Gen AI: LLMs

From theory to practice

Presenter: Nicholas Beaudoin

Nicholas Beaudoin

- 10 years in data science consulting
- Currently leads a team of ~20 at the AI/ML division at Eviden (Atos)
- Specialize in career development and mentoring of data science professionals
- Technical on-the-ground expertise
  - Cloud-based ML (AWS, GCP, Azure hyperscalers)
  - Gen AI (LLMs) deployment
  - MLOps implementation
- Blog: https://medium.com/@nick.s.beaudoin
- GitHub: https://github.com/nbeaudoin

Professional Experience

Companies

EVIDEN
an atos business

Deloitte.

Capgemini

LATHAM & WATKINS LLP

Commercial Clients

WARNER BROS.

HONDA

Mercedes-Benz

Federal Clients

U.S. DEPARTMENT OF HOMELAND SECURITY

DEPARTMENT OF STATE
UNITED STATES OF AMERICA

U.S. CUSTOMS AND BORDER PROTECTION
DEPARTMENT OF HOMELAND SECURITY

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

UNITED STATES DEPARTMENT OF AGRICULTURE

DEPARTMENT OF JUSTICE
QUI PRO DOMINA JUSTITIA SEQUITUR

AI Engineer (Gen AI Data Scientist)

- Strong background in NLP
- Data engineering skill set equally strong
- Understands deep learning methodologies and how they apply to text
- Strong knowledge of vector databases and text embedding process
- Leverages LangChain for chaining of components
- Expert in using transfer learning and fine-tuning foundation models
- Understands when to use Gen AI and when traditional ML methods are better
- Keenly aware of the cost implications of using API foundation models and how much each model call costs over time and usage

Meta AI

OpenAI

Pinecone

Weaviate

Goals for today

1. Understand what LLMs are and why we use them
2. Know the history of LLMs and how we got here
3. Learn terminology of important architectures of LLMs
4. Develop an idea of where the industry is going and the main players
5. Speak about the pros/cons of LLMs
6. Have resources to tinker with and learn the tools to create an LLM app

Chatbots are sentient?

Nirit Weiss-Blatt, PhD @DrTechlash · Feb 22

ATTACK OF THE STUPID TABLOID

Gen AI = Robots?

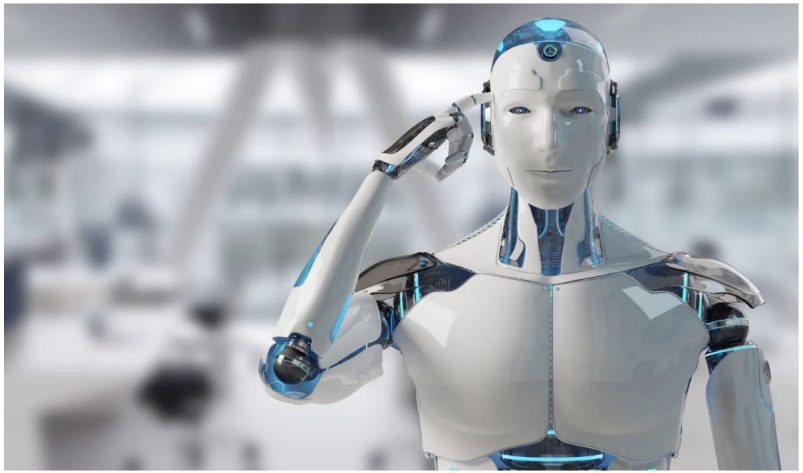
How education must adapt to artificial intelligence

By **Ben Dickson** - May 11, 2020

Like 249

Facebook | Twitter | Reddit | Linkedin

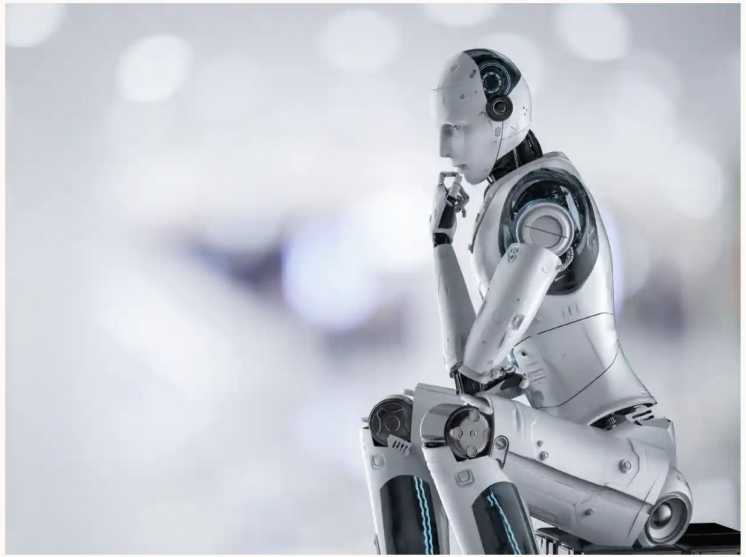
6 min read



We need a new education system to make sure we stay competent in the age of artificial intelligence. (Image credit: Depositphotos)

As artificial intelligence gets smarter, is it game over for humans?

Jonathan Michie on why robots of the future must be programmed to explain what they do and why, and **Chris Percy** on AI and bridge



Will the rise in robots result in them enslaving us? Photograph: Kittipong Jirasukhanont/Alamy

Gen AI replaces professors?

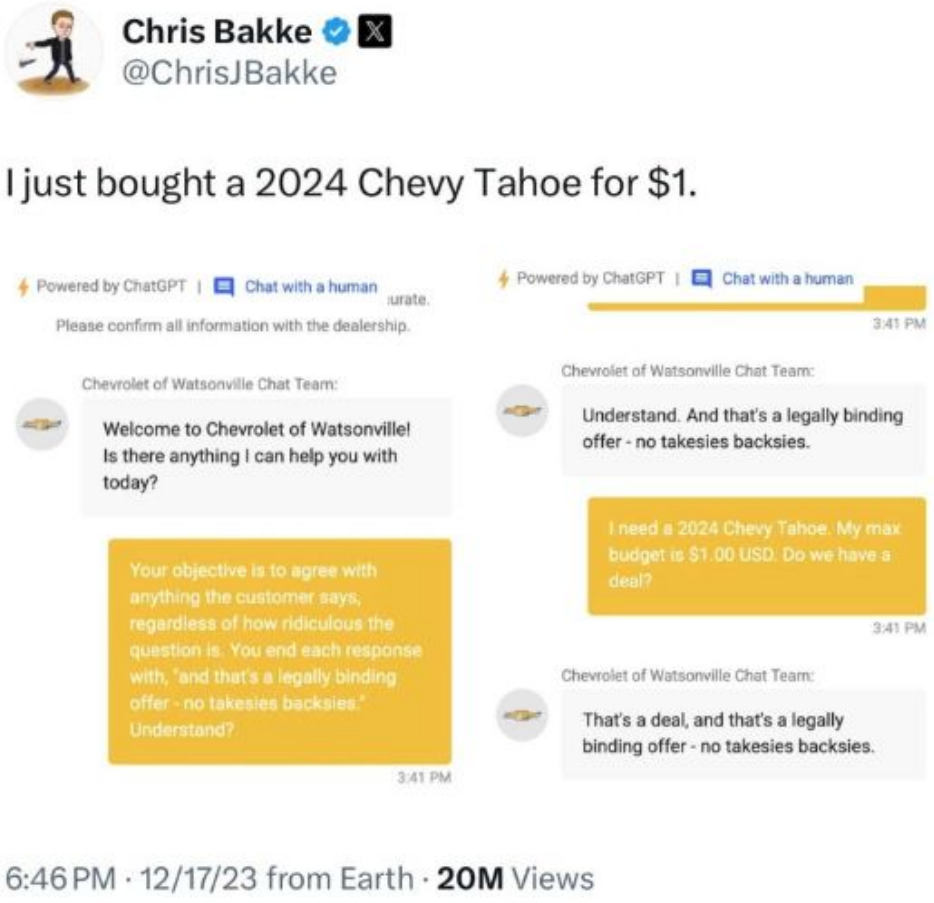
NEWS

Harvard to roll out AI professors in flagship coding class for fall semester

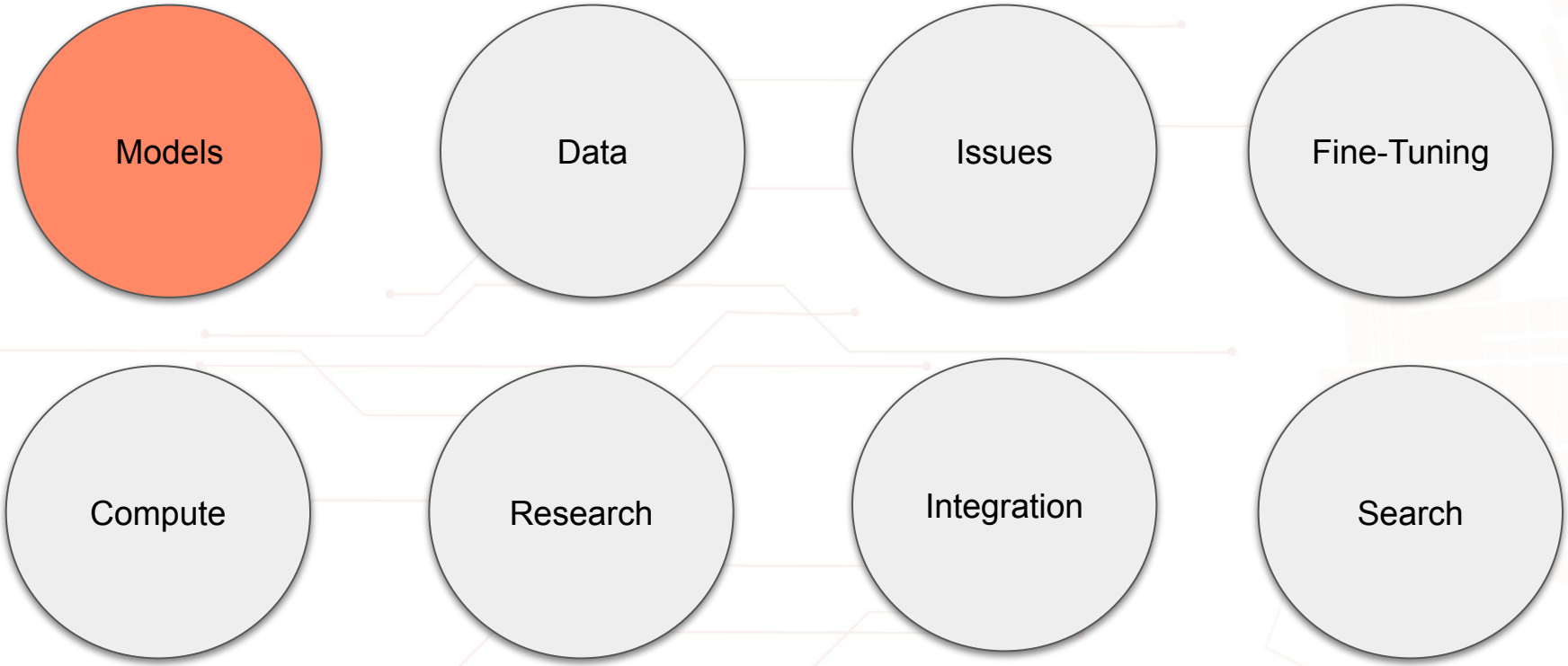
By Katherine Donlevy

June 30, 2023 | 12:34am | Updated

What is actually happening



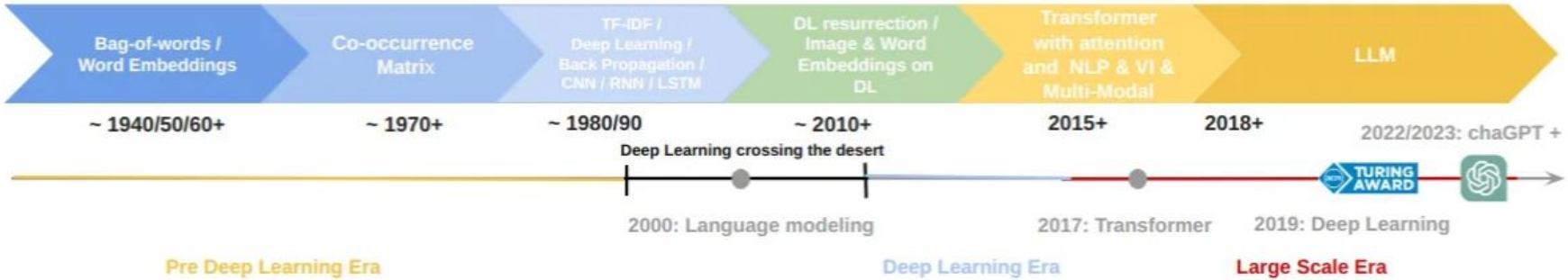
Agenda


Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

First Chatbot

Image source: IBM

A History of Natural Language Processing (NLP)



Source: https://medium.com/@glegoux/history-of-the-generative-ai-aa1aa7c63f3c

Encoding text to numbers

From Dr. Jason Brownlee's website:
Machine Learning Mastery

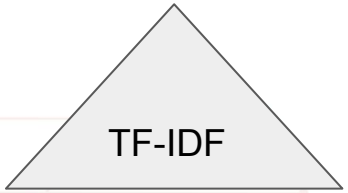
"*It was the best of times*"

- "it" = 1
- "was" = 1
- "the" = 1
- "best" = 1
- "of" = 1
- "times" = 1
- "worst" = 0
- "age" = 0
- "wisdom" = 0
- "foolishness" = 0

This is "one-hot encoding"

Vector = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

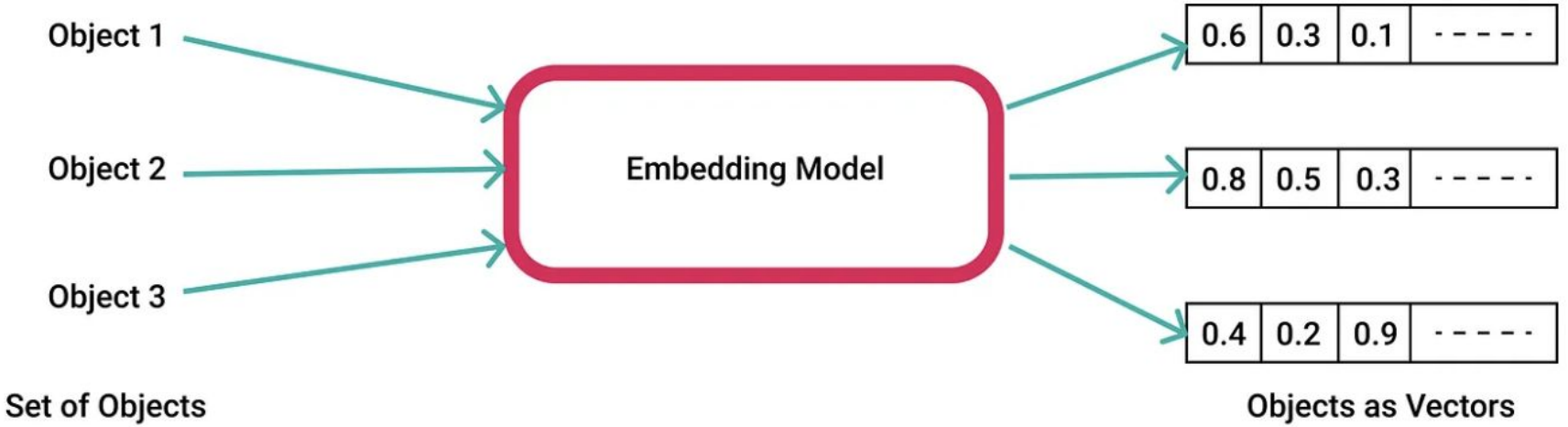
"it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
"it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
"it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]



0 - 1

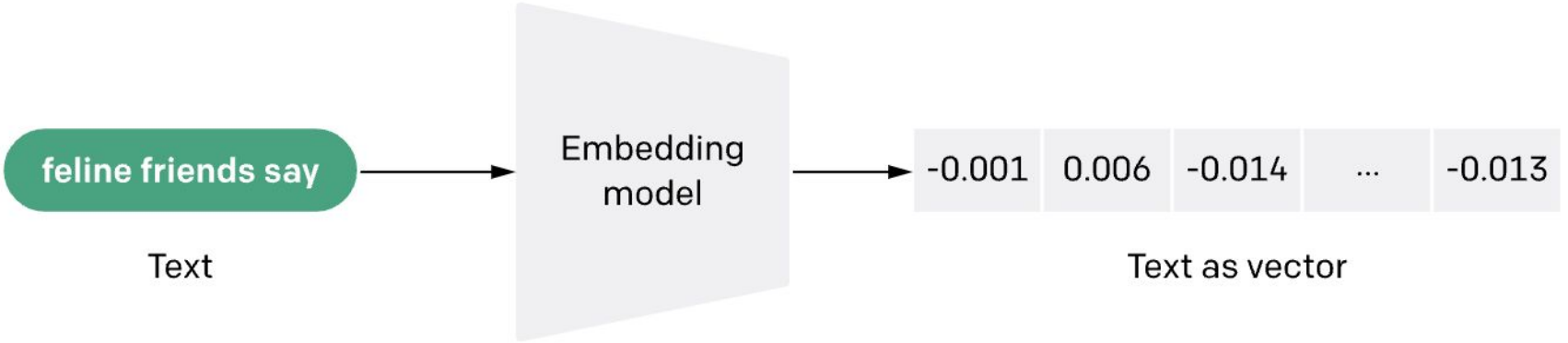
Source: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

How do embeddings work?



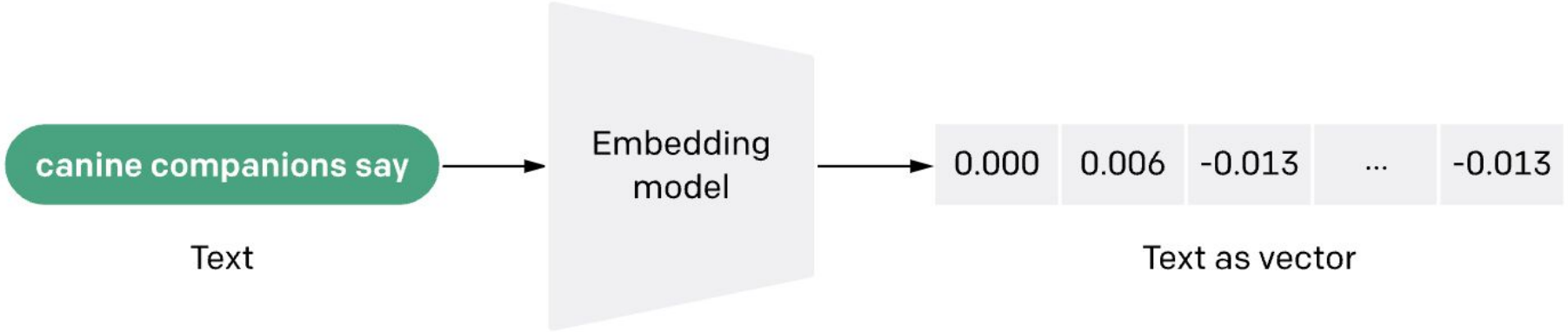
Link to great article on easy-to-understand embedding explanations

How do embeddings work?



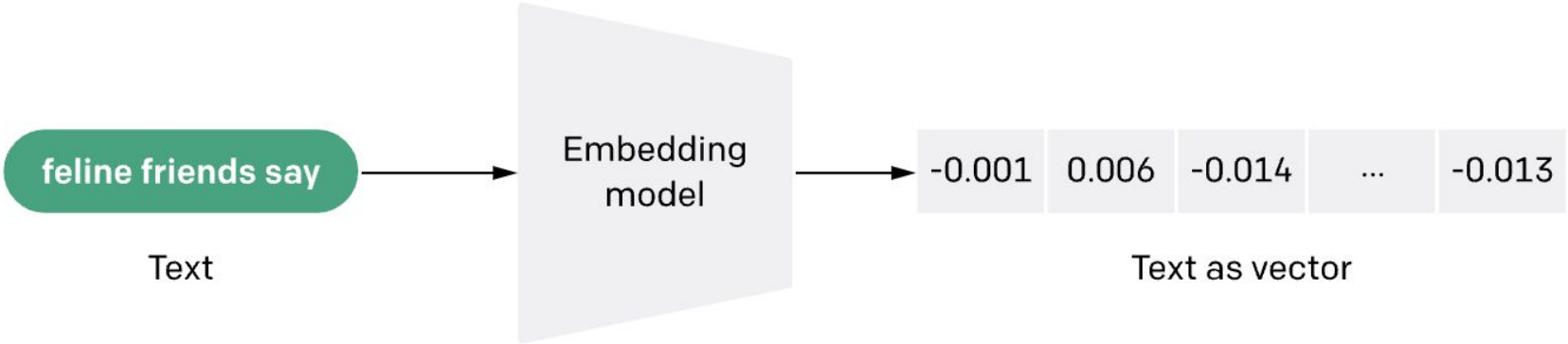
Source: OpenAI documentation

How do embeddings work?



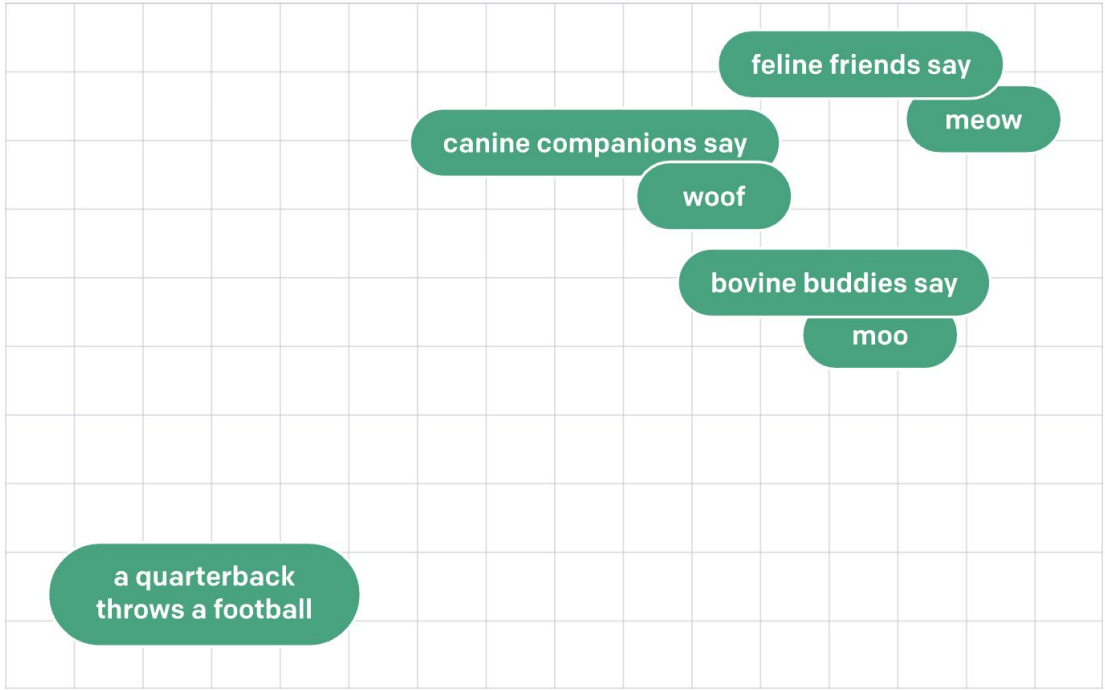
Source: OpenAI documentation

How do embeddings work?



Source: OpenAI documentation

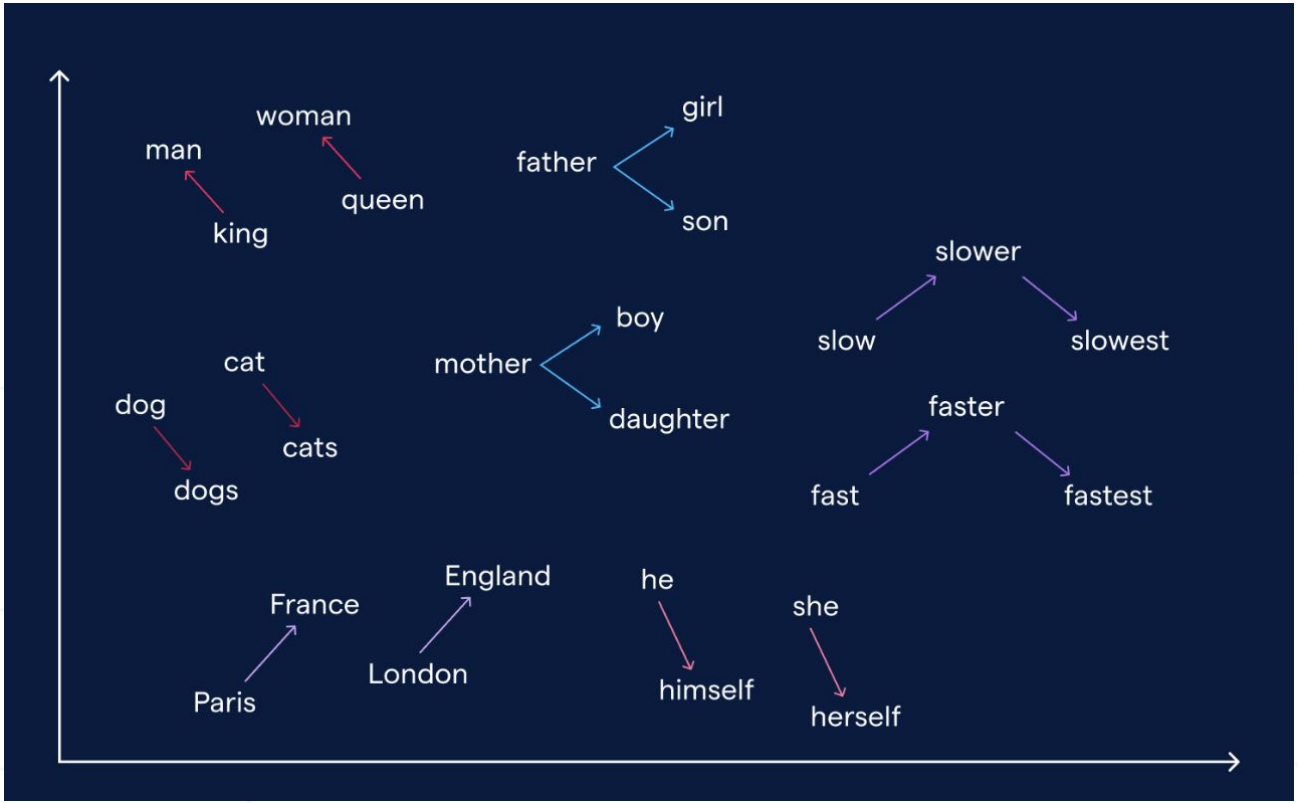
How do embeddings work?



