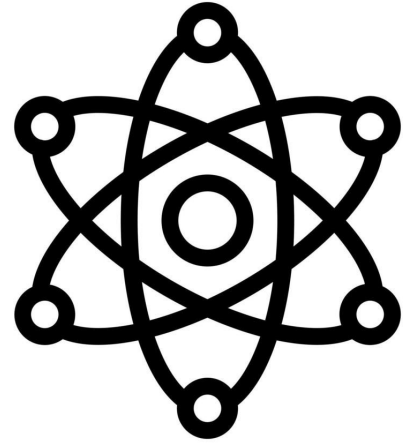
The rain in Spain falls mainly in the...





# Top-k and Top-p (Nucleus) Sampling

- Both top-k and top-p sampling are methods to introduce variety into text outputs and make them less deterministic for a given input
- In *top-k* sampling, instead of selecting from all possible tokens, only the top  $k$  most probable tokens by rank are considered
- In *top-p*, or *nucleus sampling*, only the most probable tokens whose collective probability is greater than or equal to a specified threshold,  $p$ , are considered
- For both methods, the total probability mass is redistributed amongst the new set of possible tokens



$$\sum_{x \in V(p)} P(x|x_{1:i-1}) \geq p.$$

# The rain in Spain falls mainly in the...

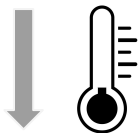
## Top-k, k = 5

token	probability	cumulative	rank
plain	0.5	0.5	1
meadow	0.15	0.65	2
field	0.1	0.75	3
mountains	0.05	0.8	4
afternoon	0.05	0.85	5
sunshine	0.025	0.875	6
cities	0.025	0.9	7
morning	0.05	0.95	8
evening	0.025	0.975	9
farms	0.025	1	10

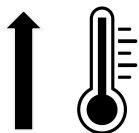
## Top-p, p = 0.8

token	probability	cumulative	rank
plain	0.5	0.5	1
meadow	0.15	0.65	2
field	0.1	0.75	3
mountains	0.05	0.8	4
afternoon	0.05	0.85	5
sunshine	0.025	0.875	6
cities	0.025	0.9	7
morning	0.05	0.95	8
evening	0.025	0.975	9
farms	0.025	1	10

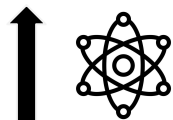
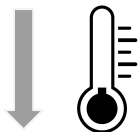
# Finding a balance: Temperature and sampling



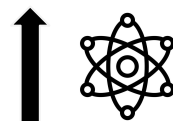
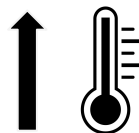
**Low temperature, low top-p:** Consider a narrow range of high-probability tokens. This combination results in highly focused and predictable output.



**High temperature, low top-p:** Consider a narrow range of high-probability tokens with near equal likelihood. The high temperature may still introduce some randomness in the output.

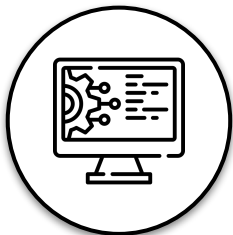


**Low temperature, high top-p:** Consider a wider range of tokens but only select the most probable ones, resulting in less varied output.



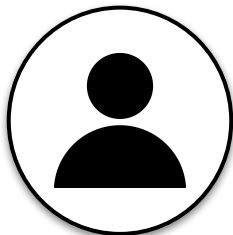
**High temperature, high top-p:** Consider a wide range of tokens with increased likelihood of selecting any individual token. Can result in highly varied but less coherent output.

# Message Roles



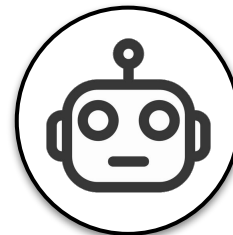
**SYSTEM**

Sets the behavior of the assistant - how it should behave at the conversation level (optional)



**USER**

Provide requests or input to which the assistant will respond (i.e. the prompts)



**ASSISTANT**

Responses from the model. Can be used to include conversation history when it is important (optional)

# Training a chat LLM - data format

```
conversation = [  
    {"role": "user", "content": "Hello, how  
are you?"},  
    {"role": "assistant", "content": "I'm  
doing great. How can I help you today?"},  
]  
  
Tokenizer = AutoTokenizer.from_pretrained(  
    "microsoft/Phi-3-mini-4k-instruct")  
  
tokenizer.apply_chat_template(conversation,  
    tokenize=False))
```

<|user|>Hello, how are  
you?<|end|>

<|assistant|>  
I'm doing great. How  
can I help you  
today?<|end|>

<|endoftext|>

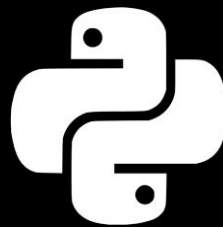
# End of Part 1

[LLMsfor.me](https://llmsfor.me)

PWYC Microcourse in LLMs and Generative AI  
January 2025

**Part 1 - Introduction to LLMs & Generative Text**

**Monday, January 6th, 2025**



**llmsfor.me**

*NLP from scratch*