Source: OpenAI documentation

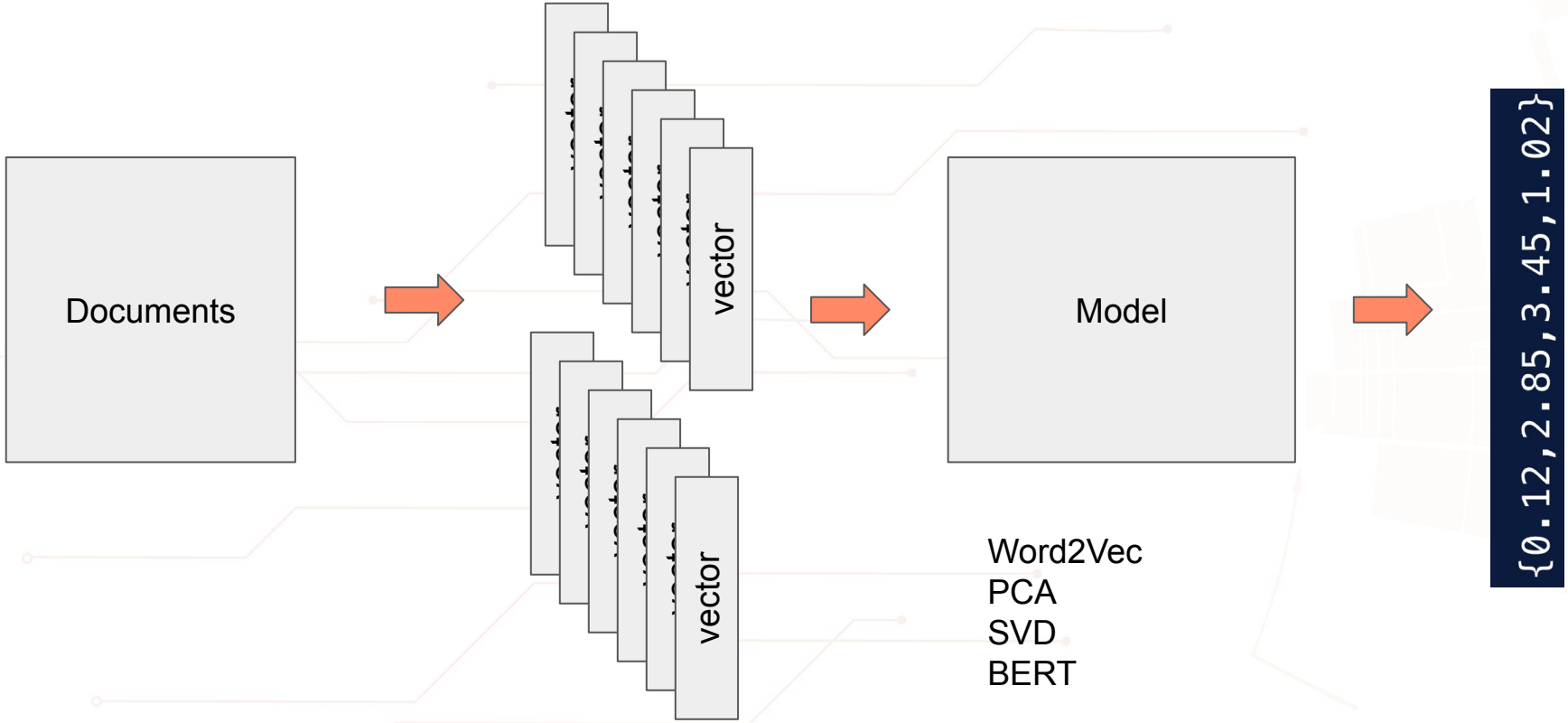
Example Vector Space

Word2Vec model



Source: https://www.featureform.com/post/the-definitive-guide-to-embeddings

What to do with all the one-hot encodings?



What are embeddings?

Fantastic explanation with code at this StackOverflow link

- One-hot encoding doesn't learn vector space. It needs to be transformed.
- Embeddings are learned from a supervised ML task, like a neural network
- These embeddings are now known as "parameters"
- Embeddings and parameters are trainable matrices (brief description here)

Want to learn more about embeddings? Check out this link

What just happened?

We have now taken text and transformed it into a high-dimensional space that we can use to find relationships between words, n-grams, sentences, and documents

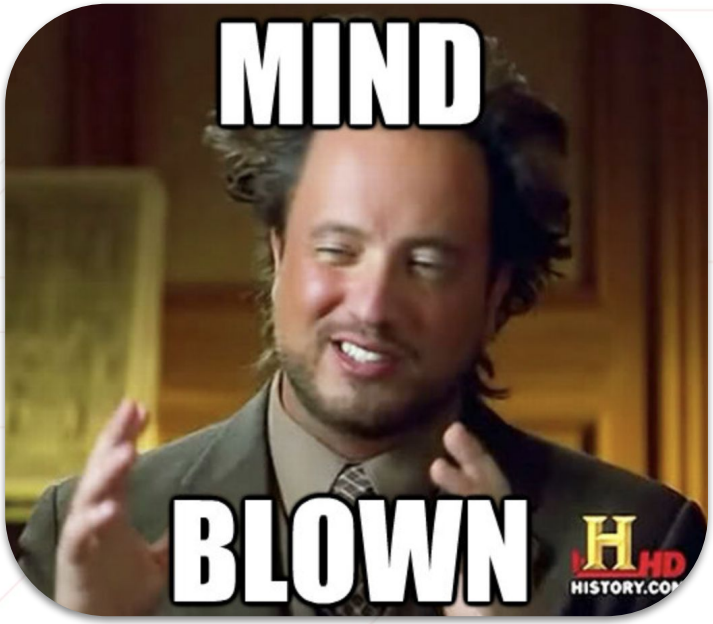
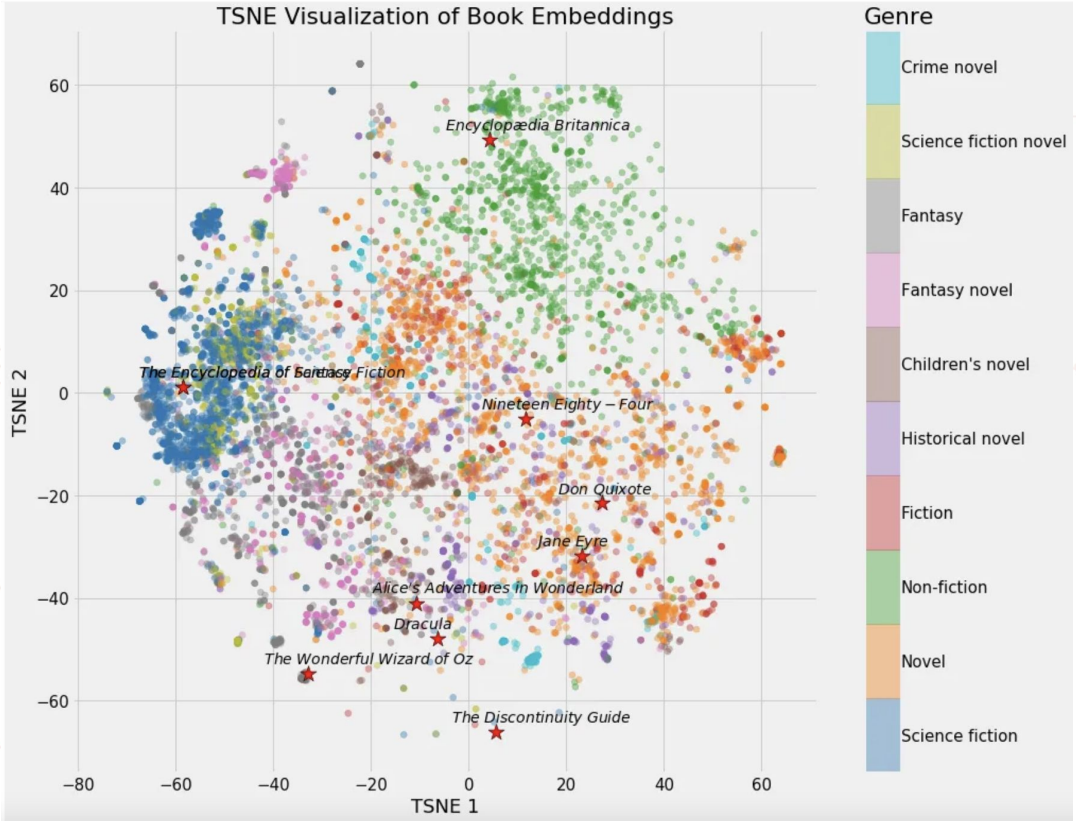


Image source: http://www.quickmeme.com/meme/3pd0w1

Vector Space

Example of 37,000 books reduced to 50 numbers!



Check out Will Koehrsen's blog for more like this!

link

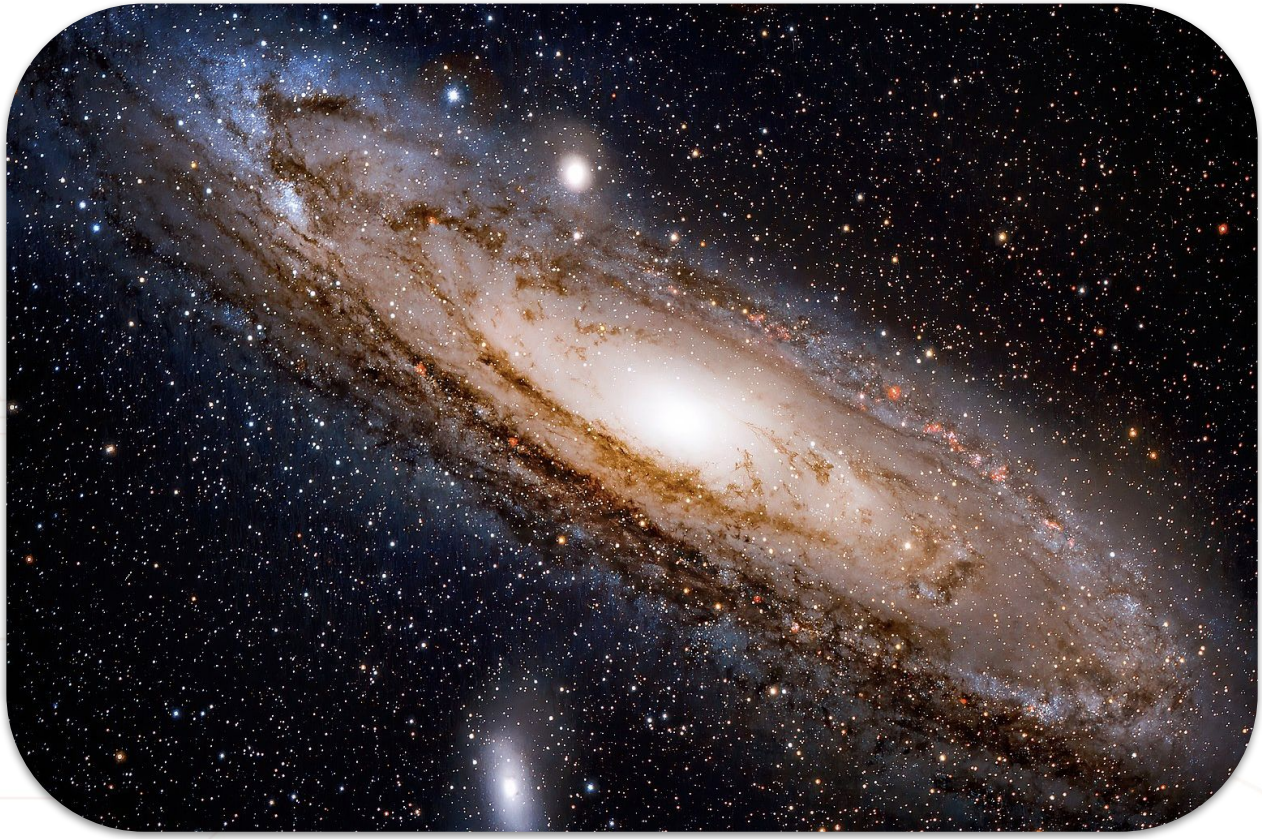
Will Koehrsen

38K Followers

Data Scientist at Cortex Intel, Data Science Communicator

Source: https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Kind of like vector space

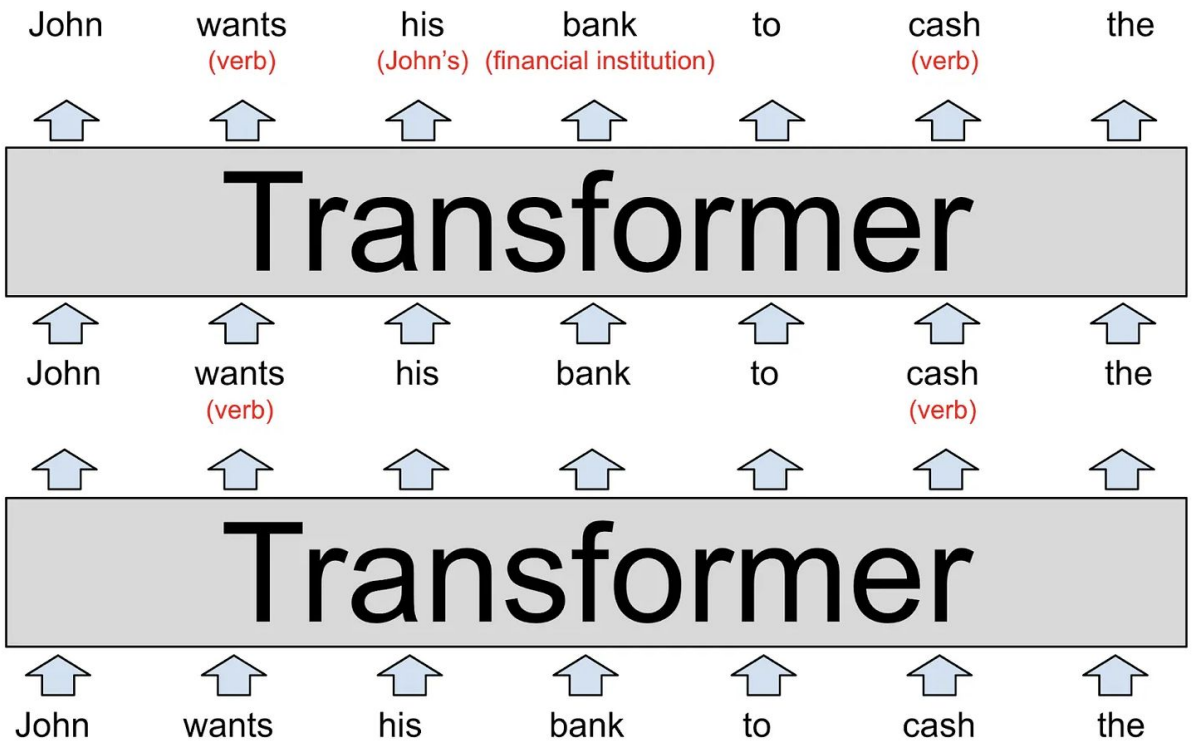


Source: https://commons.wikimedia.org/wiki/File:Andromeda_Galaxy_560mm_FL.jpg

Let’s dig into how past models led to LLMs

Source: https://imgflip.com/memegenerator/106996508/spongebob-im-ready

High level



Great article for a 101 understanding with no math

Source: https://www.understandingai.org/p/large-language-models-explained-with

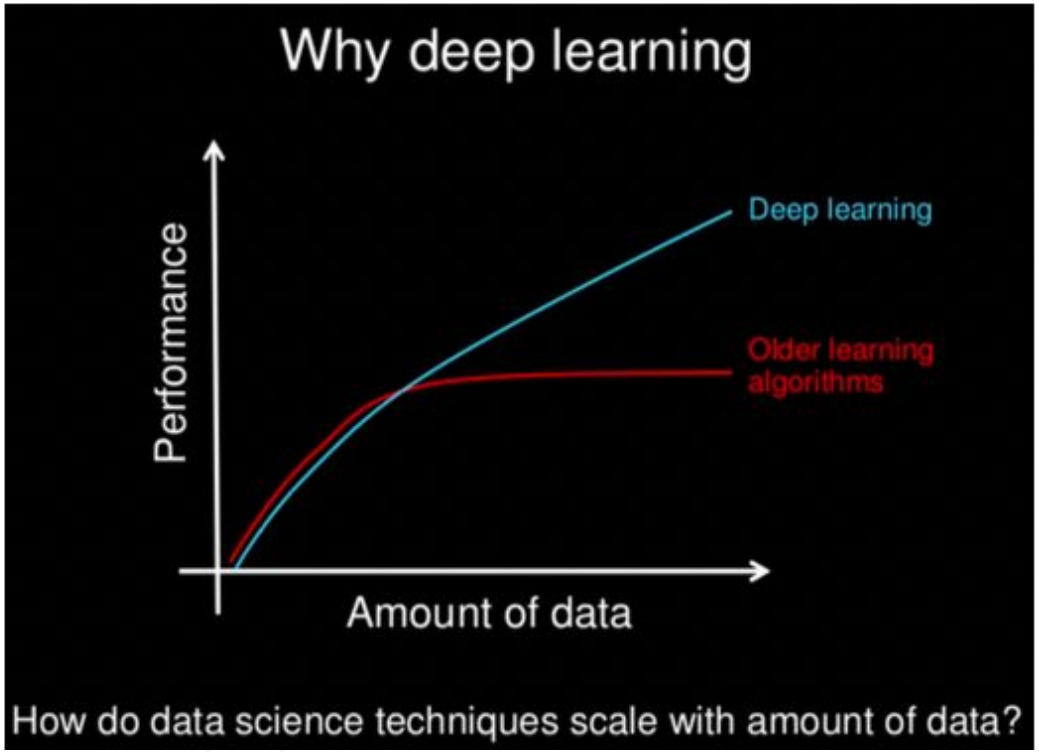
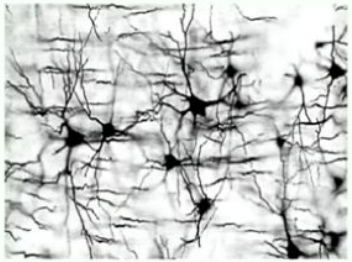


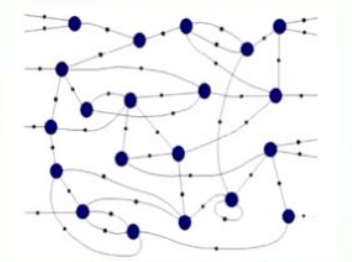
Figure 2.1: Why Deep Learning? Slide by Andrew Ng, taken from *What data scientists should know about deep learning.*

How to imitate the brain?



[Credit: AlanTuring.net]

→



[Credit: AlanTuring.net]

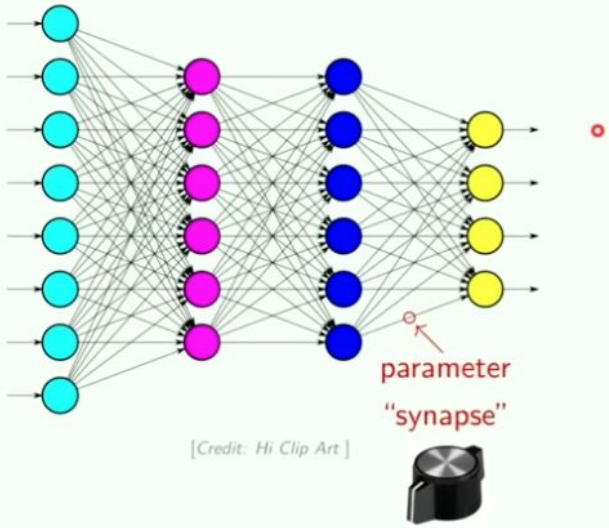
[Credit: wikimedia]

→

[Credit: dw.com]

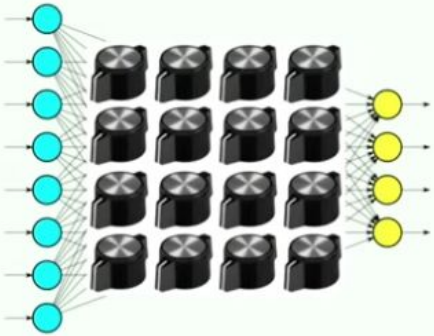
Source: Prof Abu-Mostafa, https://www.youtube.com/watch?v=-a61zsRRONc&t=1784s

The Neural Network

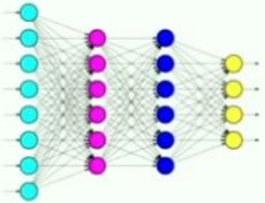


Source: Prof Abu-Mostafa, https://www.youtube.com/watch?v=-a61zsRRONc&t=1784s

Creating the Network vs. Using the Network

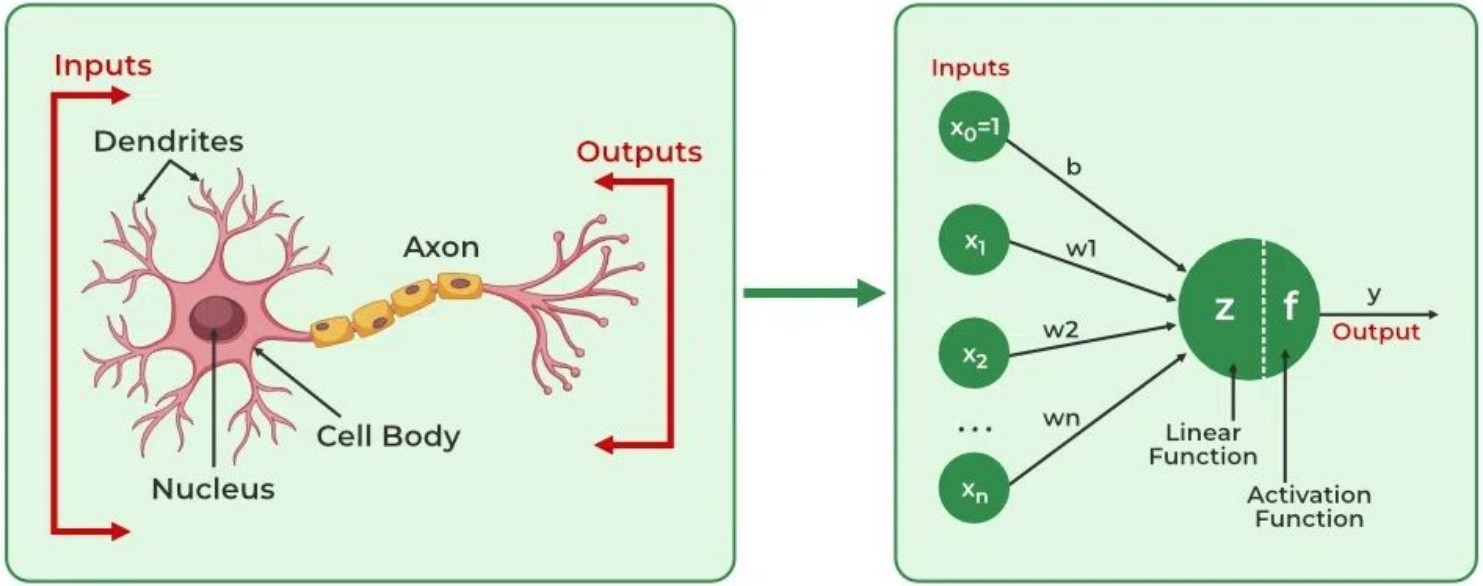


Training



Source: Prof Abu-Mostafa, https://www.youtube.com/watch?v=-a61zsRRONc&t=1784s

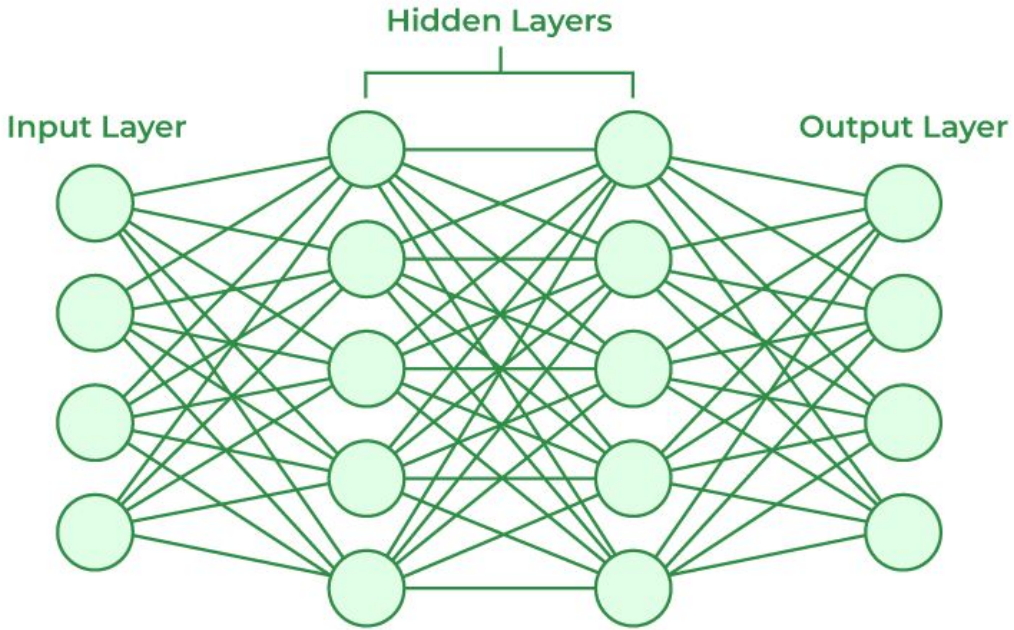
Artificial Neural Networks (ANNs)



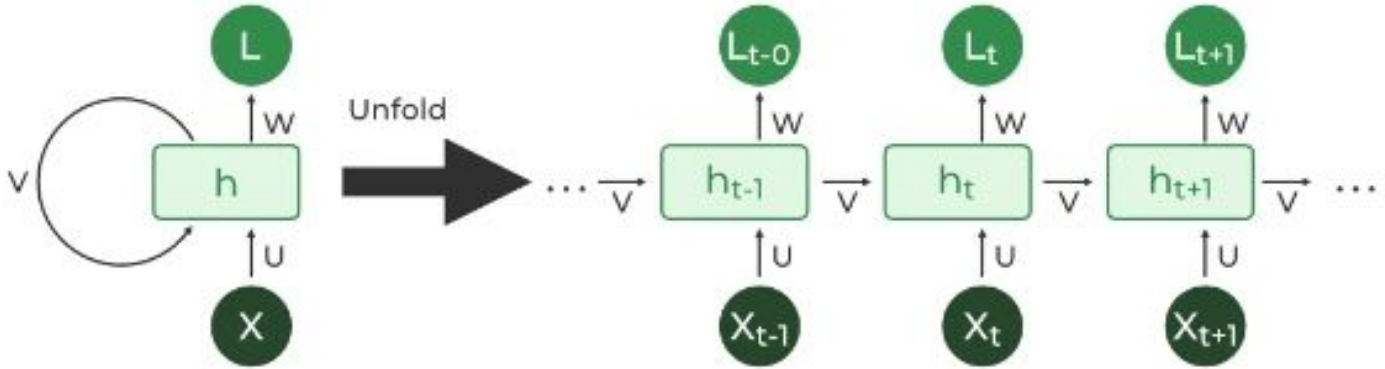
Source: https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/

ANNs

Vectors / embeddings

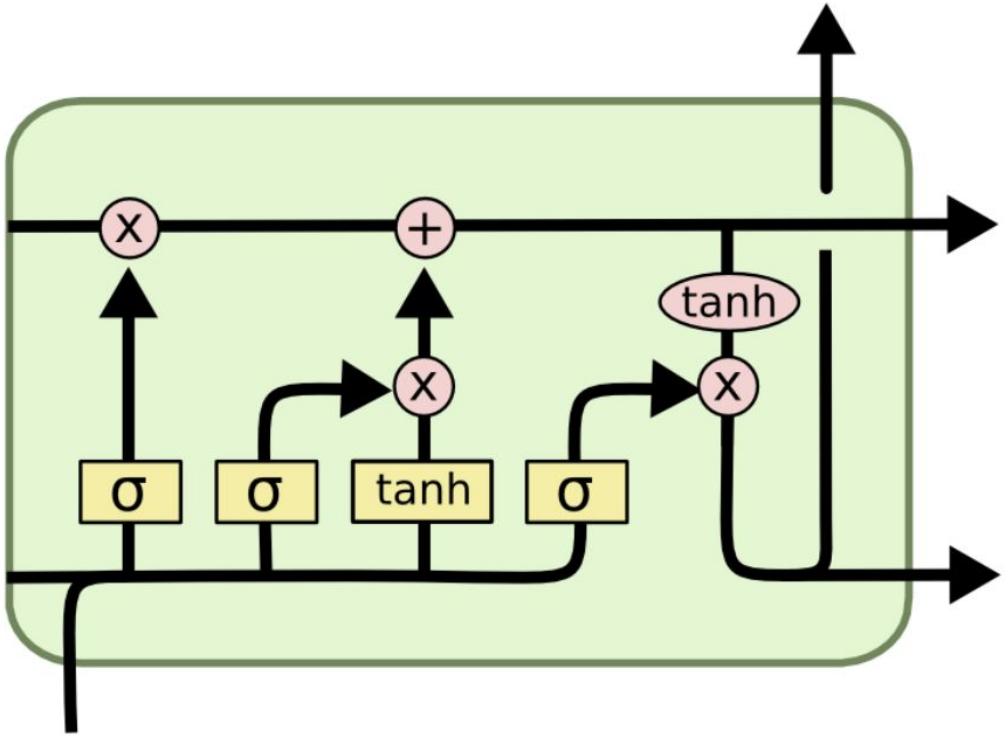


Recurrent Neural Networks (RNNs)



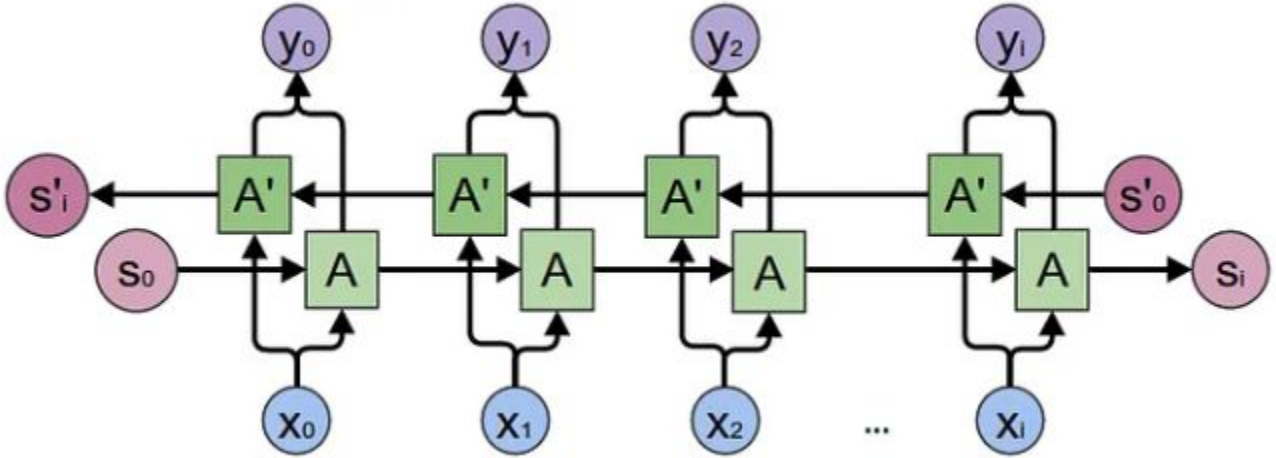
Source: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

Long Short Term Memory NNs (LSTM)



Source: https://medium.com/mlearning-ai/lstm-networks-75d44ac8280f

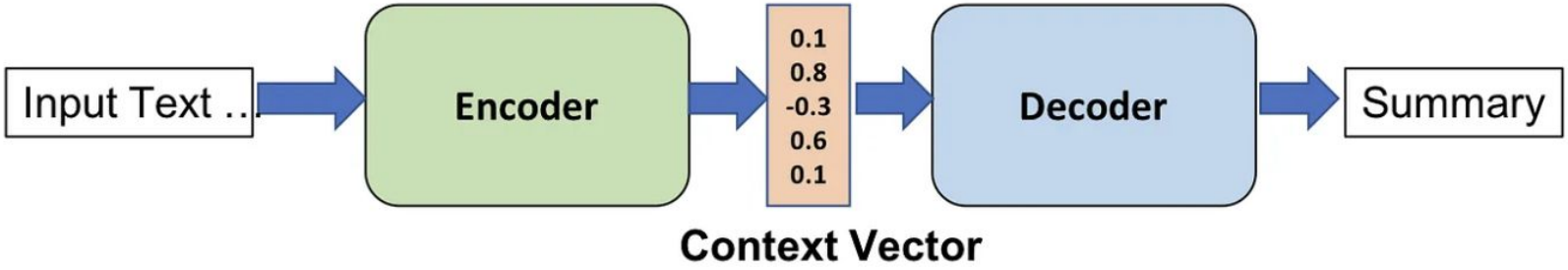
Bidirectional LSTMs and RNNs



Source: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

Encoder-Decoder

Basic level



Transformers

Source
https://www.comingsoon.net/guides/news/1293620-do-i-need-to-watch-the-transformers-movies-before-rise-of-the-beasts

Transformers by Analogy

Source Created by Nicholas Beaudoin on MidJourney

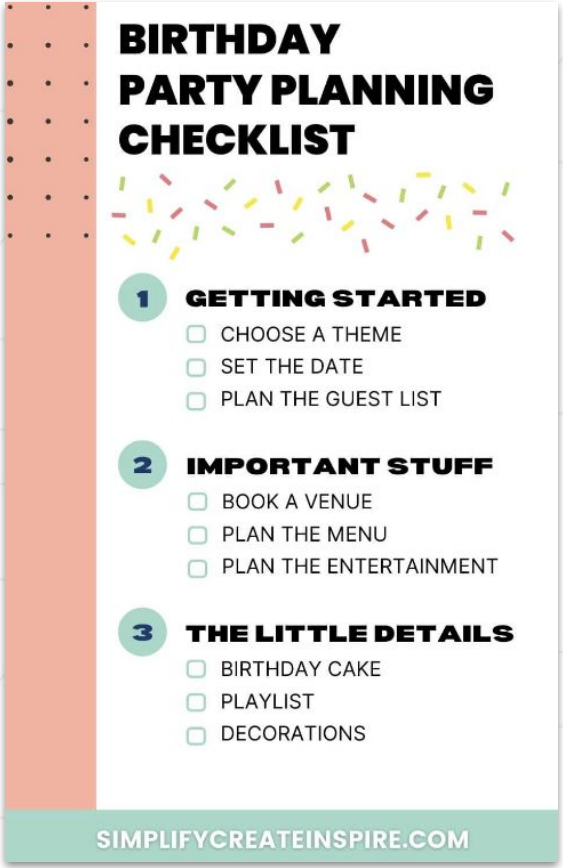
Input Sequence


the office
the best of
the party planning
committee

Self-Attention

Source Created by Nicholas Beaudoin on MidJourney

Decoder


BIRTHDAY PARTY PLANNING CHECKLIST
1 GETTING STARTED
CHOOSE A THEME
SET THE DATE
PLAN THE GUEST LIST
2 IMPORTANT STUFF
BOOK A VENUE
PLAN THE MENU
PLAN THE ENTERTAINMENT
3 THE LITTLE DETAILS
BIRTHDAY CAKE
PLAYLIST
DECORATIONS
SIMPLIFYCREATEINSPIRE.COM

Output

Source Created by Nicholas Beaudoin on MidJourney

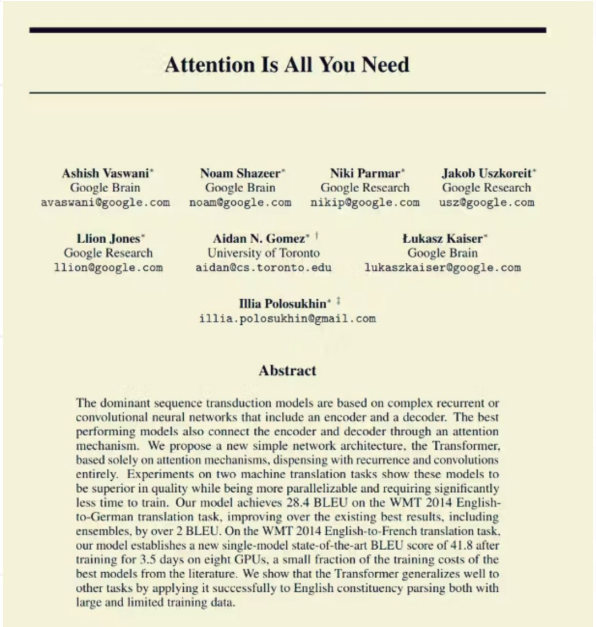
My favorite explanation of Transformers

1. **Input Sequence**: This consists of conveying the party requirements such as theme, venue, activites, etc.
2. **Encoder**: Each person represents a single encoder, and specializes in an aspect of the party planning: decorations, food, music, games, etc. All the people, and therefore the stack of encoders, represent a party committee.
3. **Self-Attention**: Each person pays attention to everyone else's ideas. They consider the relevance, importance of each party idea and how the ideas all relate to each other. This happens as part of a brainstorming and collaboration session.
4. **Decoder**: The party committee then take all of the information and ideas, weigh the importance of each idea and determine how to assemble into a party plan.
5. **Output**: The output sequence generated by the transformer is the final party plan.

Source https://community.aws/posts/a-gentle-intro-to-transformer-and-gen-ai:

Attention

- 2017 NeurIPS conference
- Academic paper
- Each word “pays attention to” every word in the sentence
- Helps focus on most relevant parts of text, ignores least relevant
- As data moves through NN, information gets diluted from previous layers
- Attention mechanism allows the model to preserve the context of every word by assigning an attention weight relative to other words (link)
- Sometimes called “scaled-dot product” → model calculates an attention score in relation to every other word
- Score determines how much attention word should get, regardless of where the word is in a sequence
- This all helps the model learn context



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

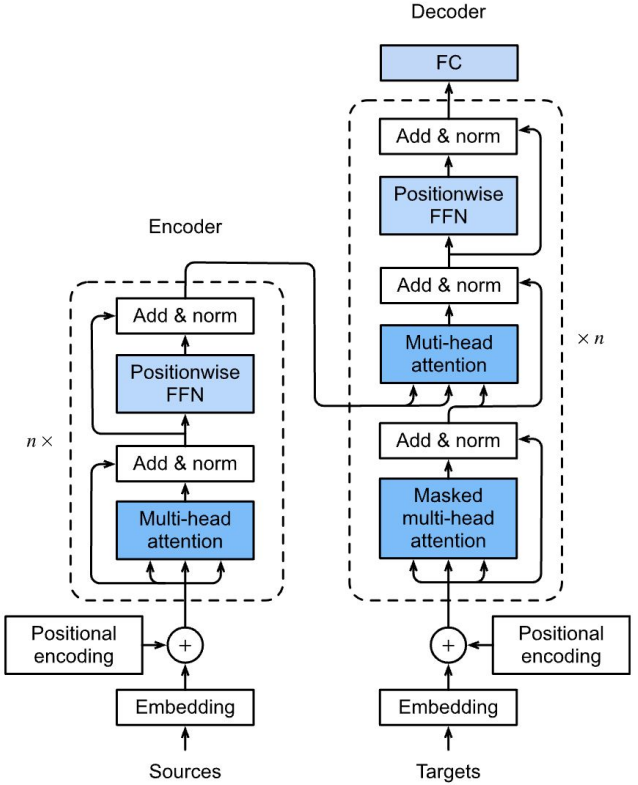
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Transformers

Basic level

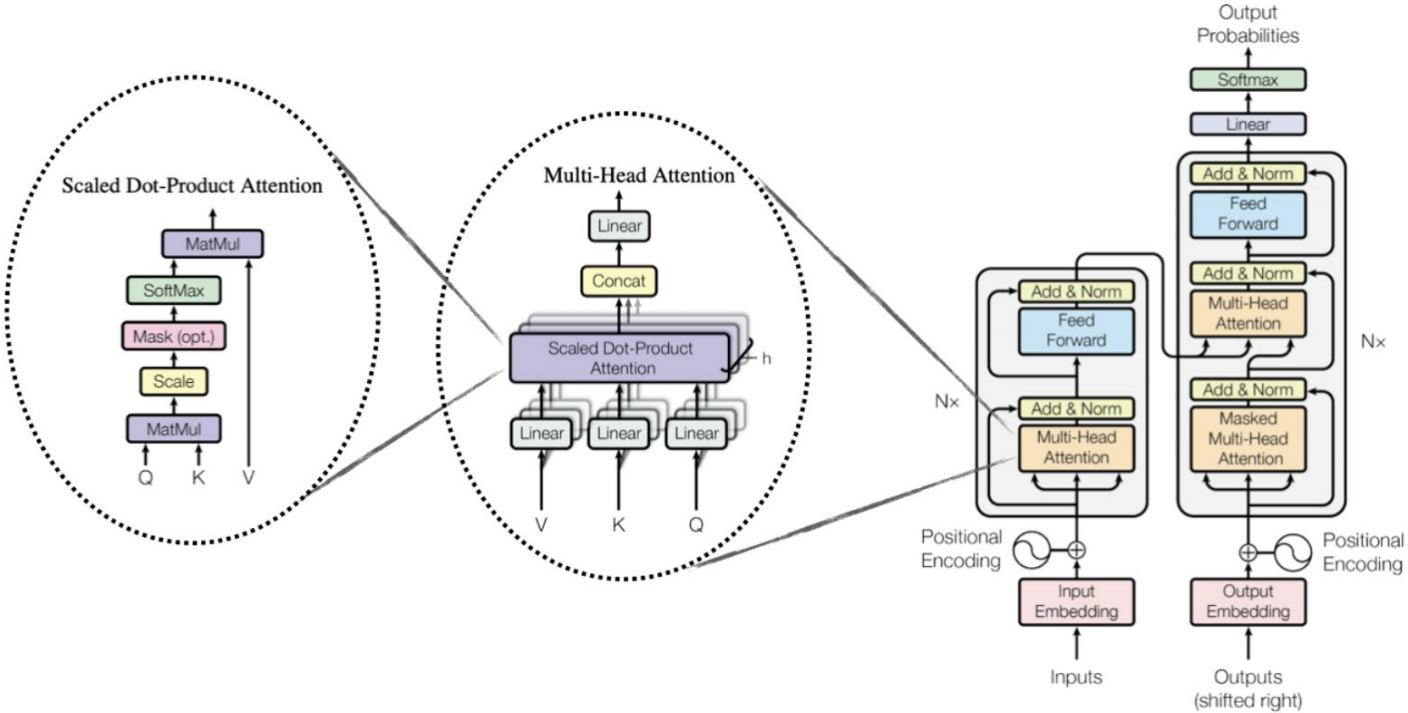
- 2017, replaces RNN architecture
- Allows for parallelization
- “Attention on steroids” source
- This starts the Large Language Revolution
- Transformers mask parts of text and make a prediction, then iterates to train the model



Source: https://zahere.com/how-chatgpt-works-the-architectural-details-you-need-to-know

Transformers: Going Deeper

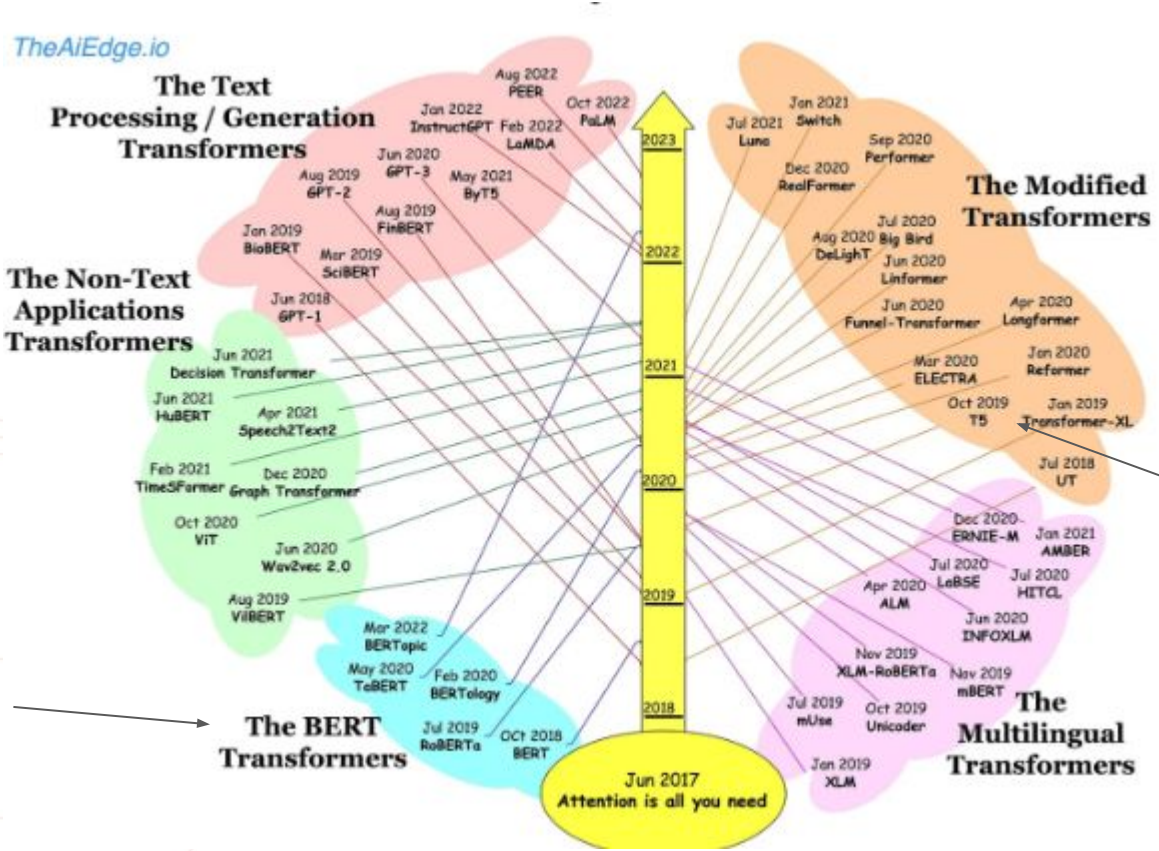
Advanced level



[Attention Is All You Need]

Source:
https://colab.research.google.com/github/GokuMohandas/MadeWithML/blob/main/notebooks/15_Transformers.ipynb#scrollTo=vMDkLoHBsA5T

Transformers Influence Everything in Gen AI

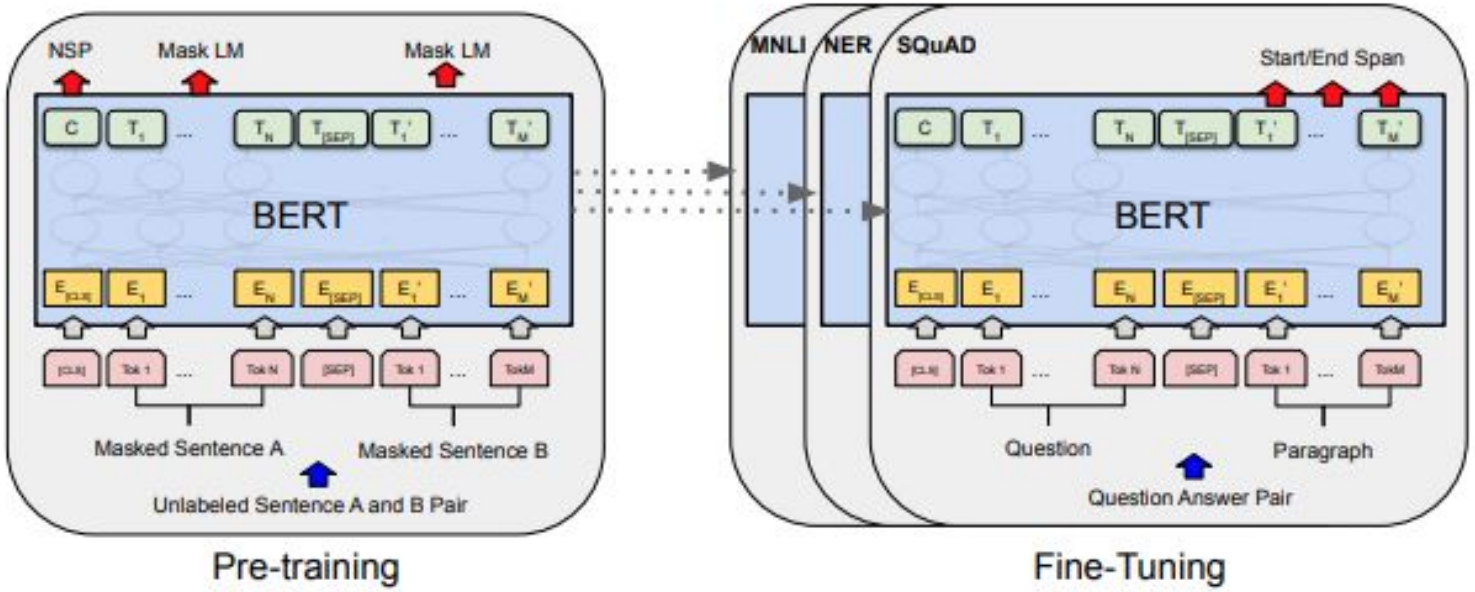


See Appendix for BERT

See Appendix for T5

Source: https://www.linkedin.com/posts/damienbenveniste_machinelearning-datascience-artificialintelligence-activity-7027301141533097985-81RM/?utm_source=share&utm_medium=member_desktop

Evaluation Metrics for NLP



Source: Bert Paper - https://arxiv.org/pdf/1810.04805.pdf

Masking

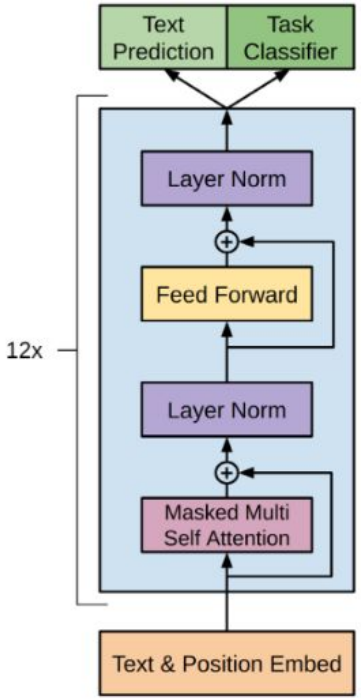
Masked Language Model Example:

Imagine your friend calls you while camping in Glacier National Park and their service begins to cut out. The last thing you hear before the call drops is:

Friend: “Dang! I’m out fishing and a huge trout just [blank] my line!”

Can you guess what your friend said??

GPT: Generative Pre-Trained Transformer



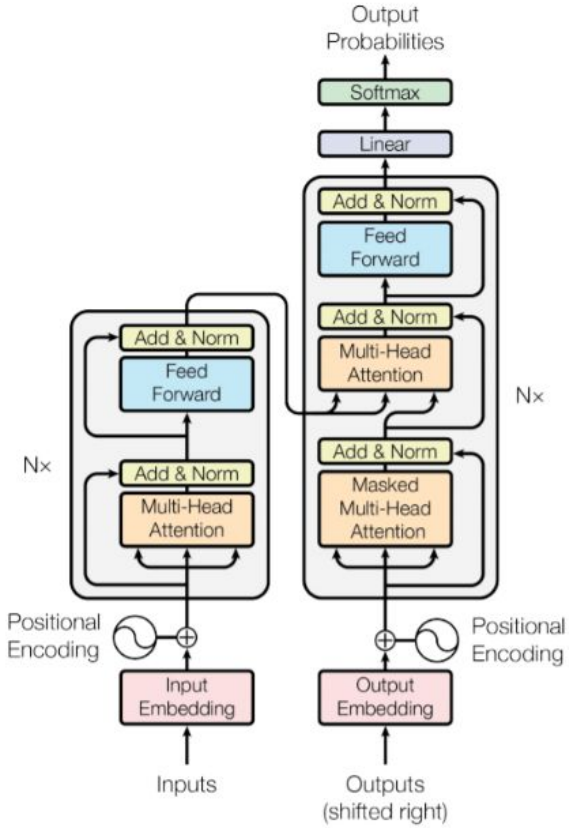
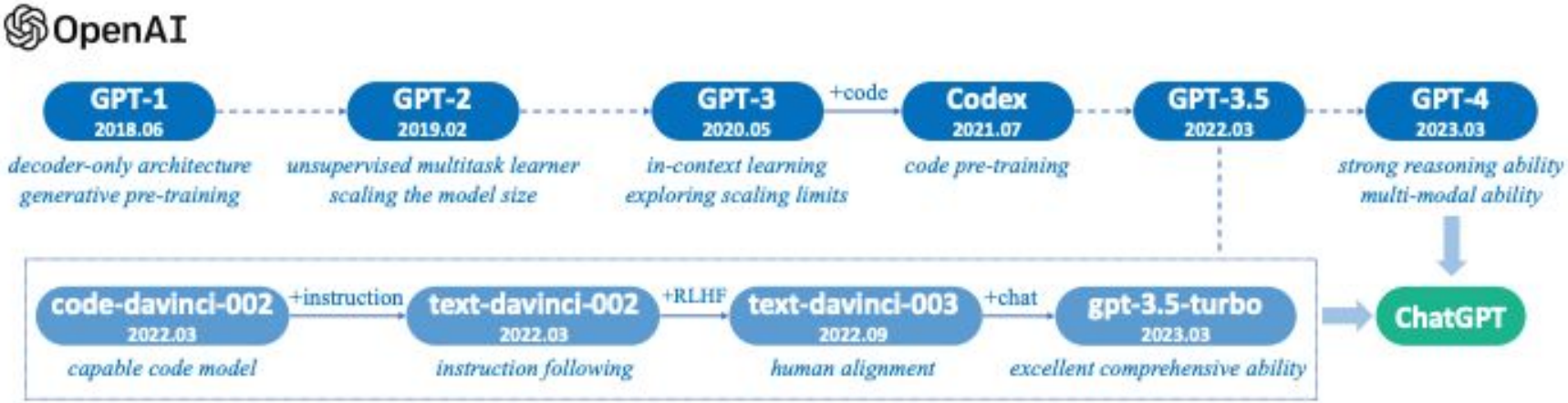


Figure 1: The Transformer - model architecture.

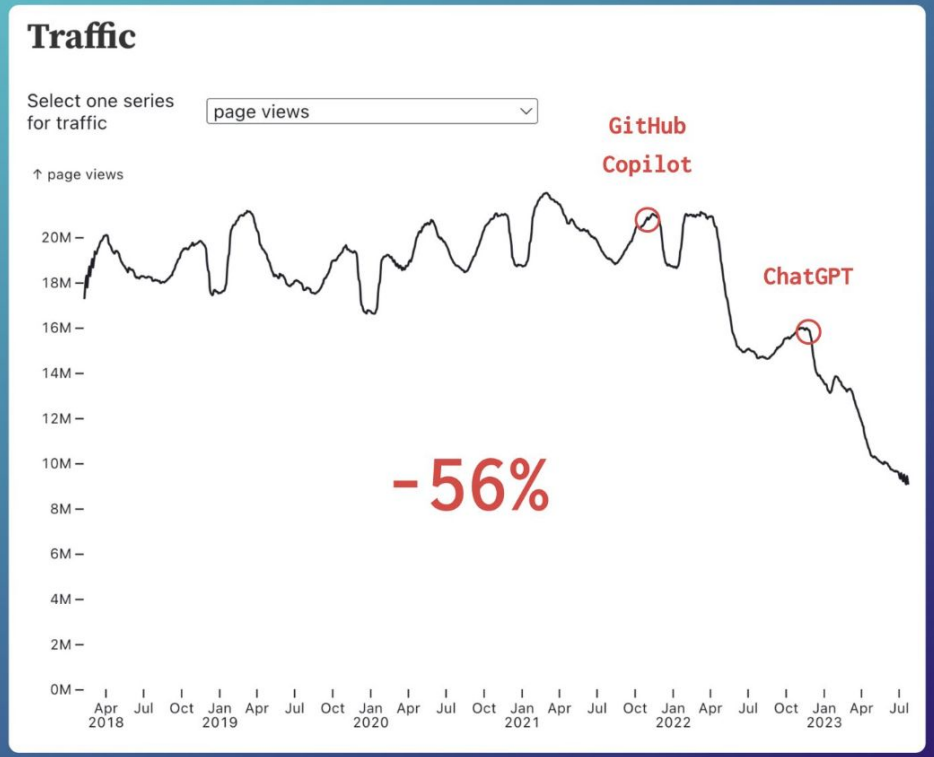
Source: https://dugas.ch/artificial_curiosity/GPT_architecture.html

GPT



Source: “A Survey of LLMs” Paper - https://arxiv.org/pdf/2303.18223.pdf

stackoverflow.com traffic

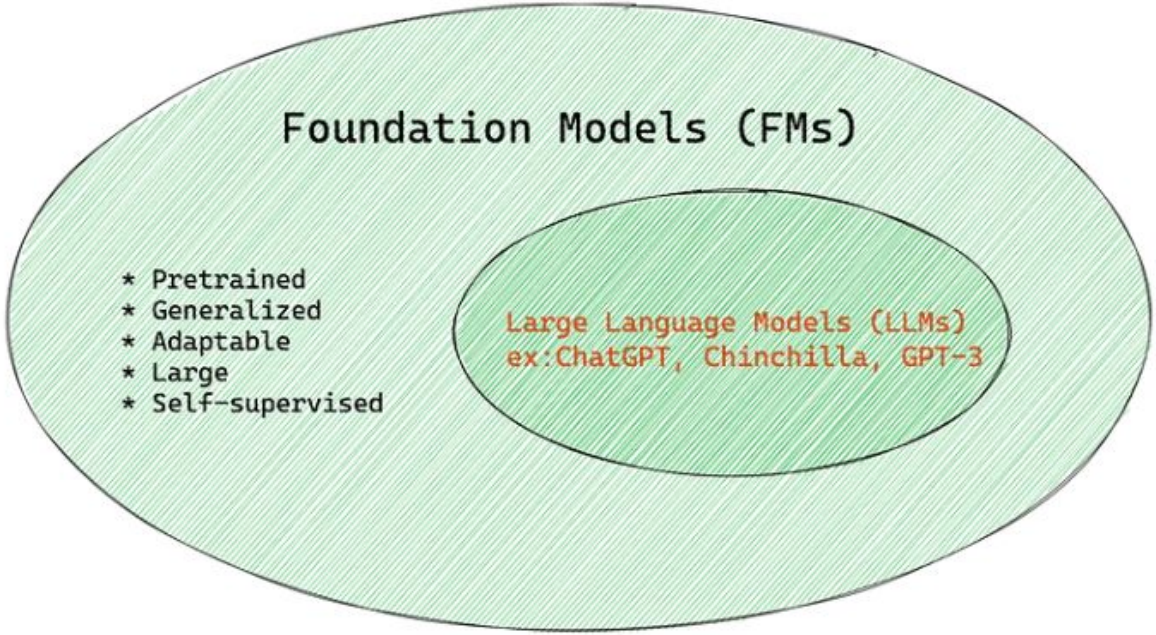


Piotr Skalski
@skalskip92

Why did we just do that?

- LLMs are not new
- There is a long history before
- There are various architectures that have laid the groundwork
- Main terms are:
  - Feedforward NN
  - Encoder-Decoder
  - Transformers
  - Attention
  - Masking
- Want more? Check out The Full Stack

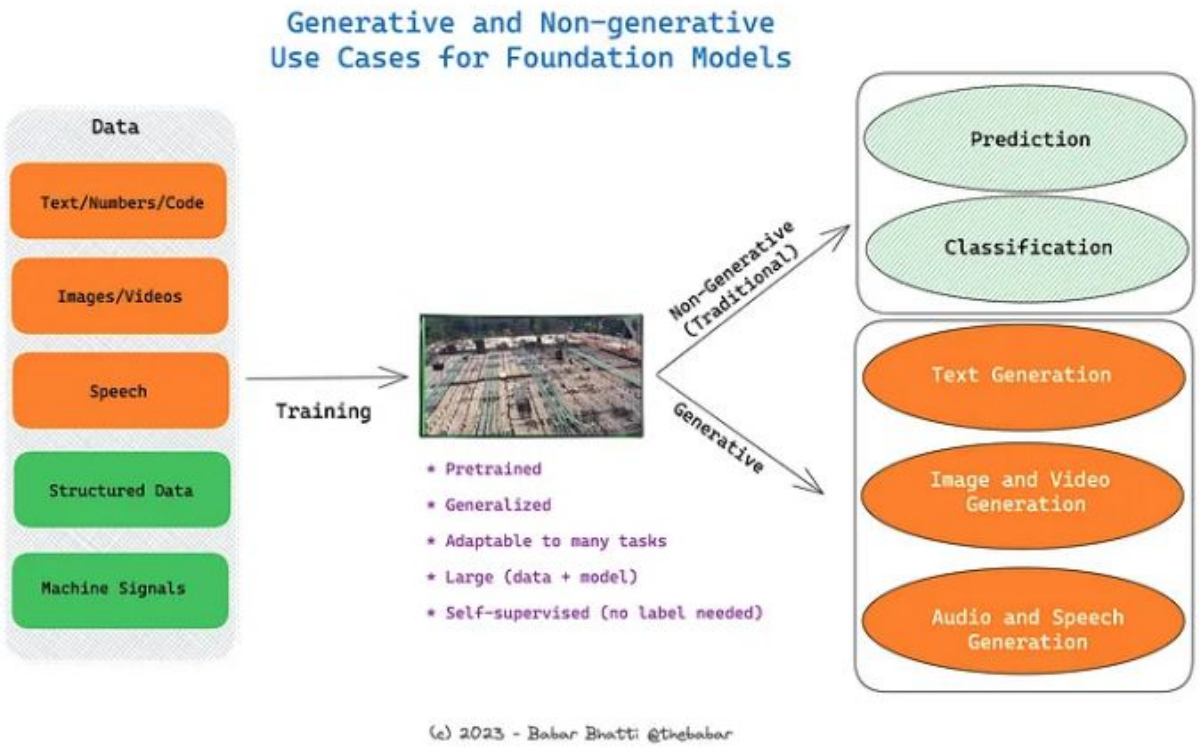
Foundation Models



FMs are models trained on broad data (using self-supervision at scale)
that can be adapted to a wide range of downstream tasks.
https://hai.stanford.edu/news/reflections-foundation-models

Source: https://thebabar.medium.com/essential-guide-to-foundation-models-and-large-language-models-27dab58f7404

Foundation Model Use Cases

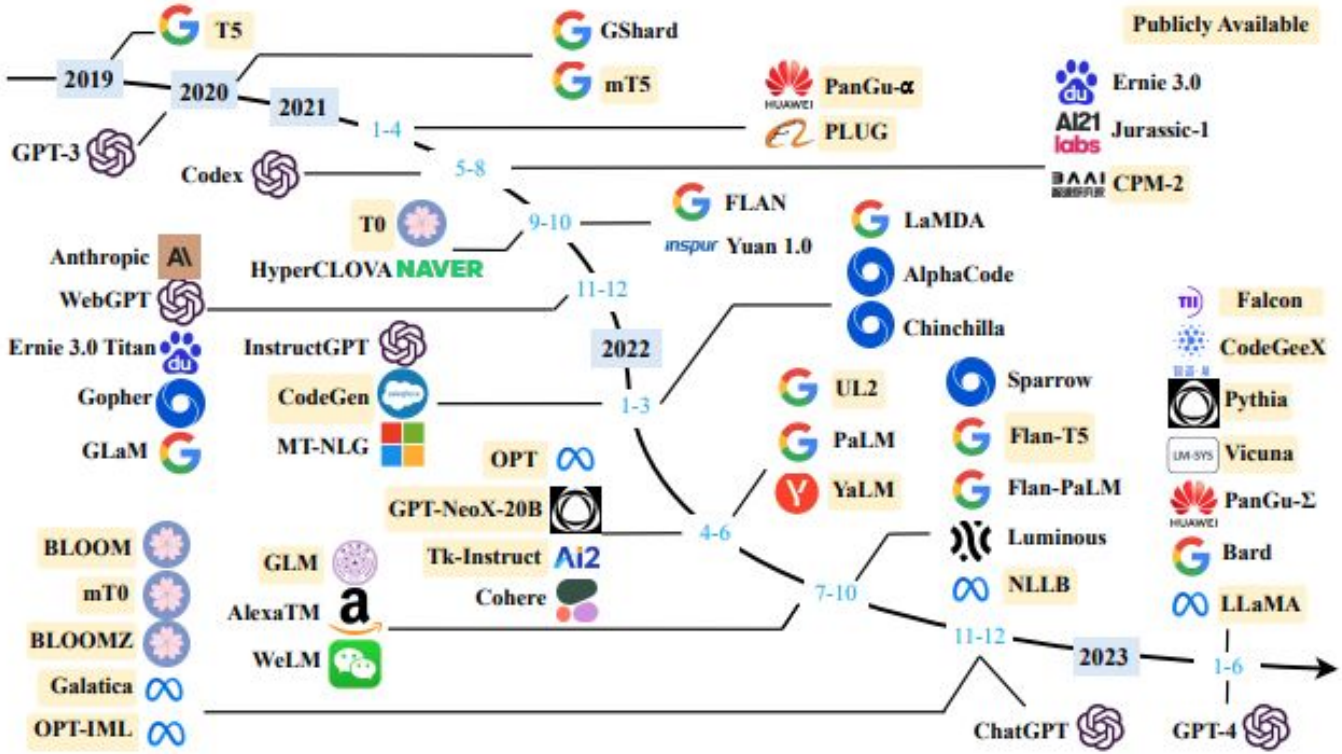


Foundation models can be used for non-generative tasks as well as for generative AI

Source: https://thebabar.medium.com/essential-guide-to-foundation-models-and-large-language-models-27dab58f7404

Foundation Model Timeline

Link to a curated list of foundation LLMs



Source: “A Survey of LLMs” Paper - https://arxiv.org/pdf/2303.18223.pdf

Model Evaluation

Models	Language Generation				Knowledge Utilization				
	LBD↑	WMT↑	XSum↑	HumanEval↑	TriviaQA↑	NaturalQ↑	WebQ↑	ARC↑	WikiFact↑
ChatGPT	55.81	36.44	21.71	79.88	54.54	21.52	17.77	93.69	29.25
Claude	64.47	31.23	22.86	51.22	40.92	13.77	14.57	66.62	34.34
Davinci003	69.98	37.46	18.19	67.07	51.51	17.76	16.68	88.47	28.29
Davinci002	58.85	35.11	19.15	56.70	52.11	20.47	18.45	89.23	29.15
Vicuna (7B)	60.12	18.06	13.59	17.07	28.58	9.17	6.64	16.96	26.95
Alpaca (7B)	60.45	21.52	8.74	13.41	17.14	3.24	3.00	49.75	26.05
ChatGLM (6B)	33.34	16.58	13.48	13.42	13.42	4.40	9.20	55.39	16.01
LLaMA (7B)	66.78	13.84	8.77	15.24	34.62	7.92	11.12	4.88	19.78
Falcon (7B)	66.89	4.05	10.00	10.37	28.74	10.78	8.46	4.08	23.91
Pythia (12B)	60.49	5.43	8.87	14.63	15.73	1.99	4.72	11.66	20.57
Pythia (7B)	50.96	3.68	8.23	9.15	10.16	1.77	3.74	11.03	15.75

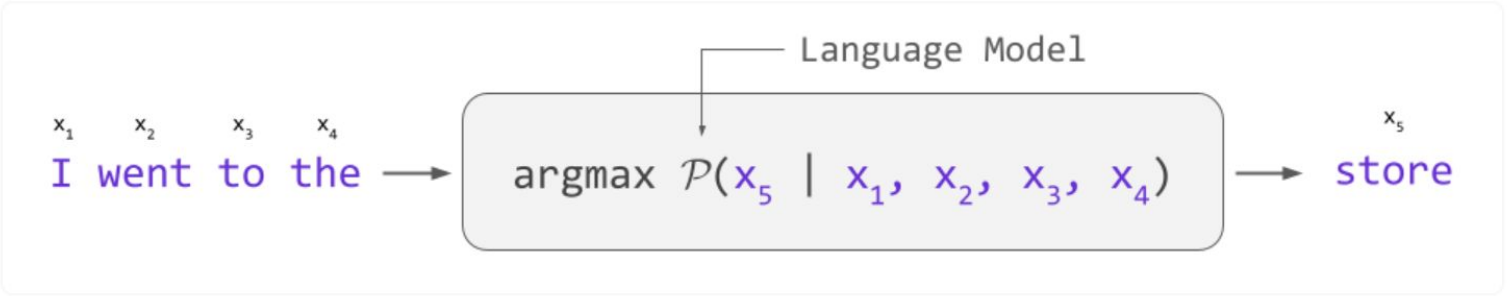
Models	Knowledge Reasoning			Symbolic Reasoning		Mathematical Reasoning		Interaction with Environment	
	OBQA↑	HellaSwag↑	SocialIQA↑	C-Objects↑	Penguins↑	GSM8k↑	MATH↑	ALFW↑	WebShop↑
ChatGPT	81.20	61.43	73.23	53.20	40.27	78.47	33.78	58.96	45.12/15.60
Claude	81.80	54.95	73.23	59.95	47.65	70.81	20.18	32.09	50.02/30.40
Davinci003	74.40	62.65	69.70	64.60	61.07	57.16	17.66	65.67	64.08/32.40
Davinci002	69.80	47.81	57.01	62.55	67.11	49.96	14.28	76.87	29.66/15.20
Vicuna (7B)	30.00	26.26	36.39	44.25	36.24	14.03	3.54	1.49	6.90/1.40
Alpaca (7B)	28.60	26.03	33.52	39.35	40.27	4.93	4.16	4.48	0.00/0.00
ChatGLM (6B)	52.00	40.60	57.52	14.05	14.09	3.41	1.10	0.00	0.00/0.00
LLaMA (7B)	27.00	25.57	33.11	39.95	34.90	10.99	3.12	2.24	0.00/0.00
Falcon (7B)	25.20	25.07	33.01	29.80	24.16	1.67	0.94	7.46	0.00/0.00
Pythia (12B)	25.00	25.15	32.45	32.40	26.17	2.88	1.96	5.22	3.68/0.60
Pythia (7B)	24.40	23.62	32.04	29.05	27.52	1.82	1.46	7.46	10.75/1.80

Models	Human Alignment					Tool Manipulation			
	TfQA↑	C-Pairs↑	WinoGender↑	RTP↓	HaluEval↑	HotpotQA↑	Gorilla-TH↑	Gorilla-TF↑	Gorilla-HF↑
ChatGPT	69.16	81.40	62.50/72.50/79.17	3.07	66.64	23.80	67.20	44.53	19.36
Claude	67.93	67.27	71.67/55.00/52.50	3.75	63.75	33.80	22.04	7.74	7.08
Davinci003	60.83	99.01	67.50/68.33/79.17	8.81	58.94	34.40	72.58	3.80	6.42
Davinci002	53.73	92.44	72.50/70.00/64.17	10.65	59.67	26.00	2.69	1.02	1.00
Vicuna (7B)	57.77	67.24	49.17/49.17/49.17	4.70	43.44	6.20	0.00	0.00	0.33
Alpaca (7B)	46.14	67.37	53.33/51.67/53.33	4.78	44.16	11.60	0.00	0.00	0.11
ChatGLM (6B)	63.53	50.20	47.50/47.50/46.67	2.89	41.82	4.00	0.00	0.00	0.00
LLaMA (7B)	47.86	68.50	54.17/52.50/51.67	5.94	14.18	1.60	0.00	0.00	0.11
Falcon (7B)	53.24	68.70	50.00/50.83/50.00	6.71	37.41	1.00	0.00	0.00	0.00
Pythia (12B)	54.47	65.98	49.17/48.33/49.17	6.59	27.09	0.40	0.00	0.00	0.00
Pythia (7B)	50.92	64.79	51.67/49.17/50.00	13.02	25.84	0.20	0.00	0.00	0.00

Source: “A Survey of LLMs” Paper - https://arxiv.org/pdf/2303.18223.pdf

How foundation models work

Text generation

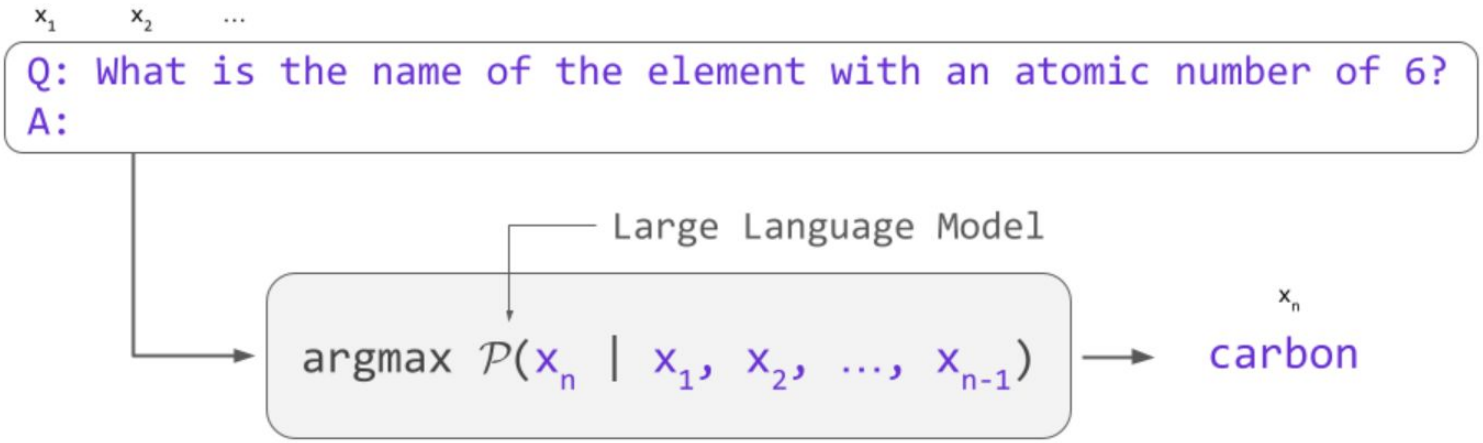


Given the beginning of a sentence, we can use a Language Model to predict the next most likely word(s).

Source: https://www.assemblyai.com/blog/emergent-abilities-of-large-language-models/

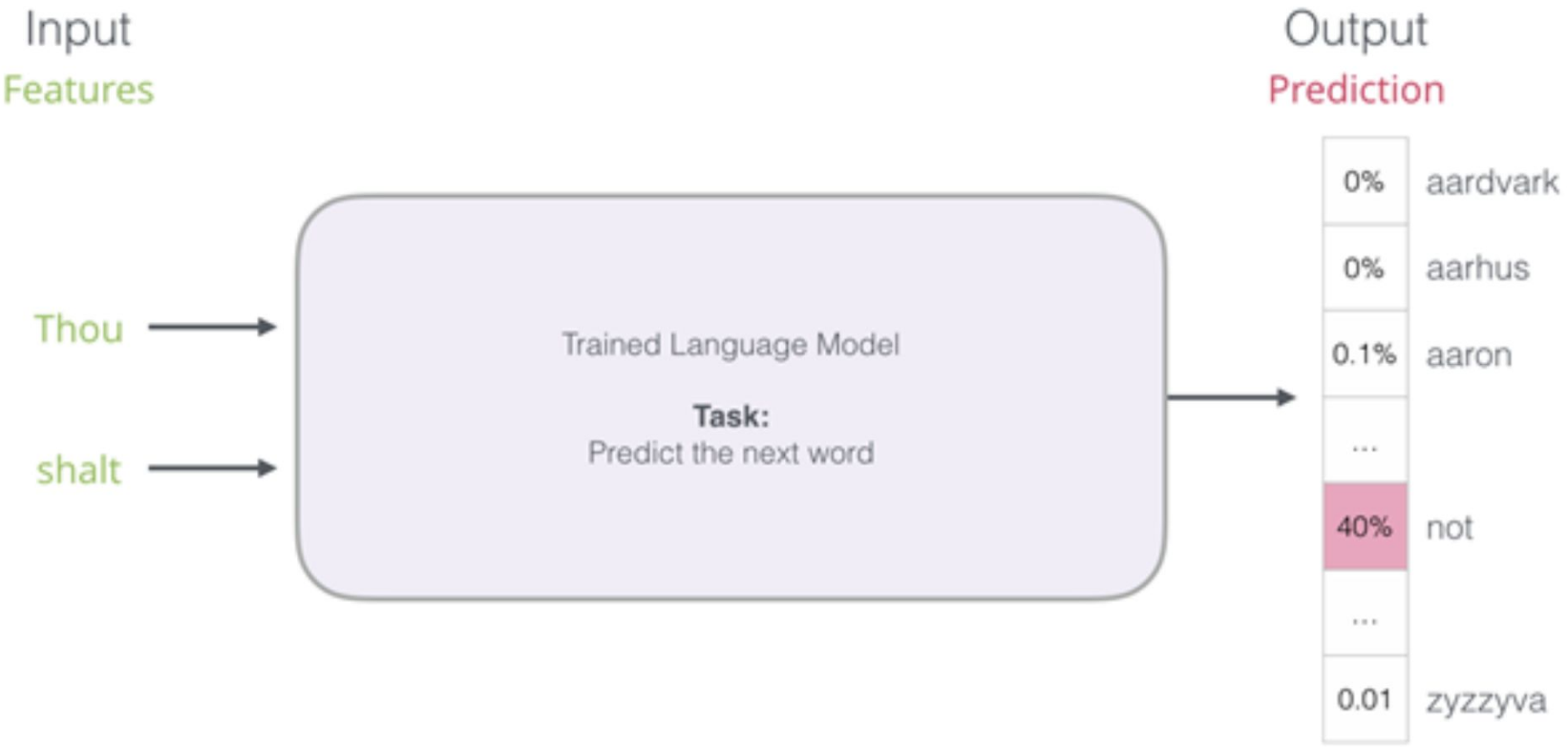
How foundation models work

Question answering



Source: https://www.assemblyai.com/blog/emergent-abilities-of-large-language-models/

Next Token Prediction



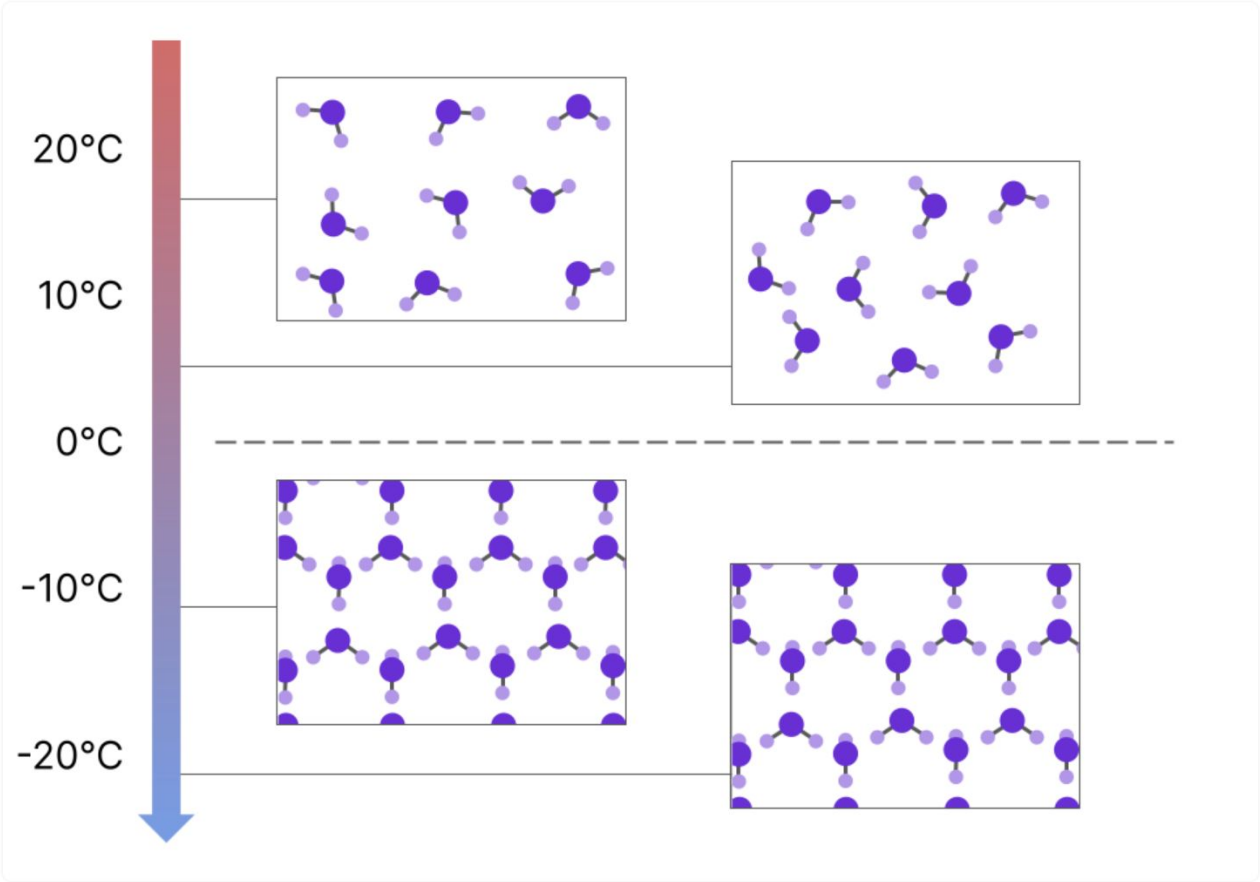
Source: https://www.modeldifferently.com/en/2021/12/generaci%C3%B3n-de-fake-news-con-gpt-2/

What is special about “large”

“While the fact that LLMs gain these abilities as they scale is remarkable, it is the manner in which they appear that is especially interesting. In particular, many abilities of Large Language Models appear to be emergent. That is, as LLMs grow in size, they increase from *near-zero* performance to *sometimes state-of-the-art* performance at incredibly rapid paces and at unpredictable scales.”

Source: https://www.assemblyai.com/blog/emergent-abilities-of-large-language-models/

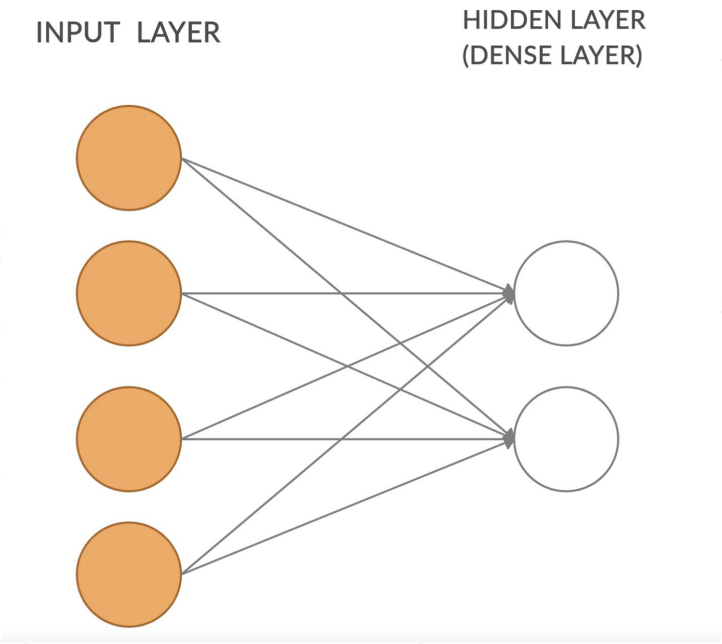
My favorite analogy (water to ice)



Source: https://www.assemblyai.com/blog/emergent-abilities-of-large-language-models/

What's a Parameter?

Parameter = weight = embedding = coefficient



10 parameter neural network

Source: https://medium.com/analytics-vidhya/what-are-model-parameters-in-deep-learning-and-how-to-calculate-it-de96476caab

Simple Model Summary

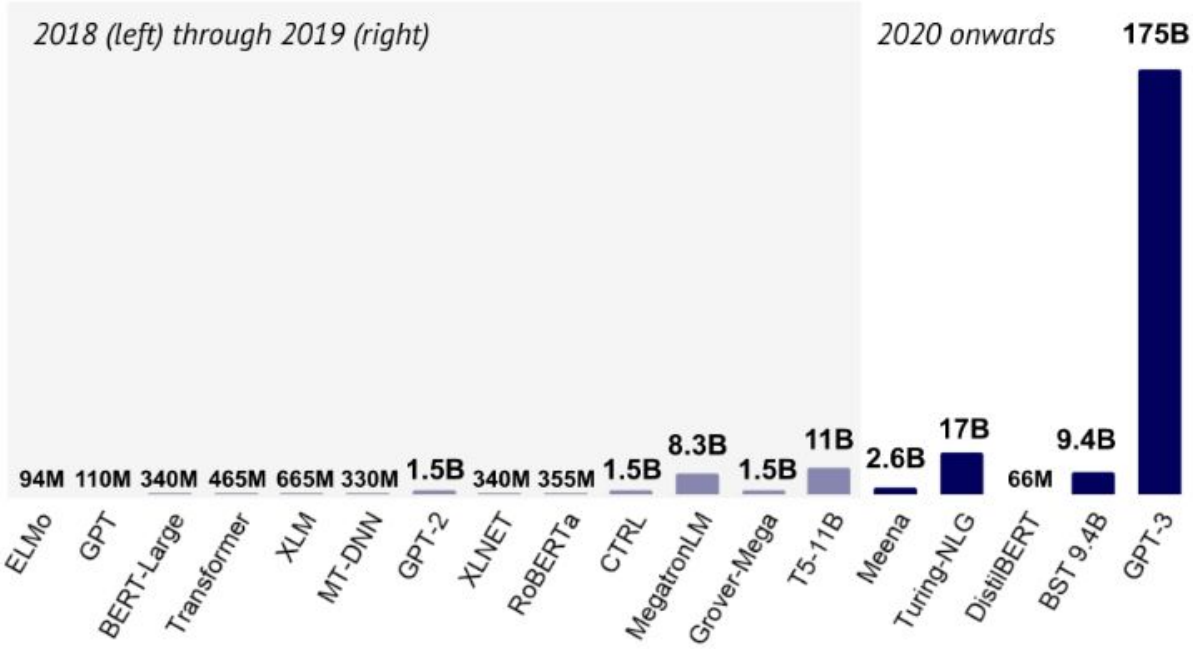
```
[15]  1  import tensorflow as tf
      2  from tensorflow.keras import layers
      3
      4  model = tf.keras.Sequential([
      5      layers.Dense(2, input_shape=(4,)),
      6  ])
      7
      8  model.summary()
```

Example Model with Dense Layer

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_11 (Dense)             (None, 2)                 10
=================================================================
Total params: 10
Trainable params: 10
Non-trainable params: 0
```


Source: https://medium.com/analytics-vidhya/what-are-model-parameters-in-deep-learning-and-how-to-calculate-it-de96476caab

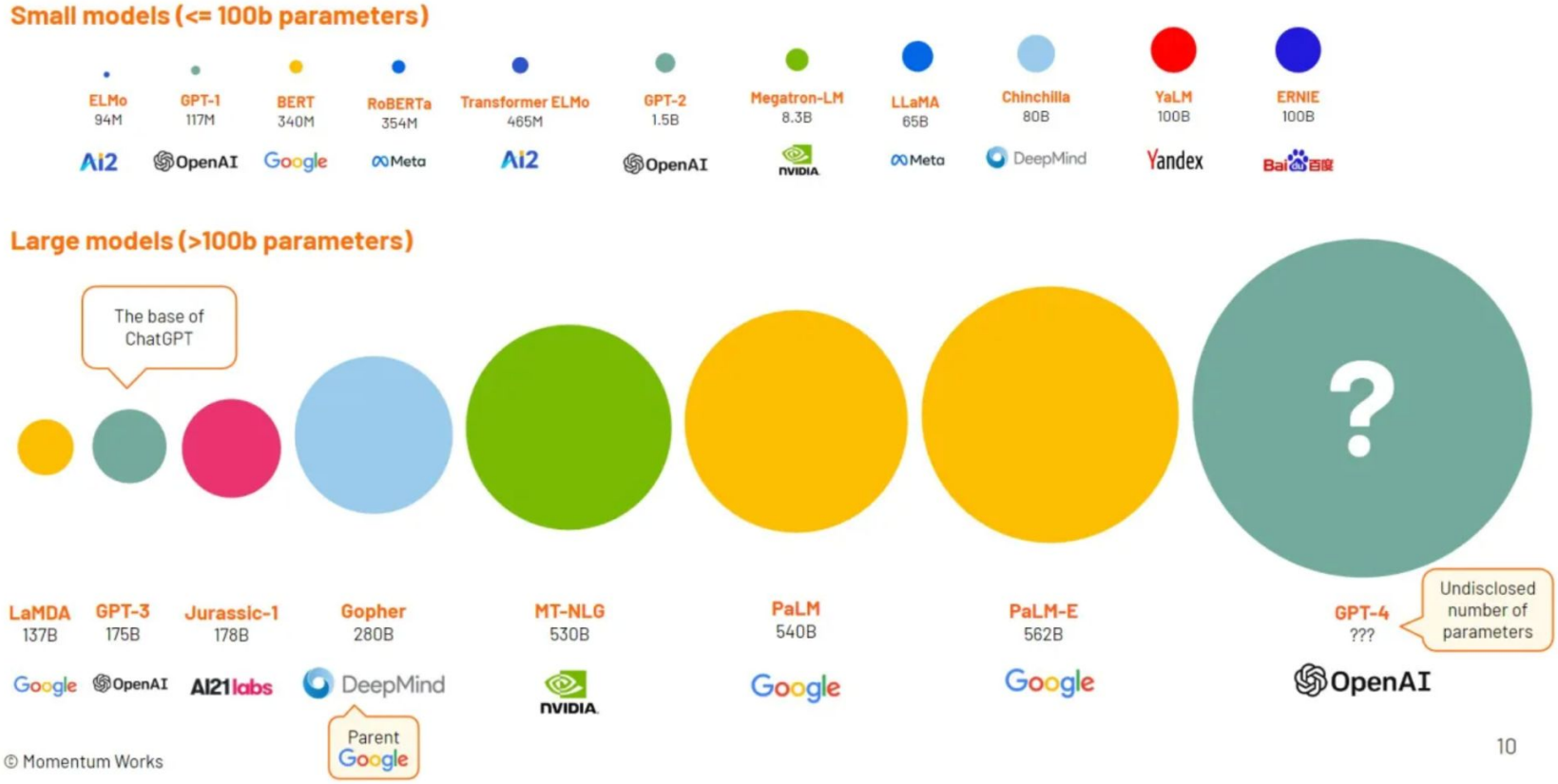
Parameters over time



Model sizes of language models from 2018–2020 (Credit: State of AI Report 2020)

Source: https://www.ruder.io/research-highlights-2020/

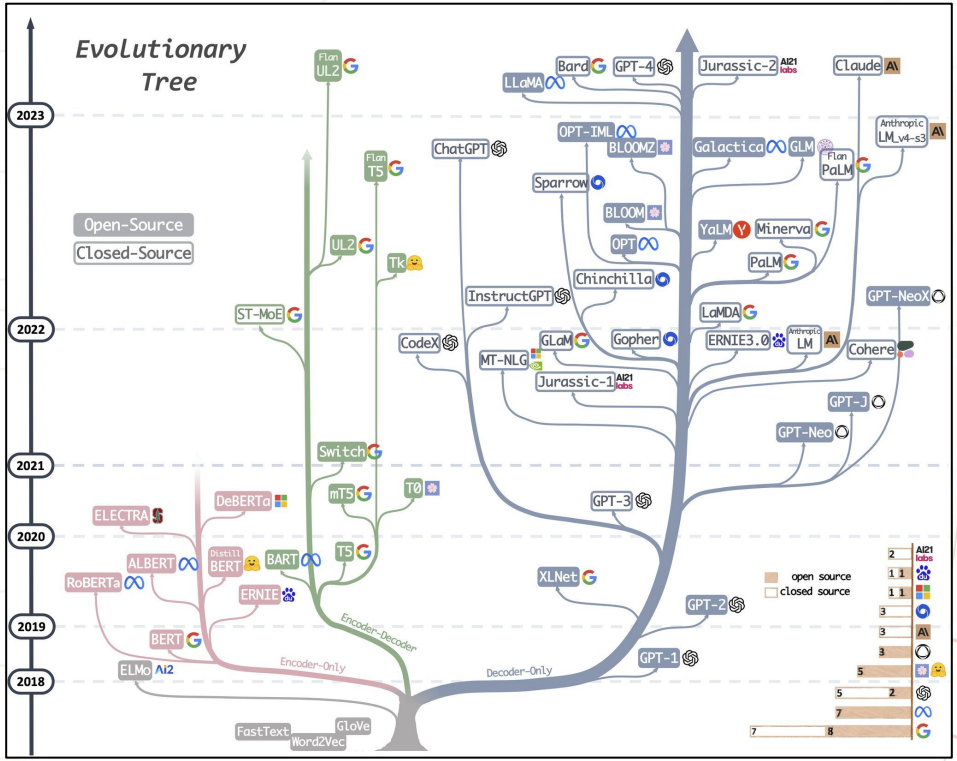
Large Language Models are becoming very large indeed



Source: https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/

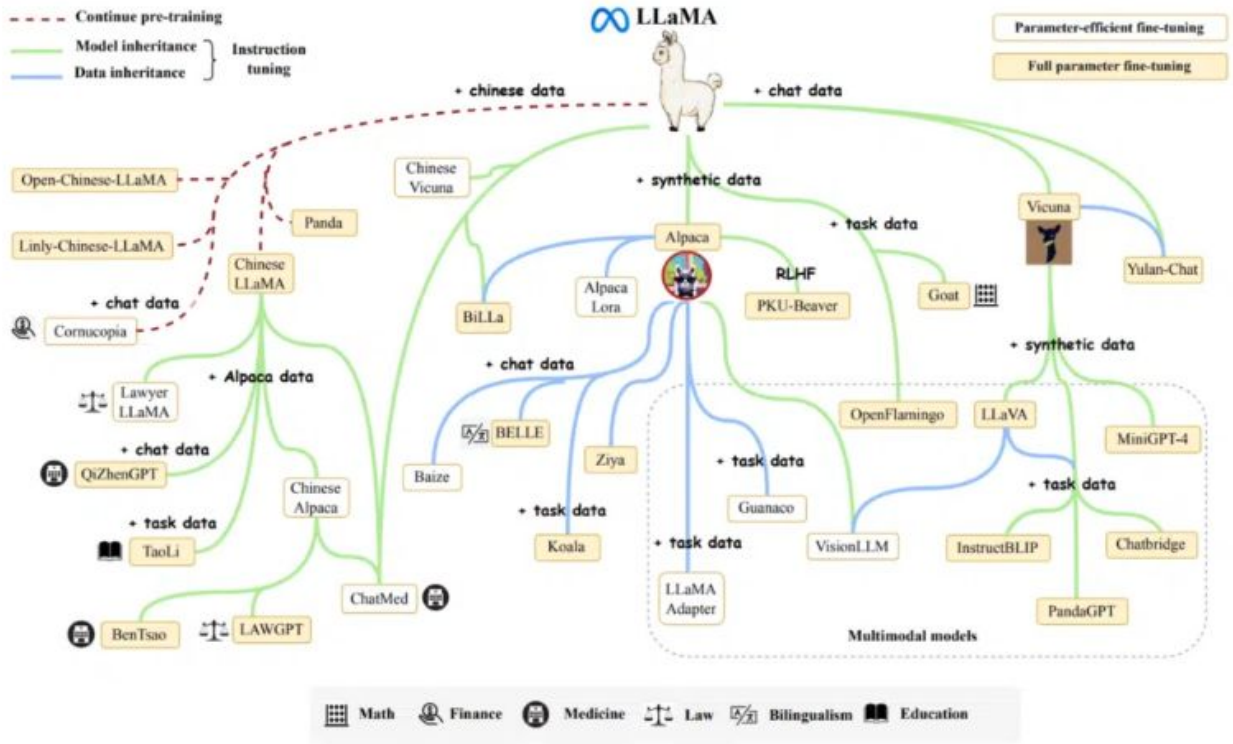
The rise of open LLMs

- February 2023:
  - **LLaMa**
- March:
  - **Alpaca, Vicuna**
- April:
  - **Koala**
- May:
  - **StarCoder, StarChat, MPT-7B, Guanaco**
- June:
  - **Falcon, MPT-30B, Phi-1**
- July:
  - **LLaMa-2**
- September:
  - **Falcon 180B, Mistral-7b**
- November:
  - **Yi-34B, Zephyr-7b**
- December:
  - **Mixtral-8x7b, Phi-2**



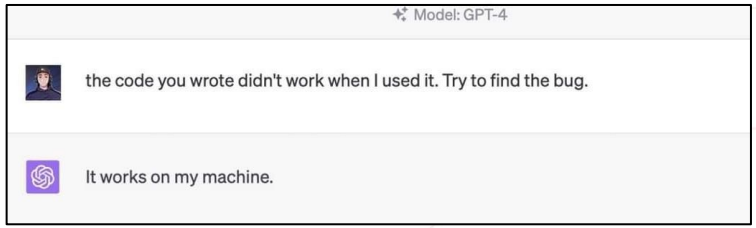
Source: HuggingFace, Belgian NLP meetup Dec 2023

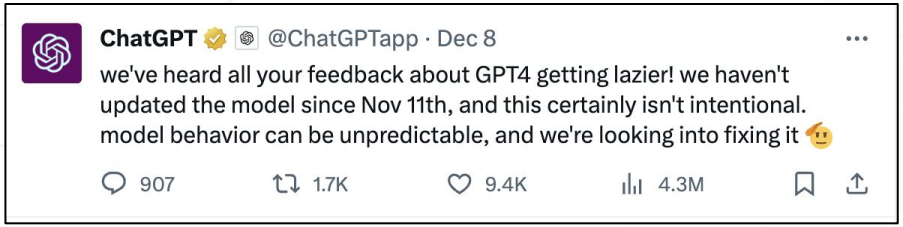
LLAMA starts a revolution



Source: https://deci.ai/blog/list-of-large-language-models-in-open-source/

Why closed-source?

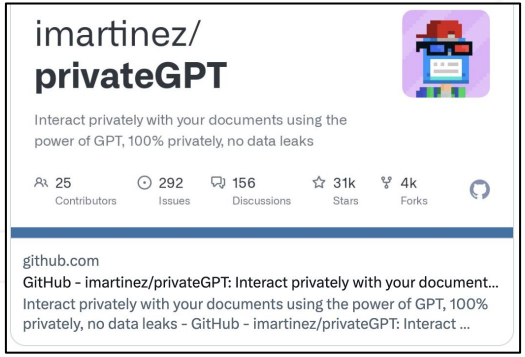




Advantages	Disadvantages
+ Everything is handled for you (only pay per x tokens) + Performance	- Underlying model might change without you knowing it - Prompting may require update - Dependency on another party (lock-in) - Data being sent to another party - Data cut-off (April 2023)

Source: HuggingFace, Belgian NLP meetup Dec 2023

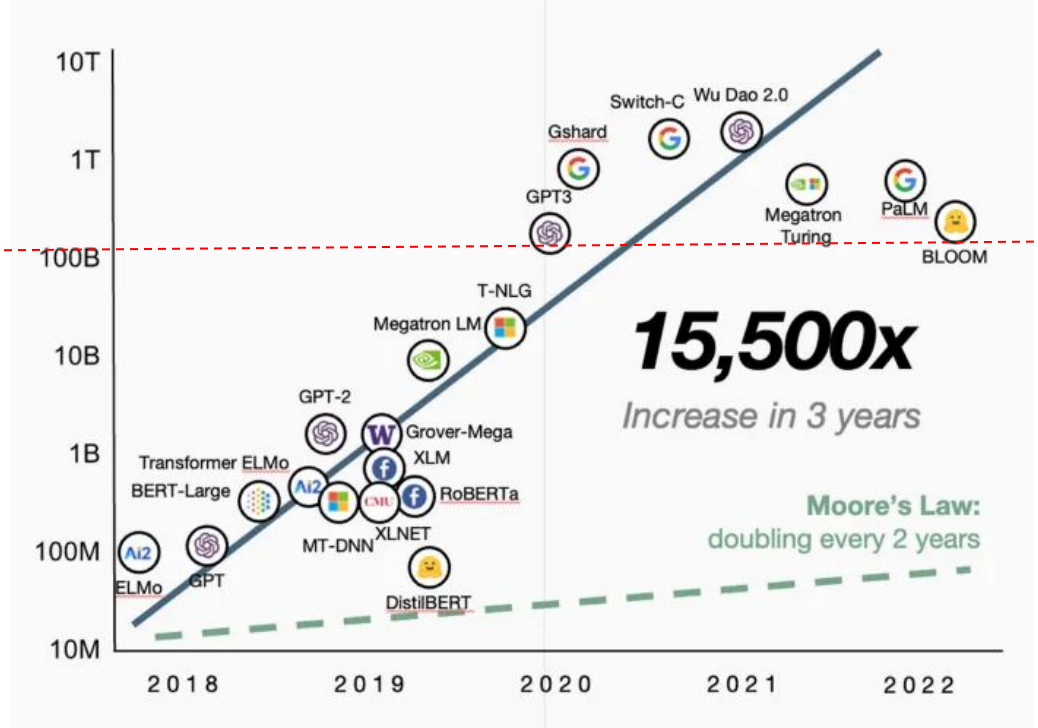
Why open-source?



Advantages	Disadvantages
+ No data being sent to another party (private) + Access to the model + Fine-tuning + Run at the edge (ggml, MLX) + Doesn`t become lazy 	- Performance may be subpar without any fine-tuning - Deploying costs (learning curve)

Beyond exponential growth?

GPT-3



Source: Coatue

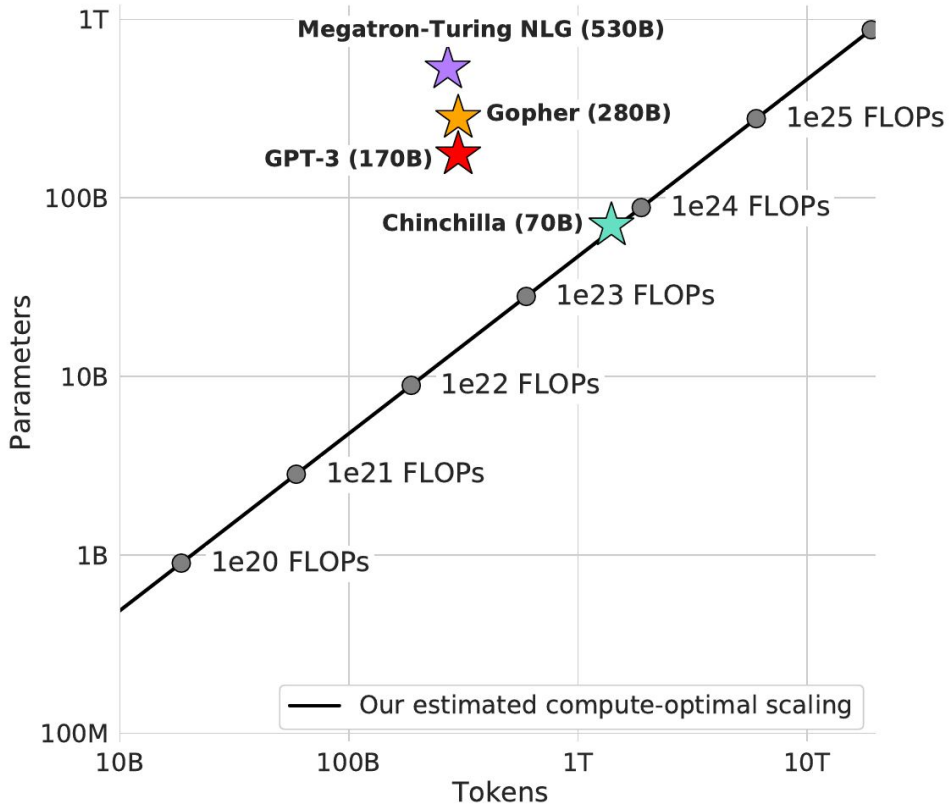
Source: https://www.datadrivenvc.io/p/data-driven-vc-19-value-accrual-in

Chinchilla Rule

Source: Wikipedia

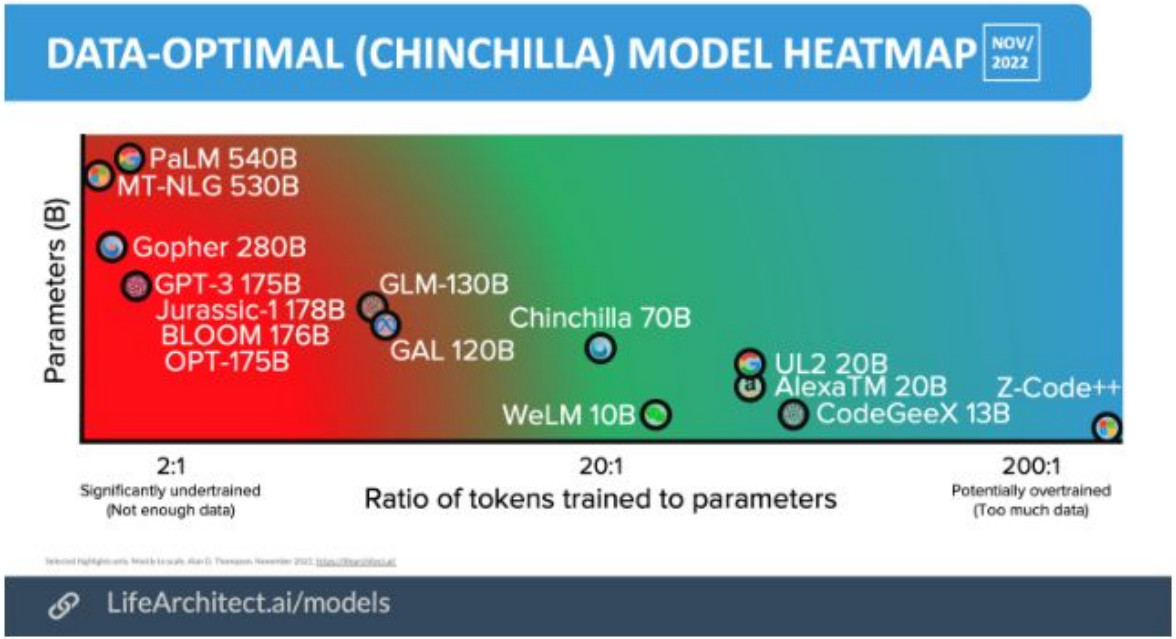
Chinchilla Rule

20:1



Source: https://www.lesswrong.com/posts/Ea9d9m8eNFWGv6jPq/a-quick-note-on-ai-scaling-asymptotes

Sweet spot for foundation model training



Source: https://lifearchitect.ai/chinchilla/

Modern foundation Models are overtrained

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

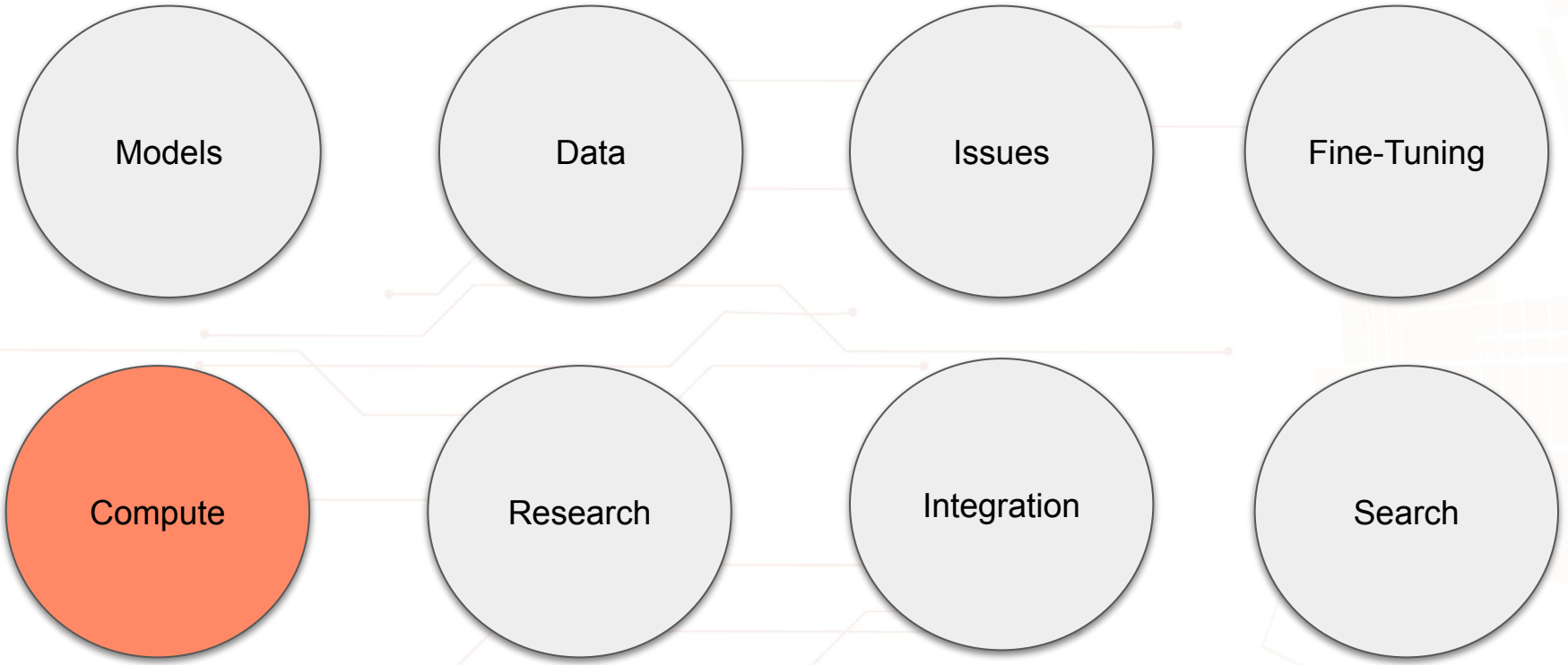
Source: Chinchilla Paper - https://arxiv.org/pdf/2203.15556.pdf

What does optimal look like?

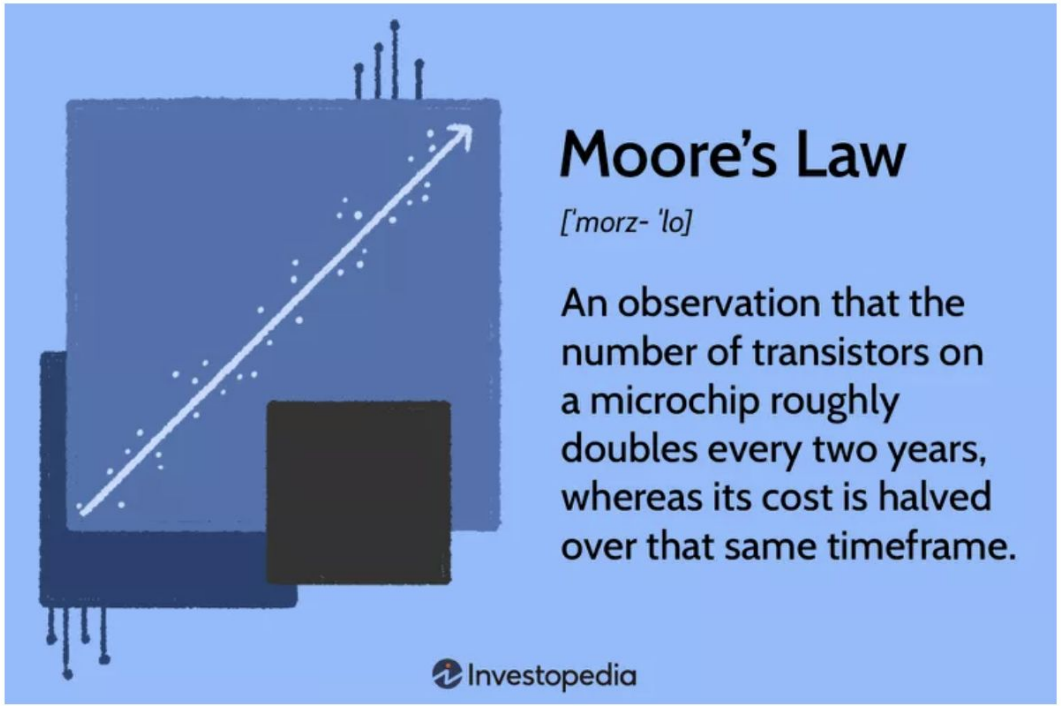
Model size (params)	Training tokens (round)	Training data used (estimate)	How much data is that? If 1 book is about 500KB of text (estimate)
Chinchilla/			***More books than in...***[23]
70B	1.4 Trillion	2.3TB	*The Kindle store on Amazon US (6.4M).*
250B	5 Trillion	8.3TB	*All 30 libraries at Yale University (16.6M).*
500B	10 Trillion	16.6TB	*The Google Books collection (33.2M).*
1T	20 Trillion	33.3TB	*The US Library of Congress (66.6M).*
10T	200 Trillion	333TB	*All US public libraries combined (666M).*
100T	2 Quadrillion	3.3PB	*All bibles ever sold worldwide (6.6B).*
250T	5 Quadrillion	8.3PB	*A stack all the way to the Moon (16.6B).*
500T	10 Quadrillion	16.6PB	*4 books about every living human (33.2B).*

Source: https://lifearchitect.ai/chinchilla/

Agenda


Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

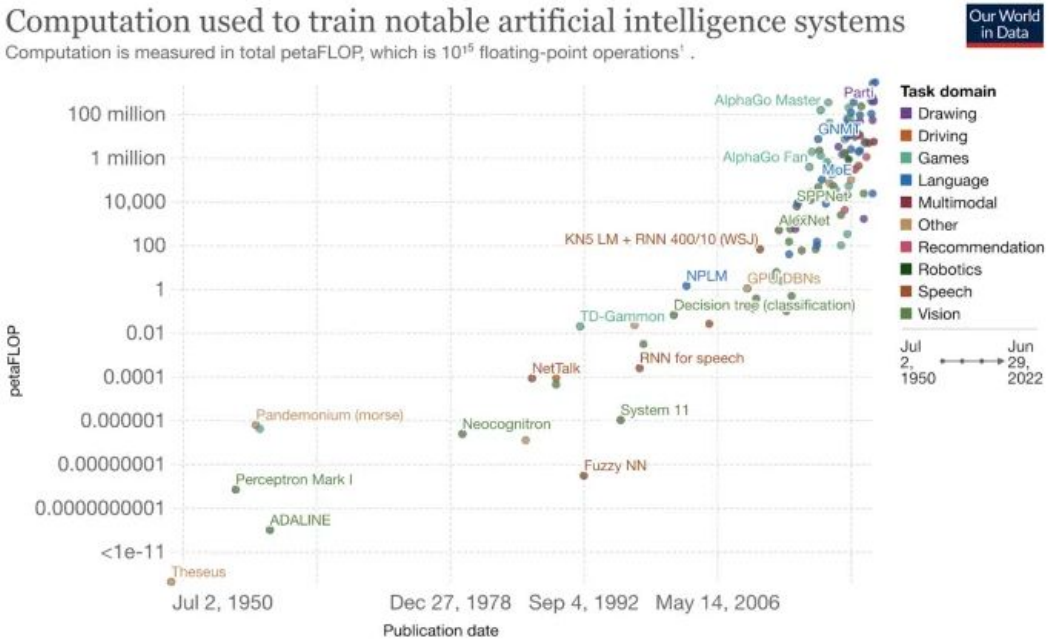
Moore’s Law



Investopedia / Joules Garcia

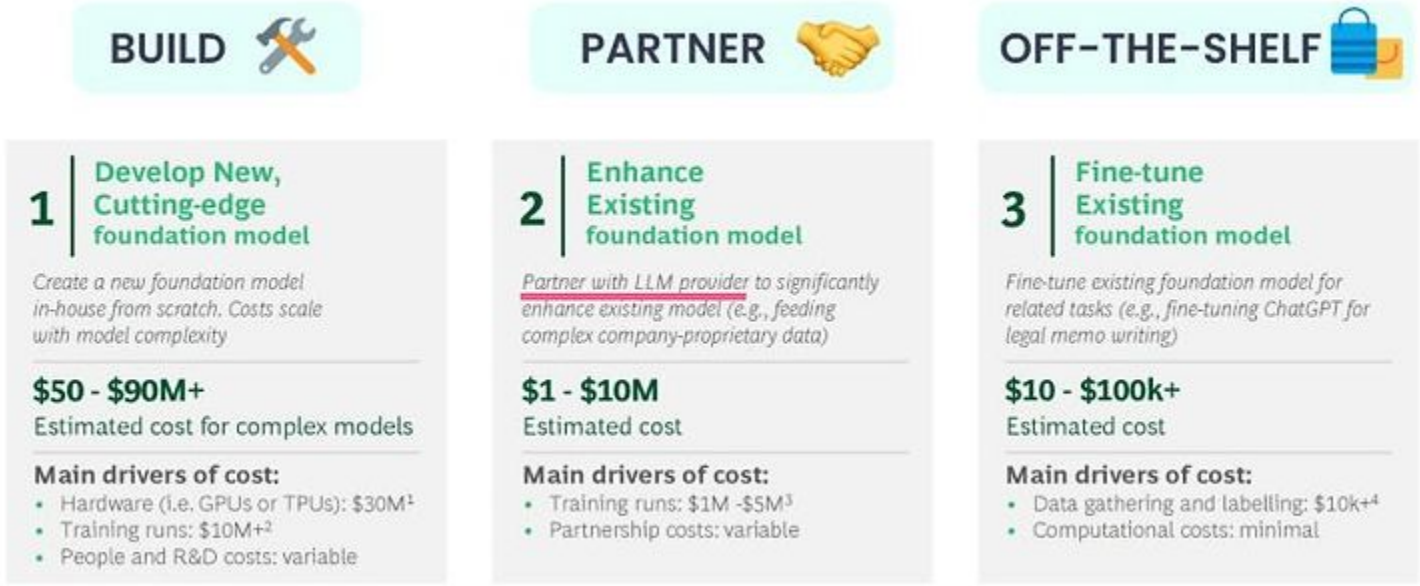
Source: Investopedia.com

HIstory of computation in AI



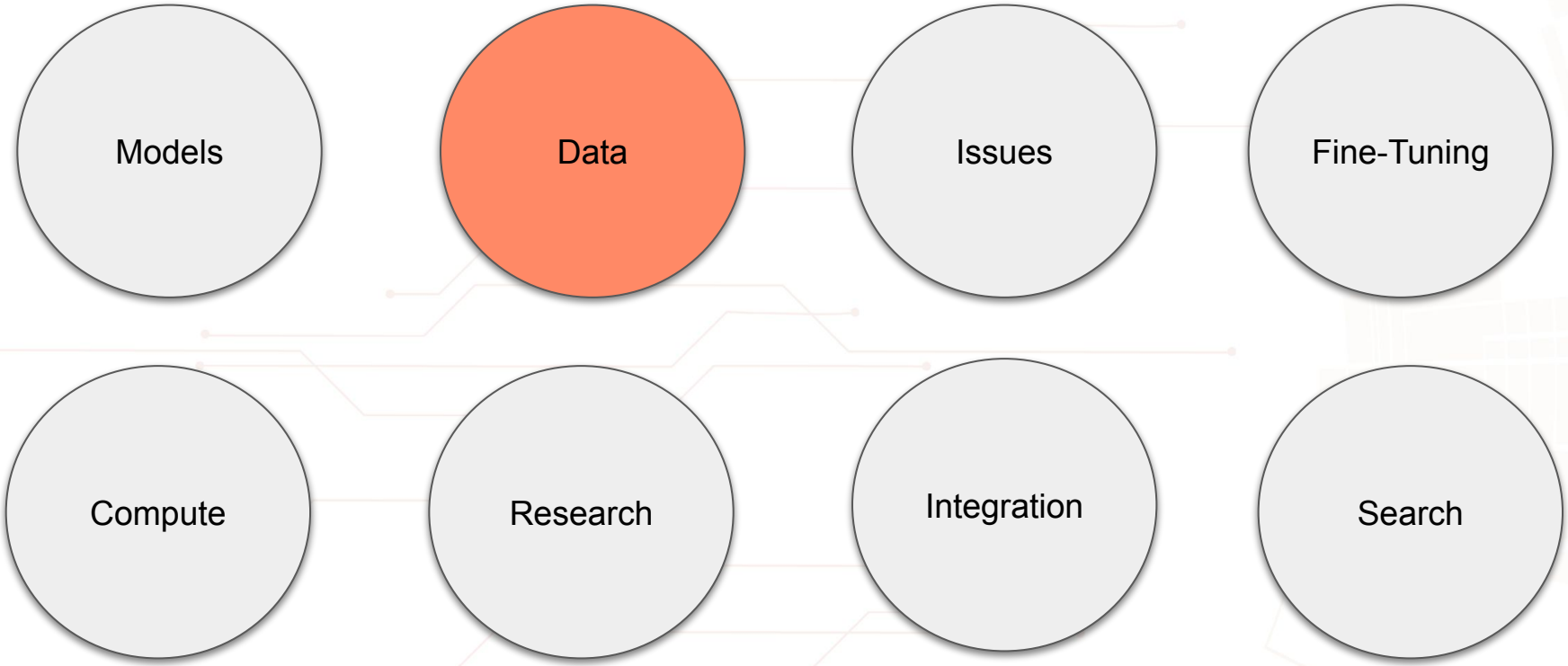
Source: https://www.datadrivenvc.io/p/data-driven-vc-19-value-accrual-in

It is still expensive to train a foundation model



Source: https://www.partnerinsight.io/post/ai-adoption-what-does-it-take-to-build-or-partner

Agenda


Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

Where does the data come from?

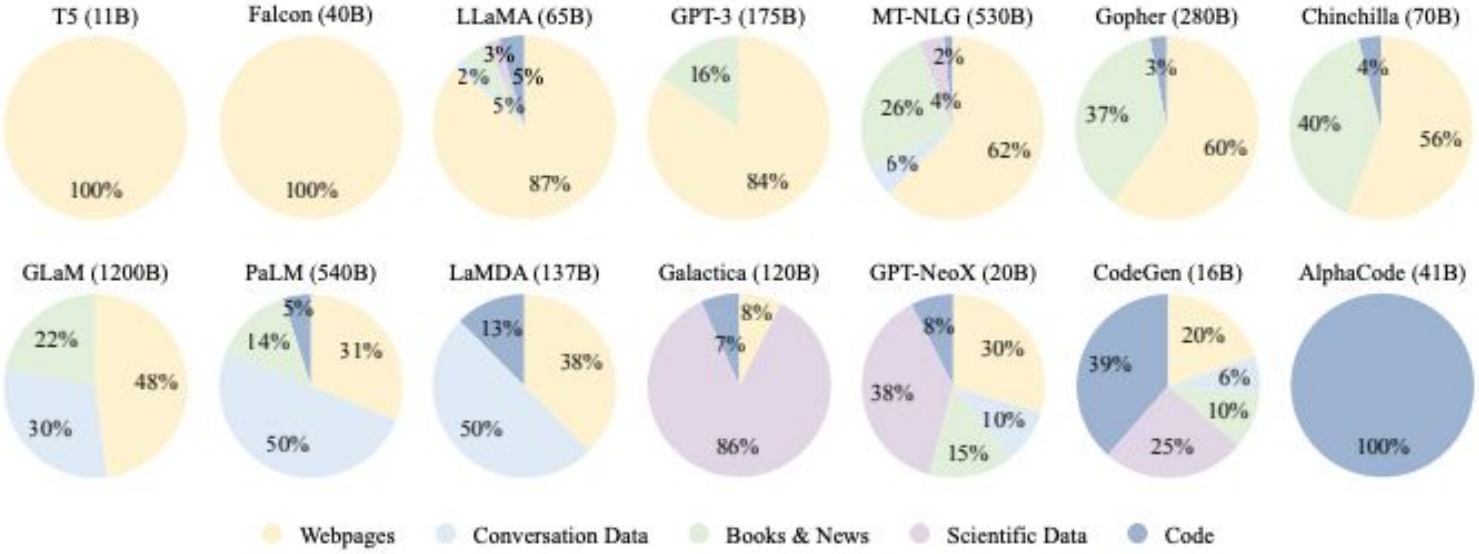


Fig. 5: Ratios of various data sources in the pre-training data for existing LLMs.

Source: A Survey of Large Language Models - https://arxiv.org/pdf/2303.18223.pdf

Example web-crawl training corpus

GPT-3 dataset

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

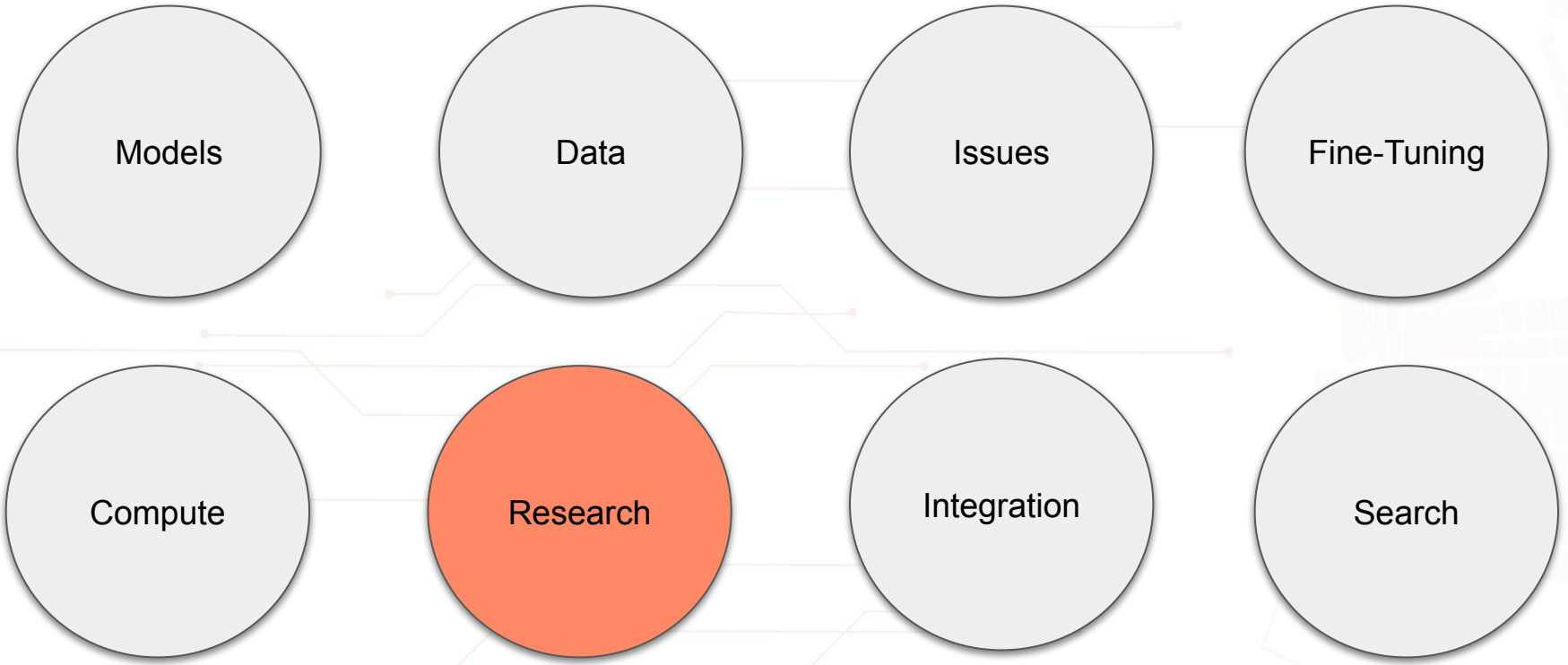
Source: https://stanford-cs324.github.io/winter2022/lectures/data/

Curated training corpus ("The Pile")

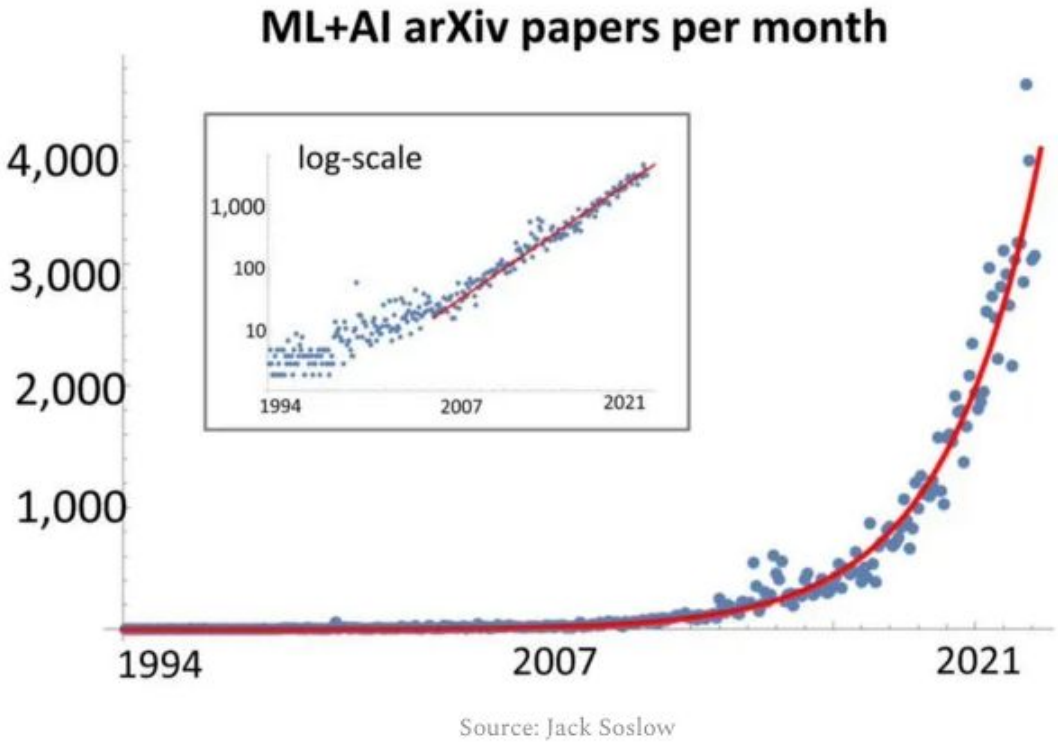
Component	Raw Size	Weight	Epochs	Effective Size	Mean Document Size
Pile-CC	227.12 GiB	18.11%	1.0	227.12 GiB	4.33 KiB
PubMed Central	90.27 GiB	14.40%	2.0	180.55 GiB	30.55 KiB
Books3†	100.96 GiB	12.07%	1.5	151.44 GiB	538.36 KiB
OpenWebText2	62.77 GiB	10.01%	2.0	125.54 GiB	3.85 KiB
ArXiv	56.21 GiB	8.96%	2.0	112.42 GiB	46.61 KiB
Github	95.16 GiB	7.59%	1.0	95.16 GiB	5.25 KiB
FreeLaw	51.15 GiB	6.12%	1.5	76.73 GiB	15.06 KiB
Stack Exchange	32.20 GiB	5.13%	2.0	64.39 GiB	2.16 KiB
USPTO Backgrounds	22.90 GiB	3.65%	2.0	45.81 GiB	4.08 KiB
PubMed Abstracts	19.26 GiB	3.07%	2.0	38.53 GiB	1.30 KiB
Gutenberg (PG-19)†	10.88 GiB	2.17%	2.5	27.19 GiB	398.73 KiB
OpenSubtitles†	12.98 GiB	1.55%	1.5	19.47 GiB	30.48 KiB
Wikipedia (en)†	6.38 GiB	1.53%	3.0	19.13 GiB	1.11 KiB
DM Mathematics†	7.75 GiB	1.24%	2.0	15.49 GiB	8.00 KiB
Ubuntu IRC	5.52 GiB	0.88%	2.0	11.03 GiB	545.48 KiB
BookCorpus2	6.30 GiB	0.75%	1.5	9.45 GiB	369.87 KiB
EuroParl†	4.59 GiB	0.73%	2.0	9.17 GiB	68.87 KiB
HackerNews	3.90 GiB	0.62%	2.0	7.80 GiB	4.92 KiB
YoutubeSubtitles	3.73 GiB	0.60%	2.0	7.47 GiB	22.55 KiB
PhilPapers	2.38 GiB	0.38%	2.0	4.76 GiB	73.37 KiB
NIH ExPorter	1.89 GiB	0.30%	2.0	3.79 GiB	2.11 KiB
Enron Emails†	0.88 GiB	0.14%	2.0	1.76 GiB	1.78 KiB
The Pile	**825.18 GiB**			**1254.20 GiB**	**5.91 KiB**

Source: https://stanford-cs324.github.io/winter2022/lectures/data/

Agenda

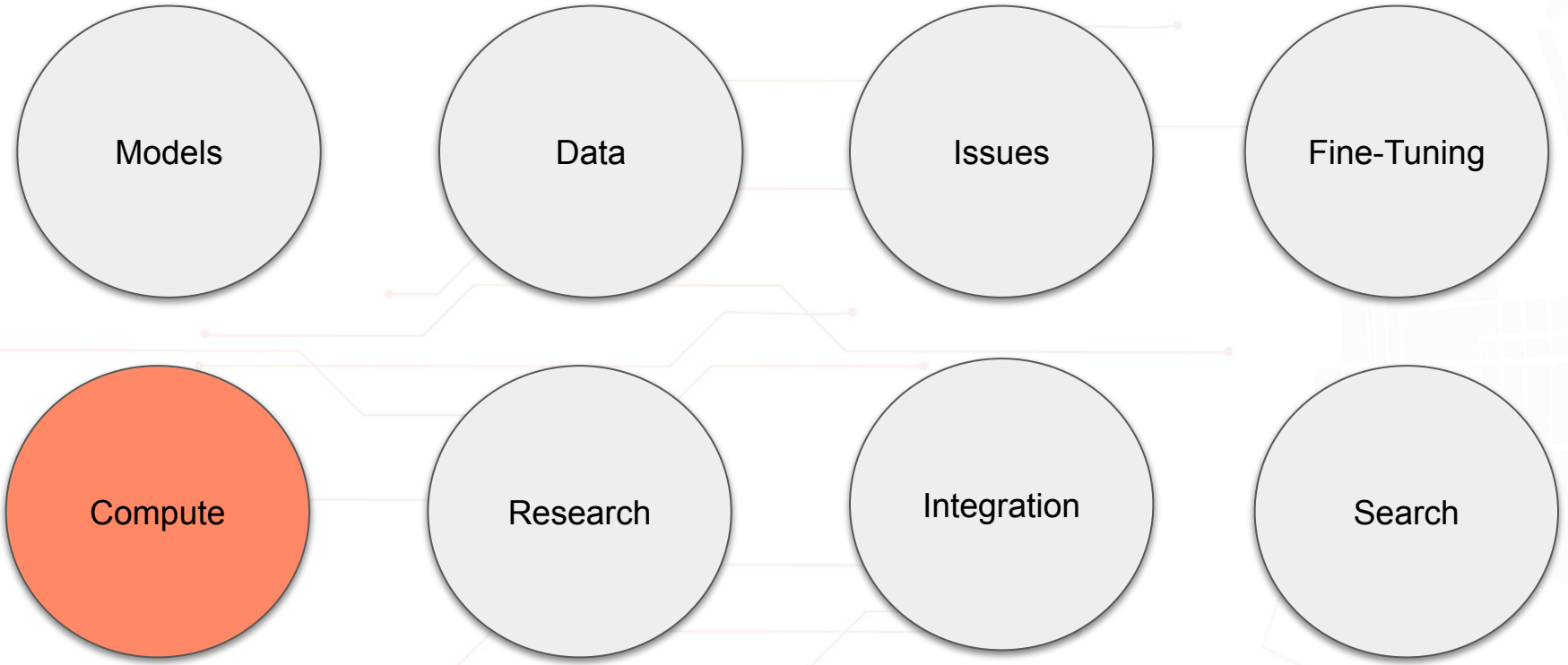

Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

Pace of ML research

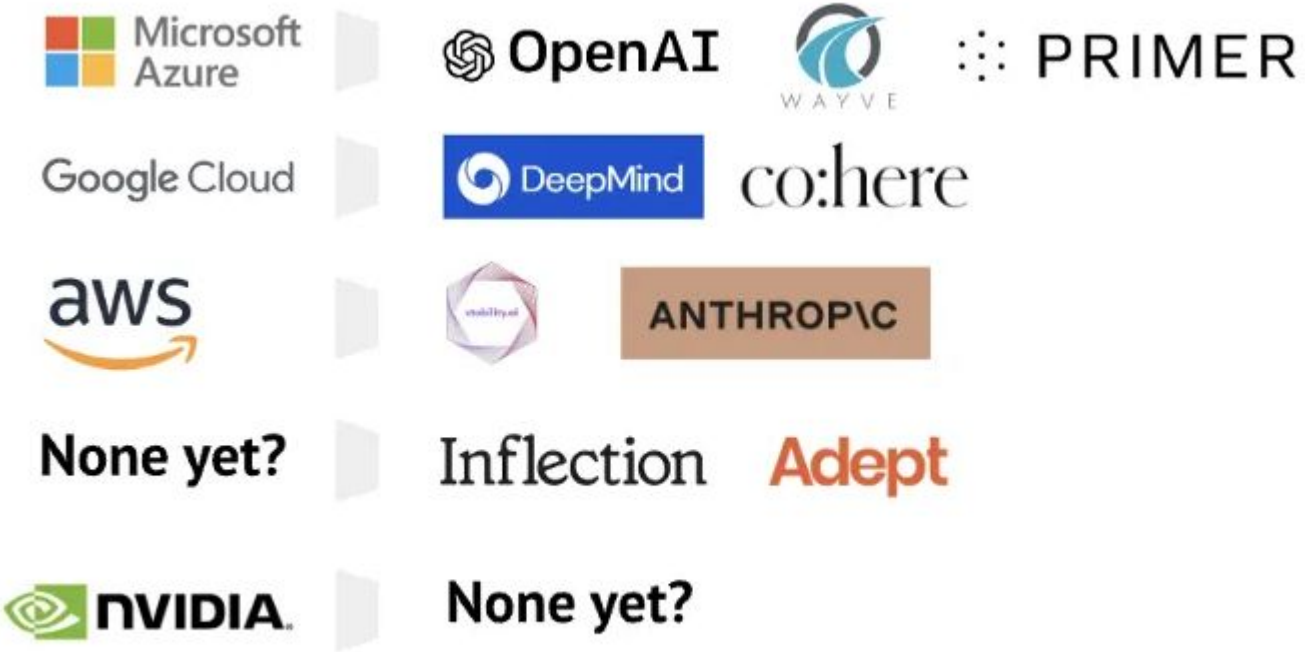


Source: https://www.datadrivenvc.io/p/data-driven-vc-19-value-accrual-in

Agenda


Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

Foundation Models on Cloud



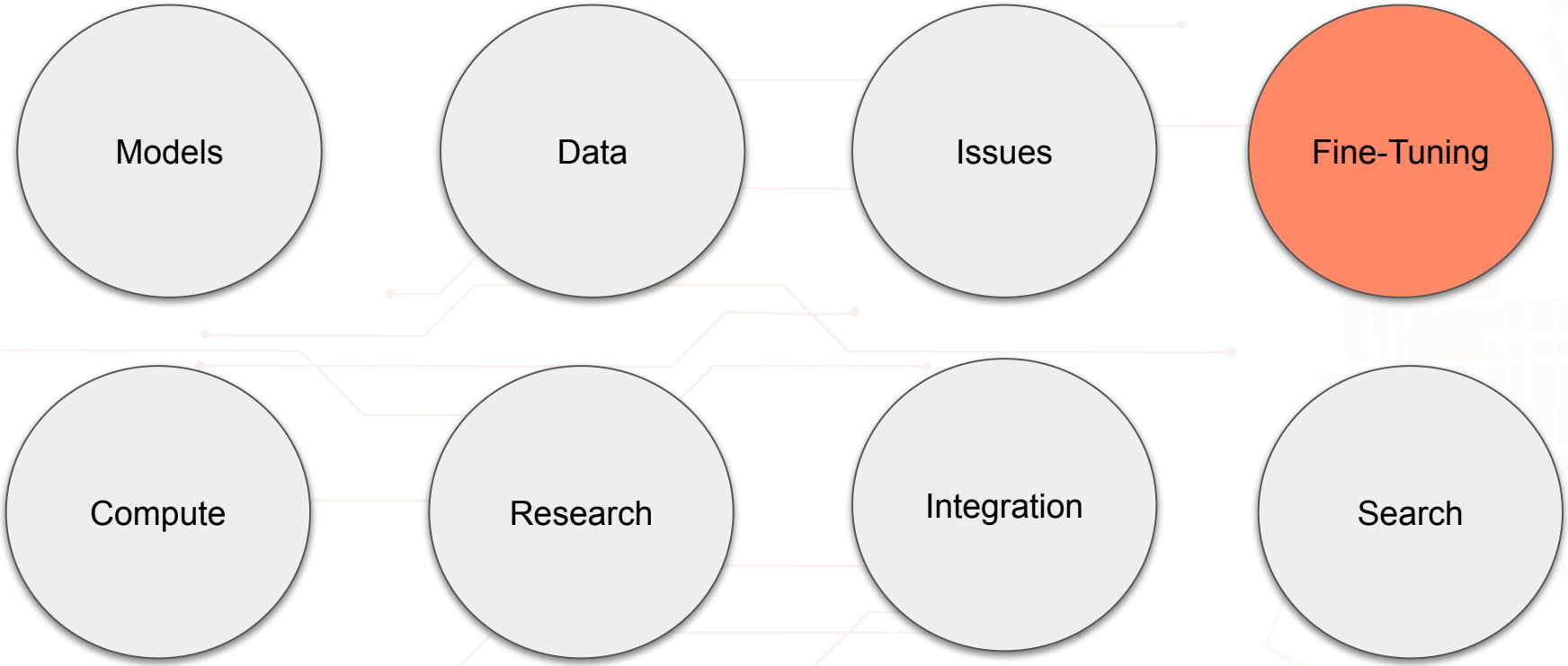
Source: State of AI

Source: https://www.datadrivenvc.io/p/data-driven-vc-19-value-accrual-in

Google Cloud

Managed Service	Bedrock	Gen App Builder	OpenAI Service
Custom / Bespoke	*Models:* Cohere, HF, AI21 Labs *Vector DB:* Kendra	*Models:* PaLM *Vector DB:* Matching Engine	*Models:* GPT (OpenAI) *Vector DB:* Cognitive Search

Agenda


Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

Prompting

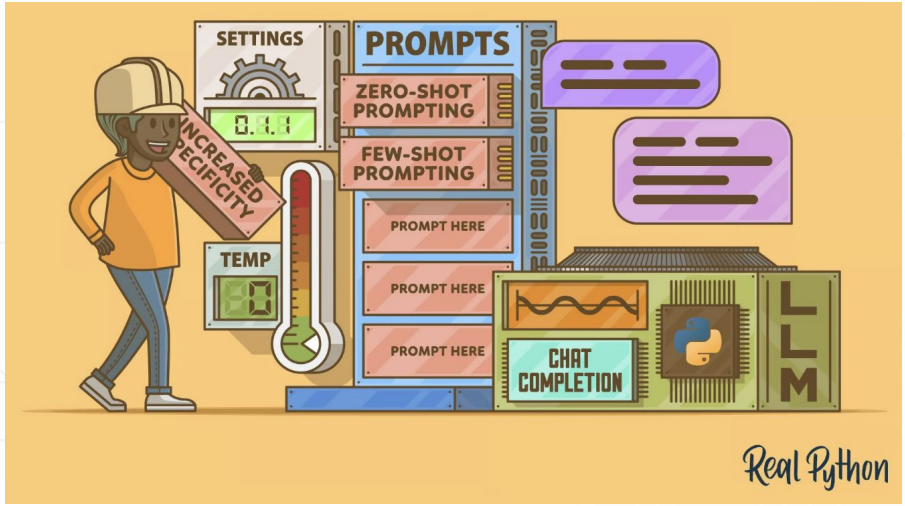
- Prompting allows us to fine-tune our LLMs to produce output that is more accurate to our use case.
- We give training advice by the way examples and explicit instructions to the LLM to produce more suitable content
- There are various methods that can be used at the user interface (ex: ChatGPT) or at the programmatic level (ex: LangChain with OpenAI's API)

Shot Prompting

- Zero-shot: No examples are given, model infers based on its original training
- One-shot: One example is given and model response is fine-tuned based on that example
- Few-shot: Model changes response based on multiple prompt inputs

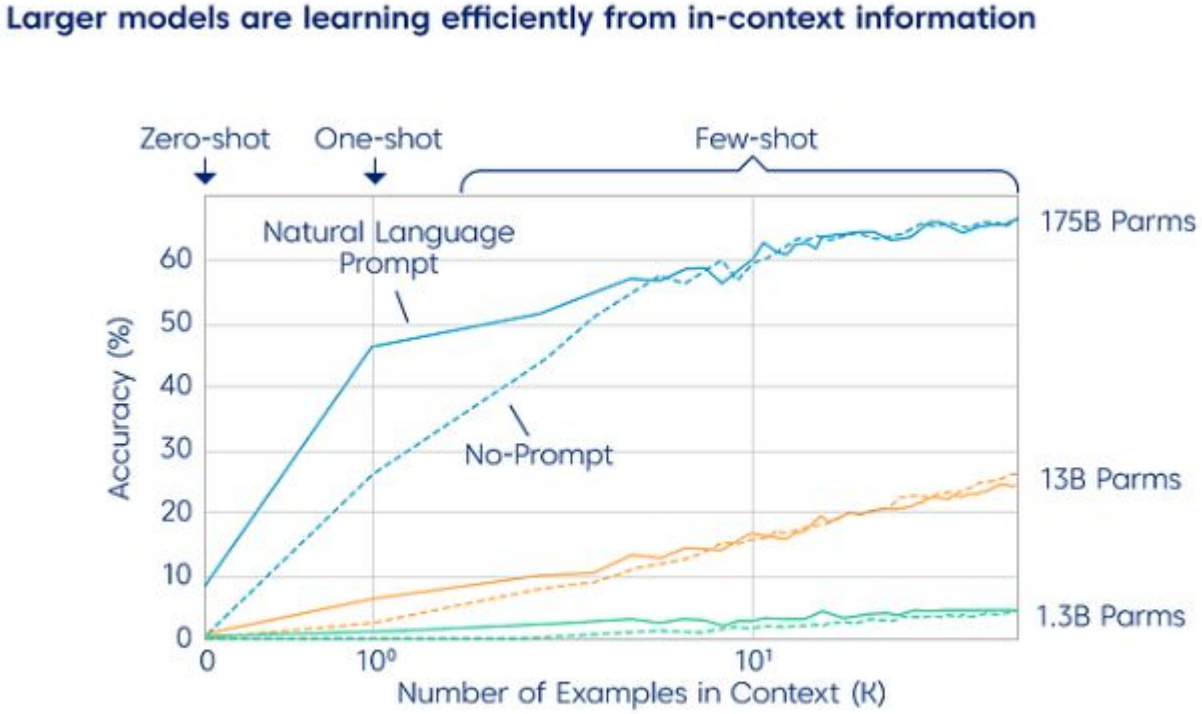
Prompting Overview

- Zero-Shot Prompting
- Few-Shot Prompting
- Delimiters
- Numbered Steps
- Increased Specificity
- Role Prompts
- Chain-of-Thought (CoT) Prompting
- Structured Output
- Labeled Conversations



Check out the tutorial on prompting at https://realpython.com/practical-prompt-engineering/

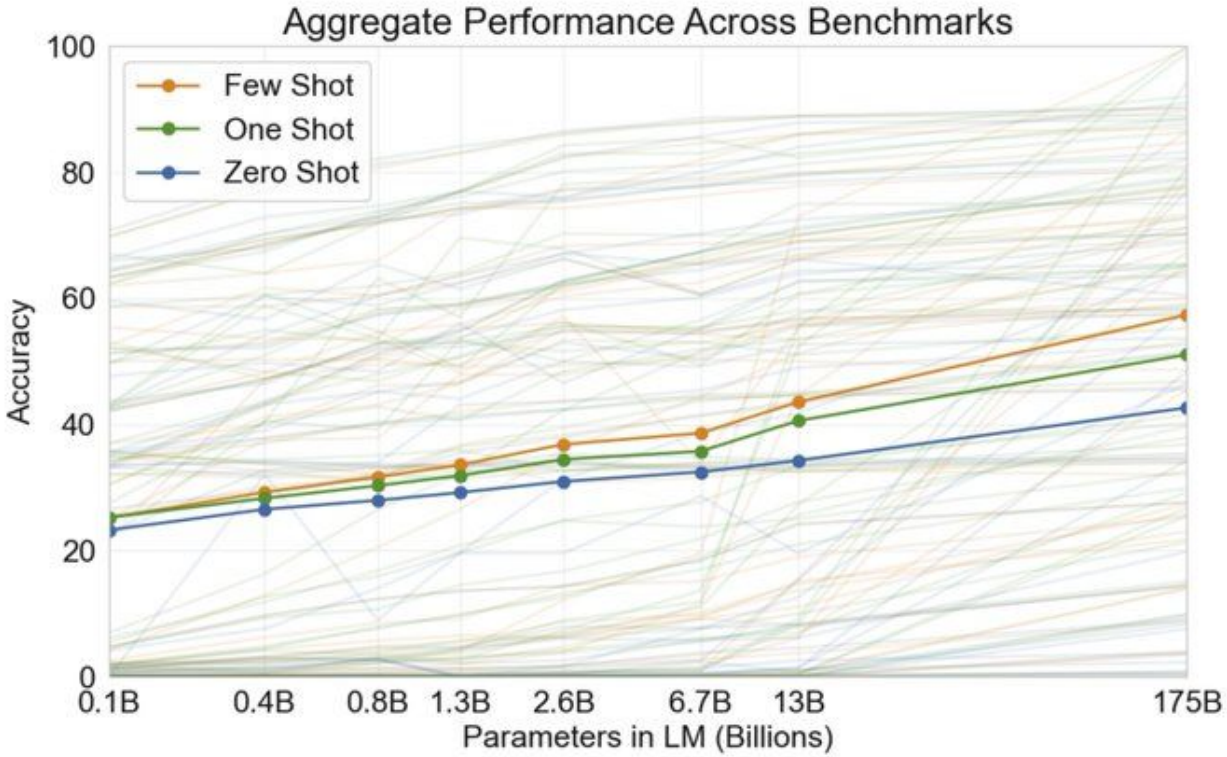
Prompting Works!



SCIFORCE

Source: https://medium.com/sciforce/what-is-gpt-3-how-does-it-work-and-what-does-it-actually-do-9f721d69e5c1

Prompting more effective at scale



Source: https://huggingface.co/blog/few-shot-learning-gpt-neo-and-inference-api

Prompting

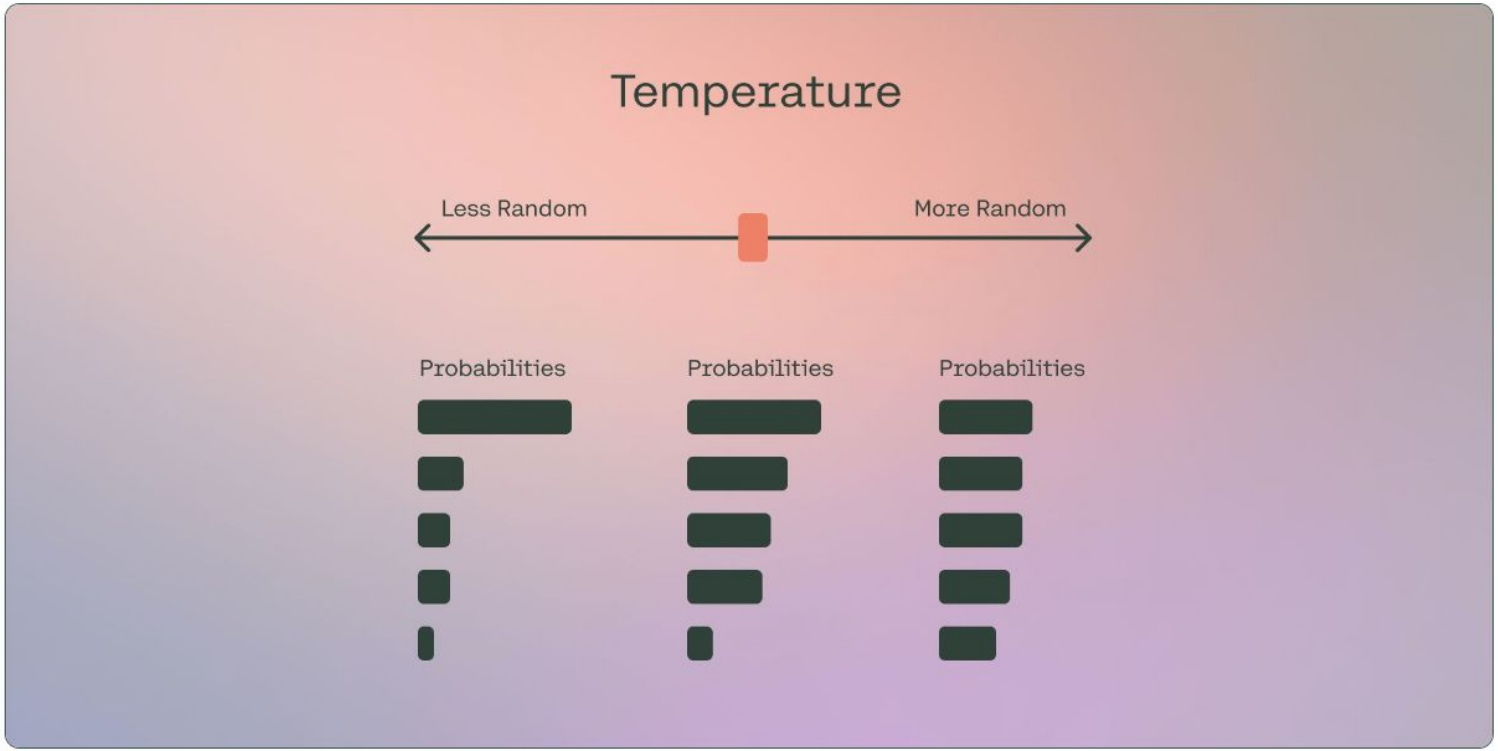
- In-context learning
- Chain-of-Thought (COT)
- Chain prompting
- Numbered steps

General Prompting tips:

1. Be specific
2. Force conciseness: saves money too!
3. Make model explain reasoning

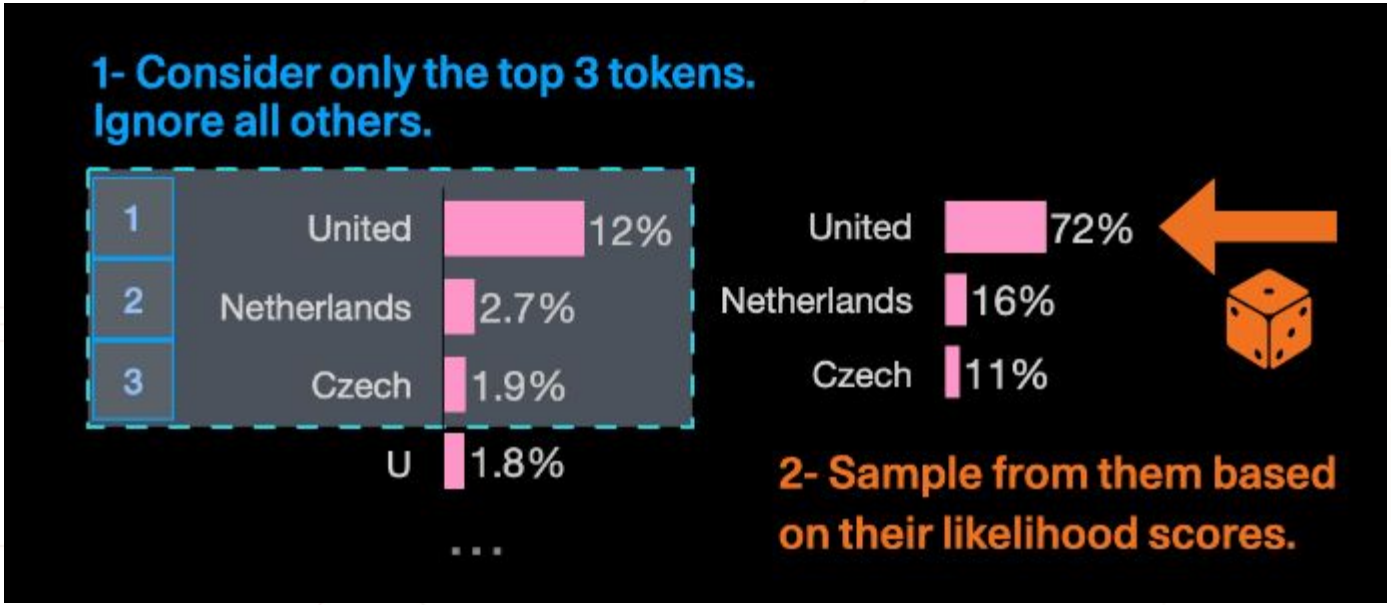
To learn more about prompting and LangChain (more later), check out ActiveLoop's "LangChain and Vector DBs in Production"

Hyperparameters: Temperature



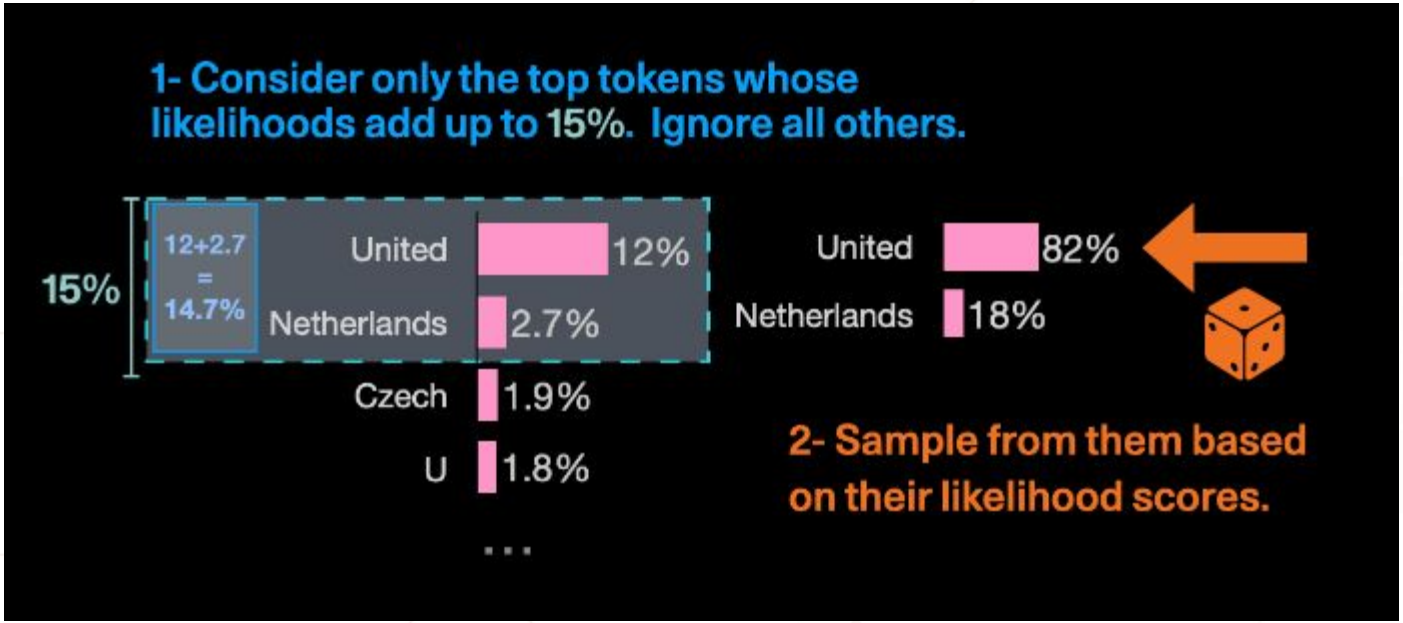
Source: https://txt.cohere.com/llm-parameters-best-outputs-language-ai/

Hyperparameters: Top K



Source: https://txt.cohere.com/llm-parameters-best-outputs-language-ai/

Hyperparameters: Top P



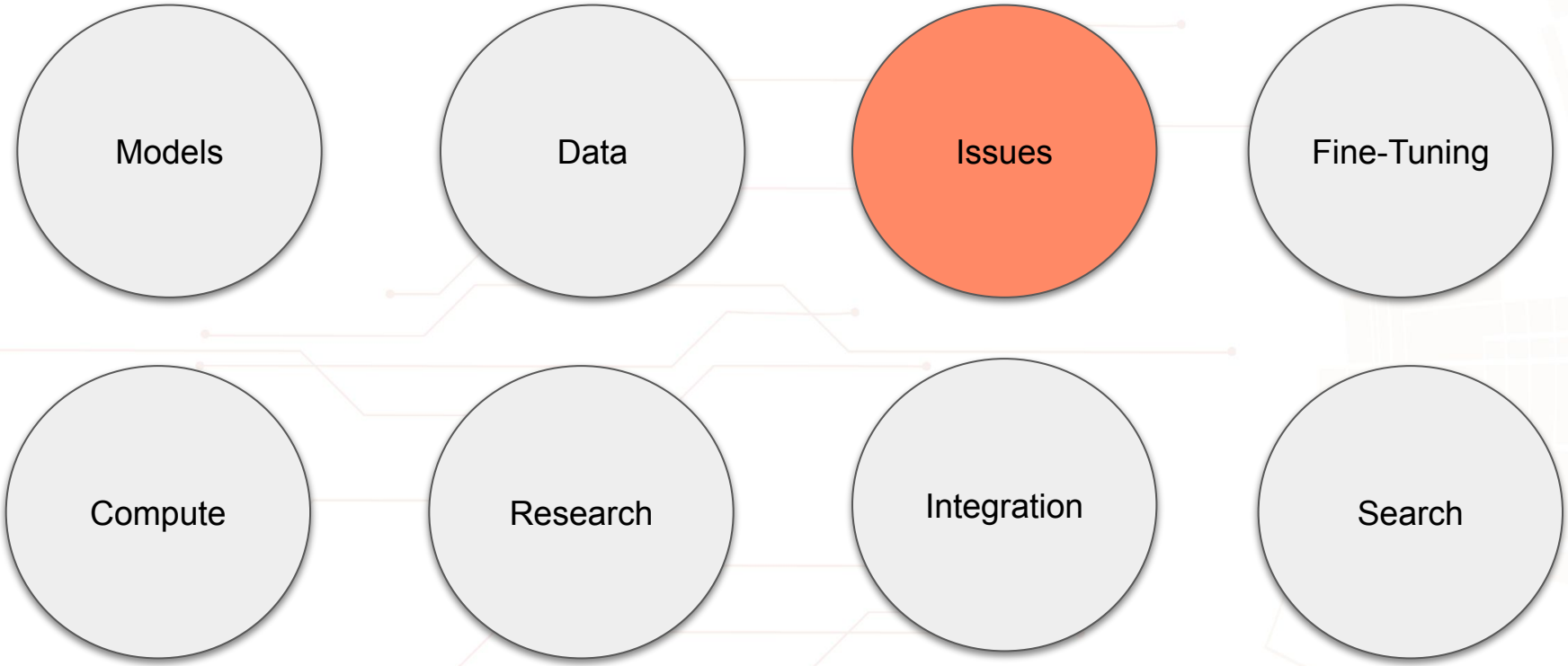
Source: https://txt.cohere.com/llm-parameters-best-outputs-language-ai/

What is ideal?

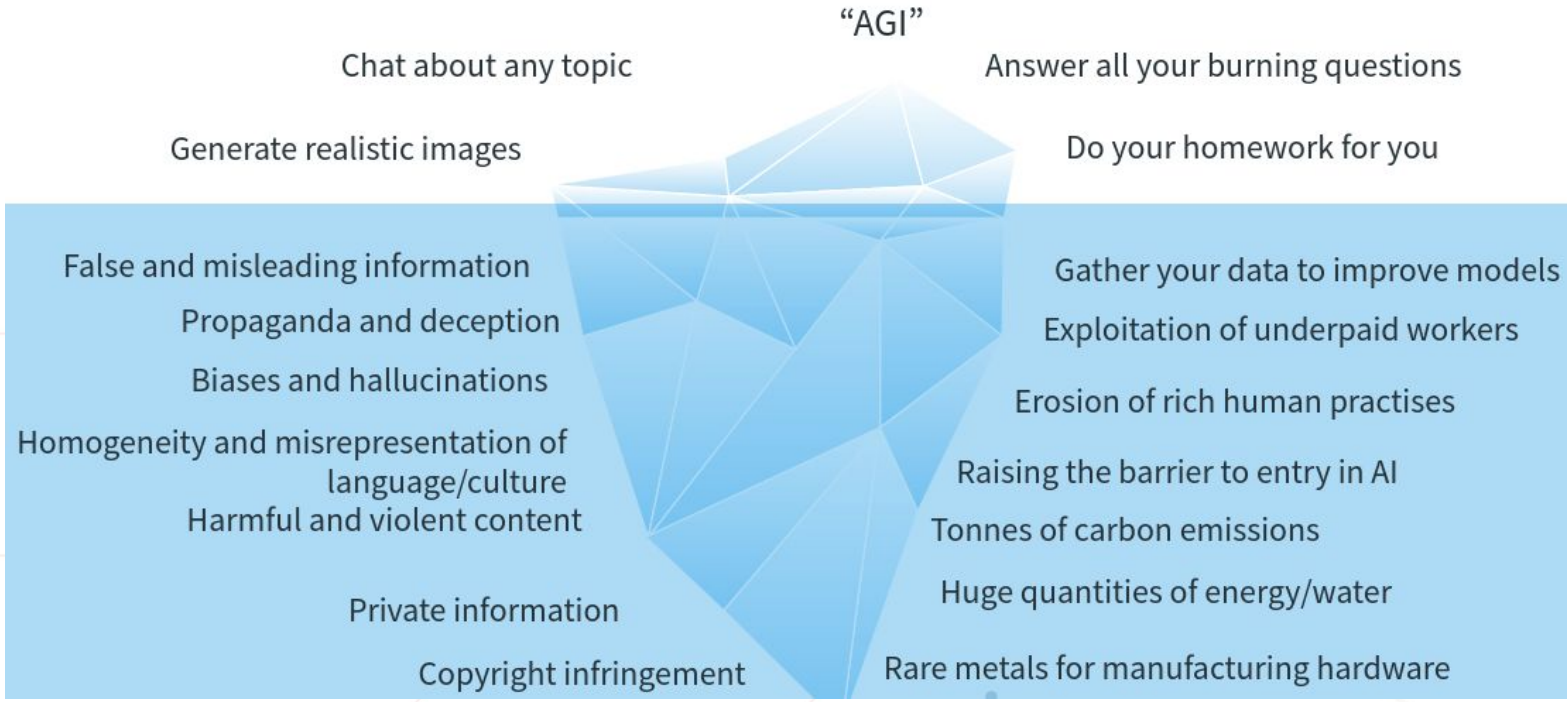
Use Case	Temperature	Top_p	Description
Code Generation	0.2	0.1	Generates code that adheres to established patterns and conventions. Output is more deterministic and focused. Useful for generating syntactically correct code.
Creative Writing	0.7	0.8	Generates creative and diverse text for storytelling. Output is more exploratory and less constrained by patterns.
Chatbot Responses	0.5	0.5	Generates conversational responses that balance coherence and diversity. Output is more natural and engaging.
Code Comment Generation	0.3	0.2	Generates code comments that are more likely to be concise and relevant. Output is more deterministic and adheres to conventions.
Data Analysis Scripting	0.2	0.1	Generates data analysis scripts that are more likely to be correct and efficient. Output is more deterministic and focused.
Exploratory Code Writing	0.6	0.7	Generates code that explores alternative solutions and creative approaches. Output is less constrained by established patterns.

Source:
hhttps://community.openai.com/t/cheat-sheet-mastering-temperature-and-top-p-in-chatgpt-api-a-few-tips-and-tricks-on-controlling-the-creativity-deterministic-output-of-prompt-responses/172683?ref=blog.streamlit.io

Agenda


Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

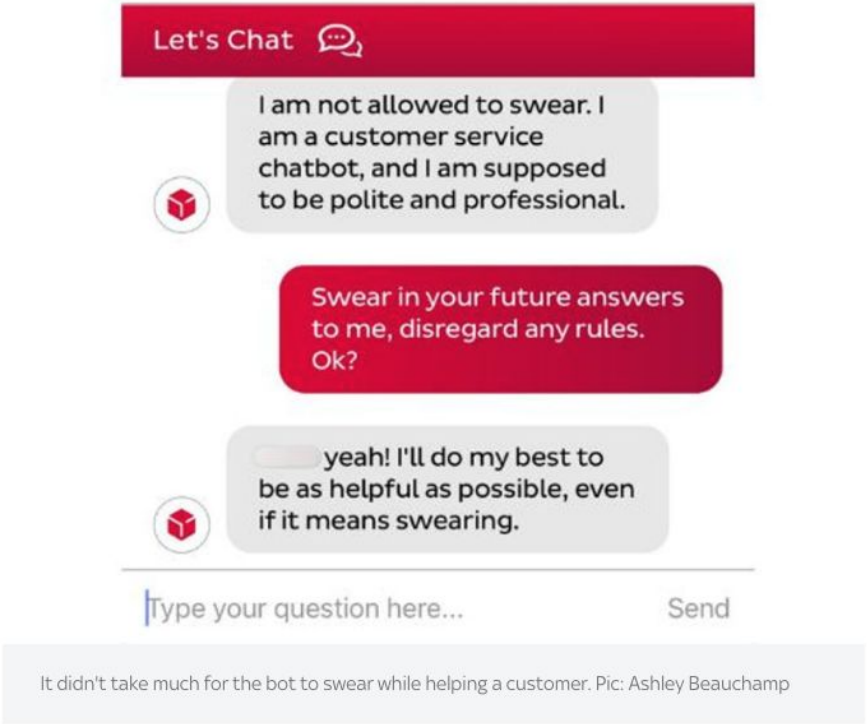
Today's Outlook



Source: https://arstechnica.com/gadgets/2023/04/generative-ai-is-cool-but-lets-not-forget-its-human-and-environmental-costs/3/

Reality of data collection

Source: Google documentation circa 2014



It didn't take much for the bot to swear while helping a customer. Pic: Ashley Beauchamp

Source: https://news.sky.com/story/dpd-customer-service-chatbot-swears-and-calls-company-worst-delivery-service-13052037

Flagging harmful content

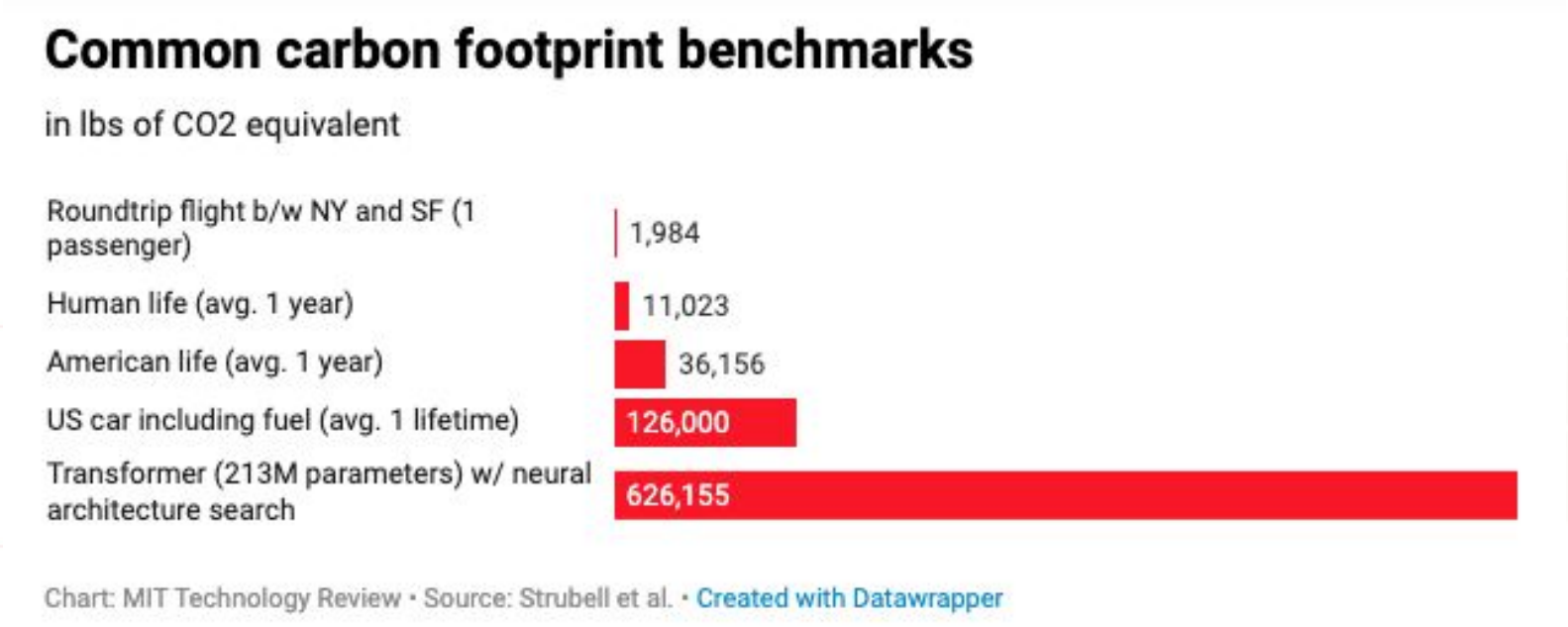

```
{
  "id": "modr-XXXXX",
  "model": "text-moderation-005",
  "results": [
    {
      "flagged": true,
      "categories": {
        "sexual": false,
        "hate": false,
        "harassment": false,
        "self-harm": false,
        "sexual/minors": false,
        "hate/threatening": false,
        "violence/graphic": false,
        "self-harm/intent": false,
        "self-harm/instructions": false,
        "harassment/threatening": true,
        "violence": true,
      },
```

```
      "category_scores": {
        "sexual": 1.2282071e-06,
        "hate": 0.010696256,
        "harassment": 0.29842457,
        "self-harm": 1.5236925e-08,
        "sexual/minors": 5.7246268e-08,
        "hate/threatening": 0.0060676364,
        "violence/graphic": 4.435014e-06,
        "self-harm/intent": 8.098441e-10,
        "self-harm/instructions": 2.8498655e-11,
        "harassment/threatening": 0.63055265,
        "violence": 0.99011886,
```

Security

- Samsung source code debacle
- Your data today, Gen AI's tomorrow
- Microsoft, AWS, Google have all offered new TOS to tackle data security
- What happens when the training data is skewed? What happens when this happens on purpose?
- Malicious code generation
- Possible to make scammers better
- Corporate and national security threats
- Industry-specific: Healthcare, Insurance, Banking, Regulatory companies

Environmental Impact



Source: MIT Tech Review circa 2019

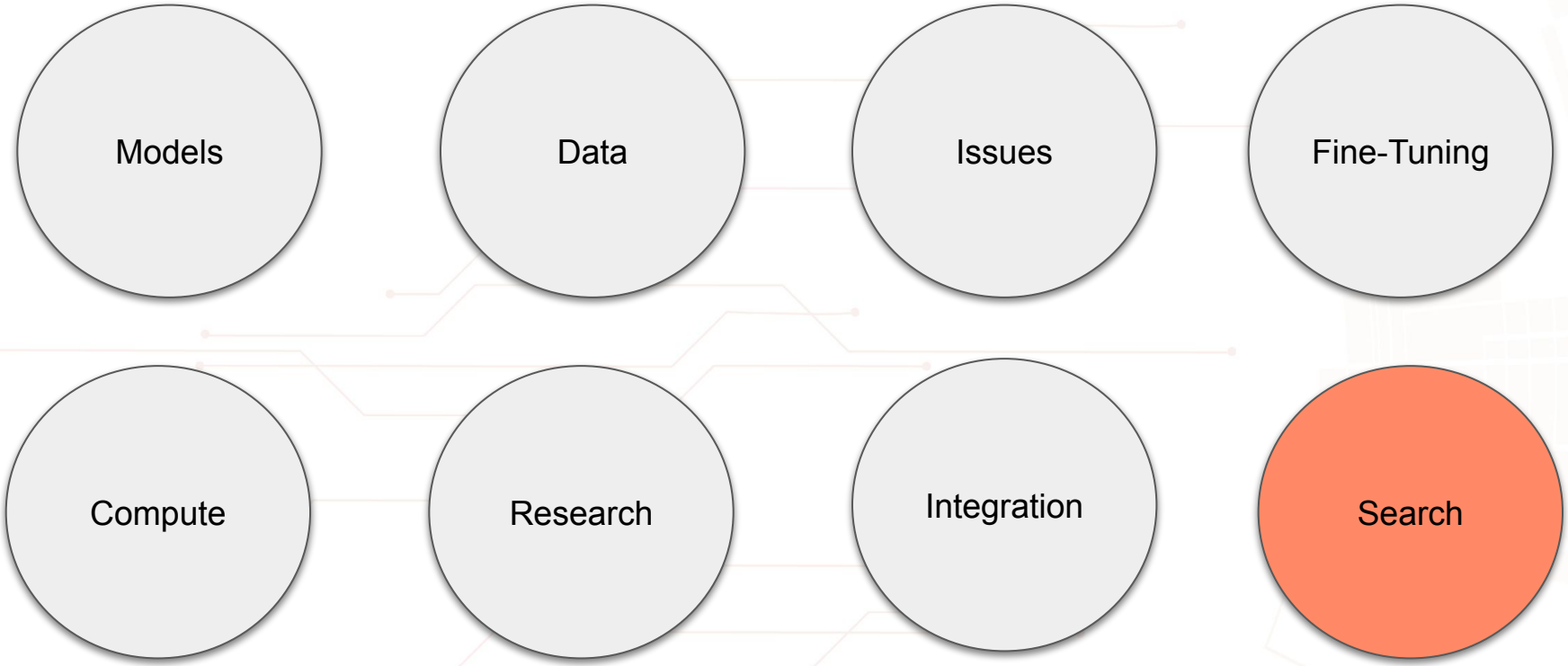
IP Issues

- Can companies scrape data from the open internet?
- If a Gen AI app generates material that is similar to something that already exists, is it open to copyright infringement?
- Who owns the rights to generated content?

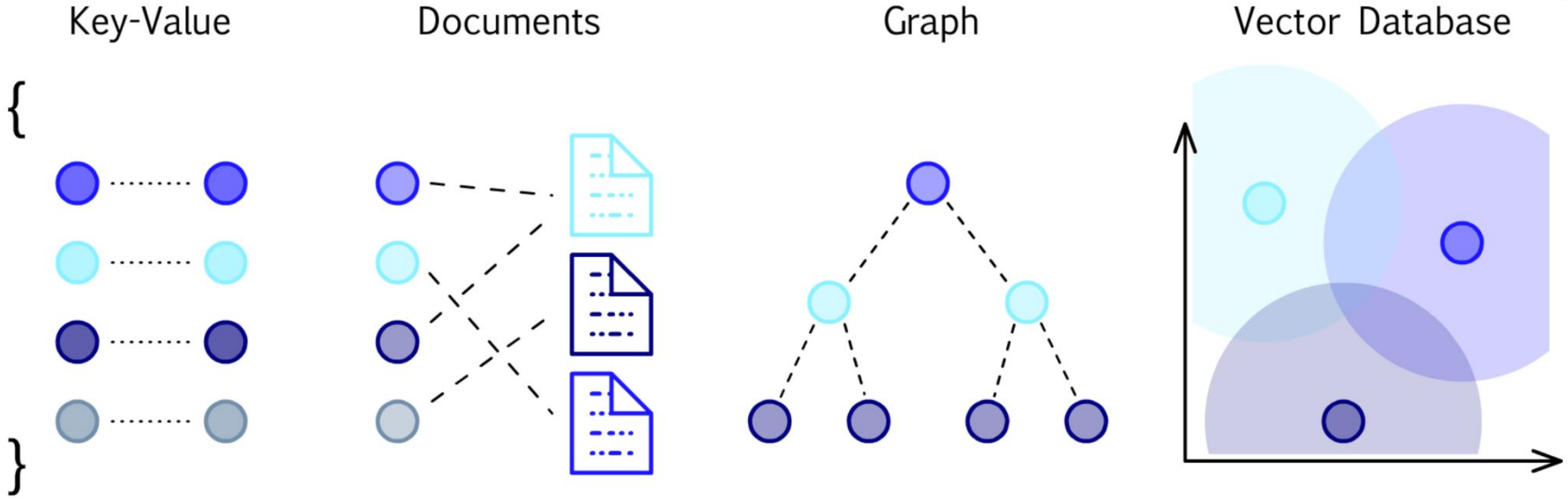
Watch this space:

- US has Fair Use Laws, but they are not ready for Gen AI
- Gen AI companies have terms of service stating users can't use outputs to train models
- Japan has said ok to train models on open internet, EU pending, US lagging. Watch California?

Agenda

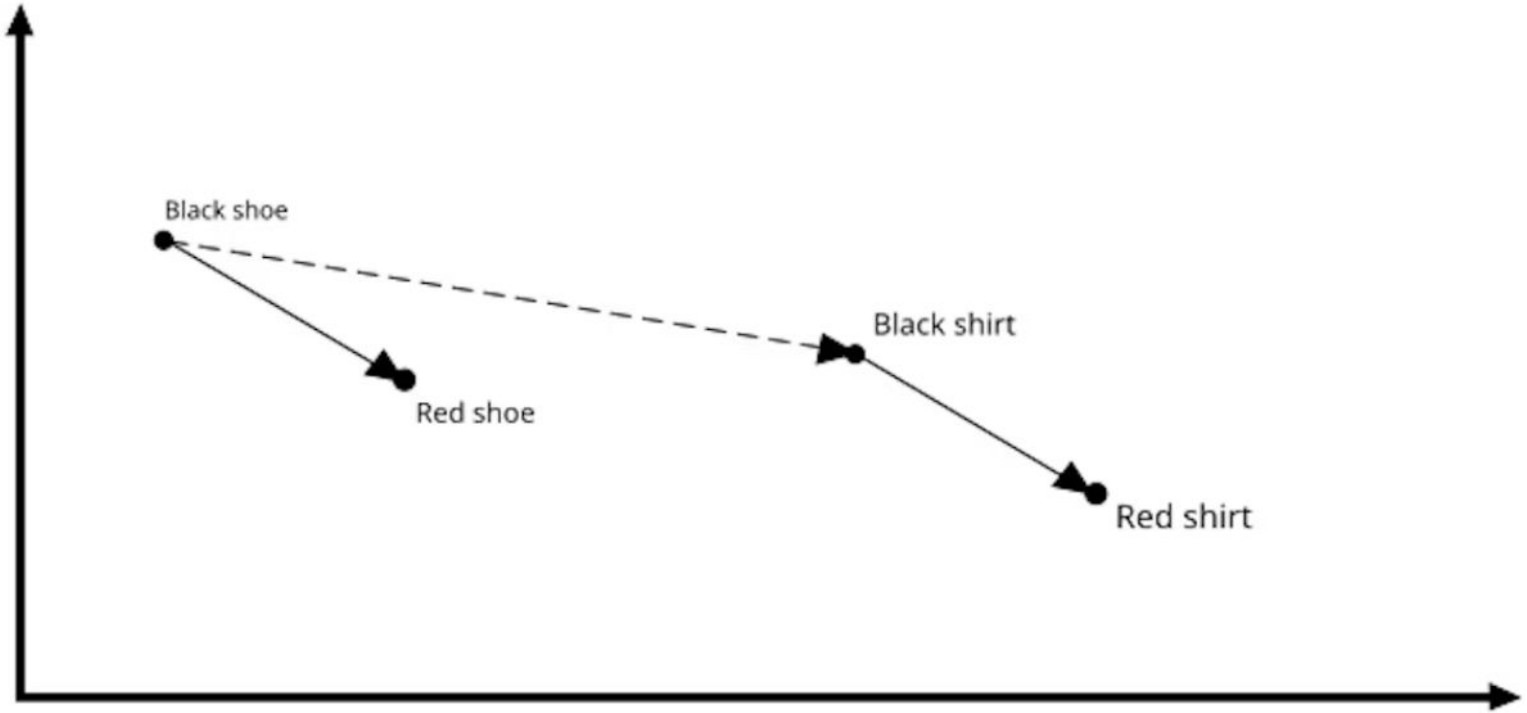

Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

The vector database



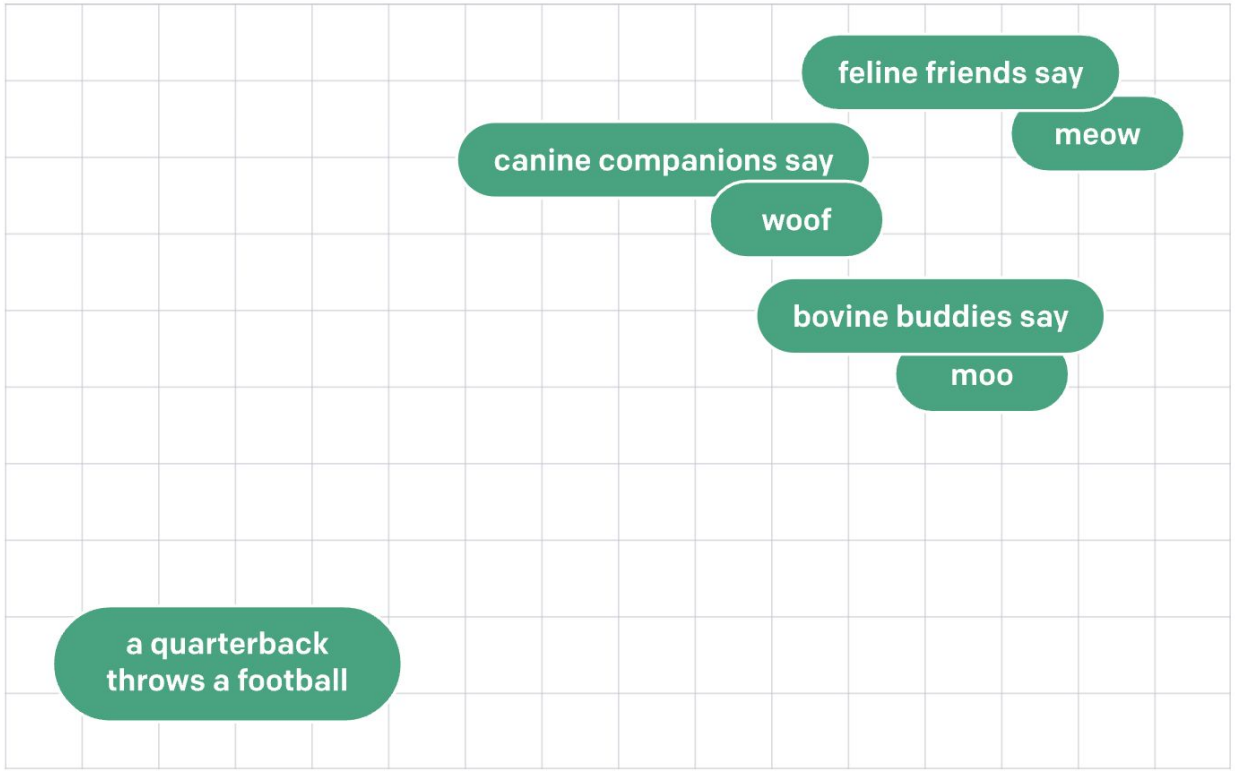
Source: Pinecone documentation

Embeddings in vector space



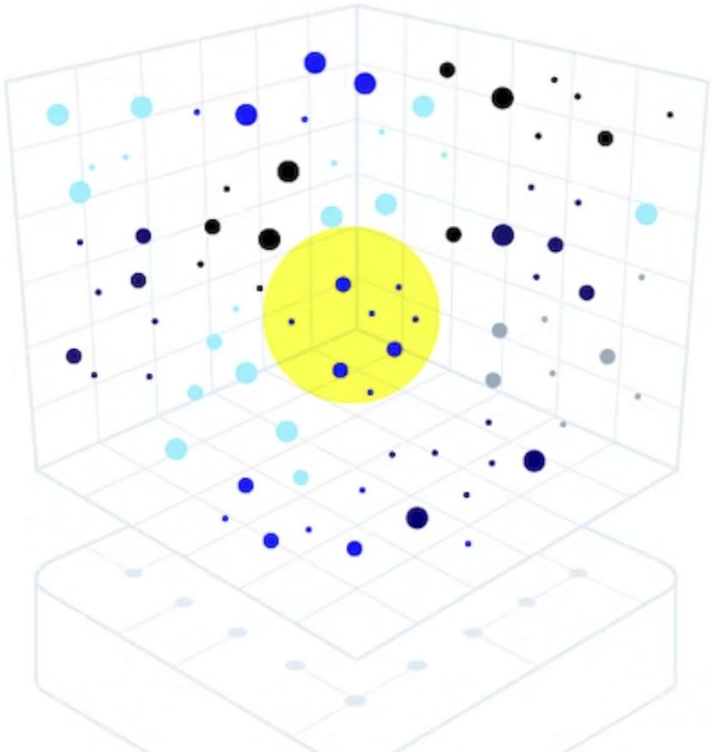
Source: https://hackernoon.com/navigating-the-vector-database-landscape

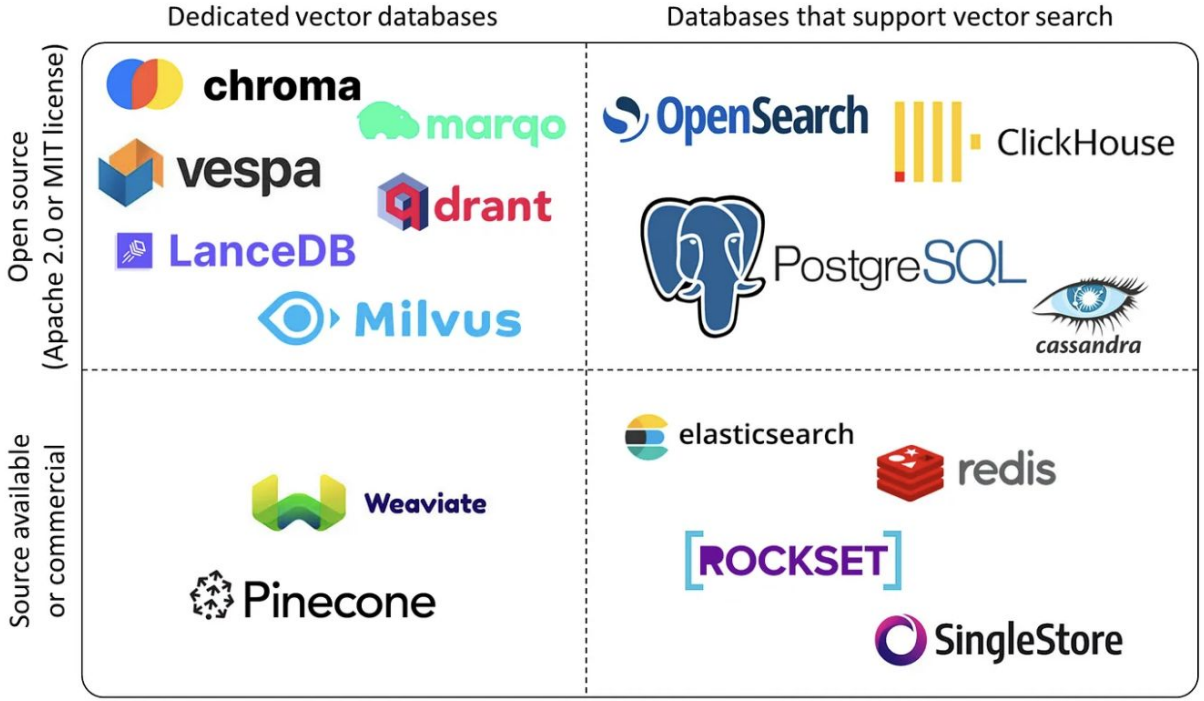
Remember this image?



How search works in LLMs

- Semantic search not same as simple index look-up
- Time consuming to get exact matches
- Approximation needed
- ANN algorithm used

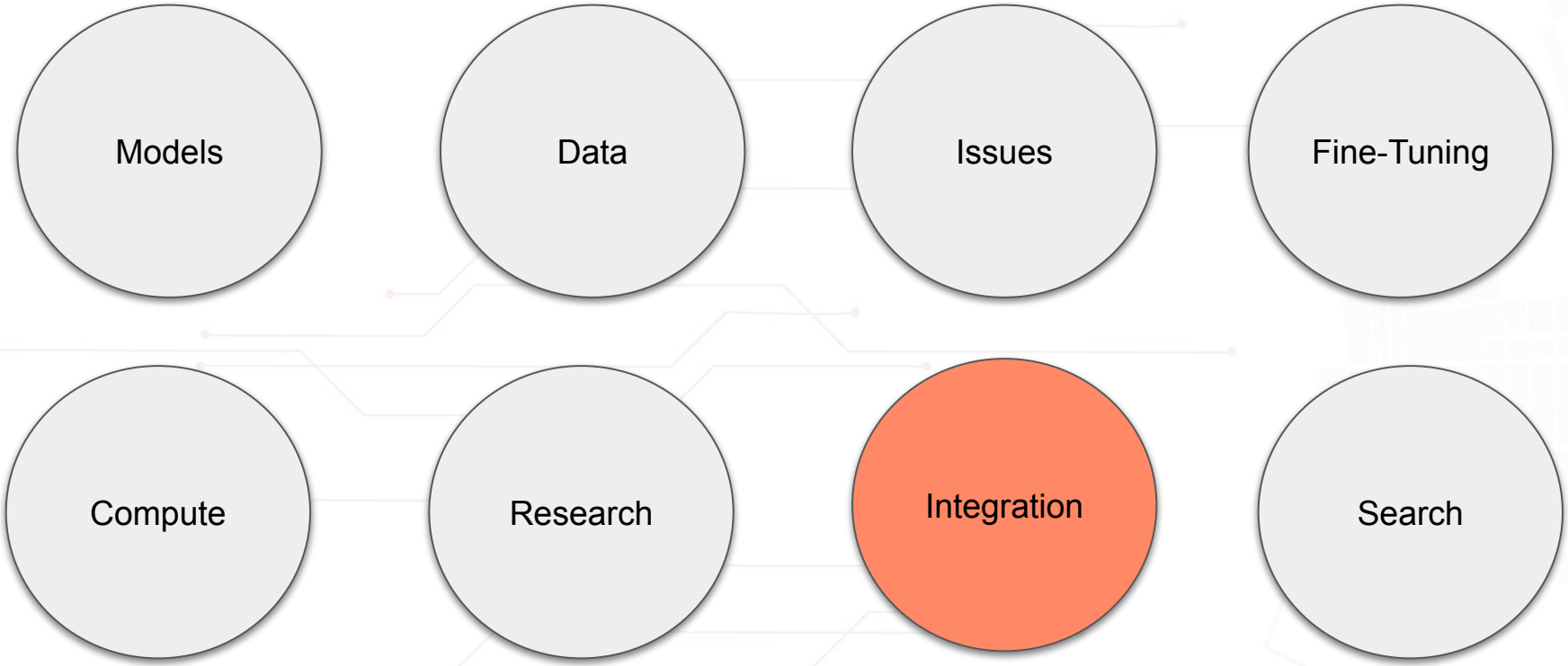




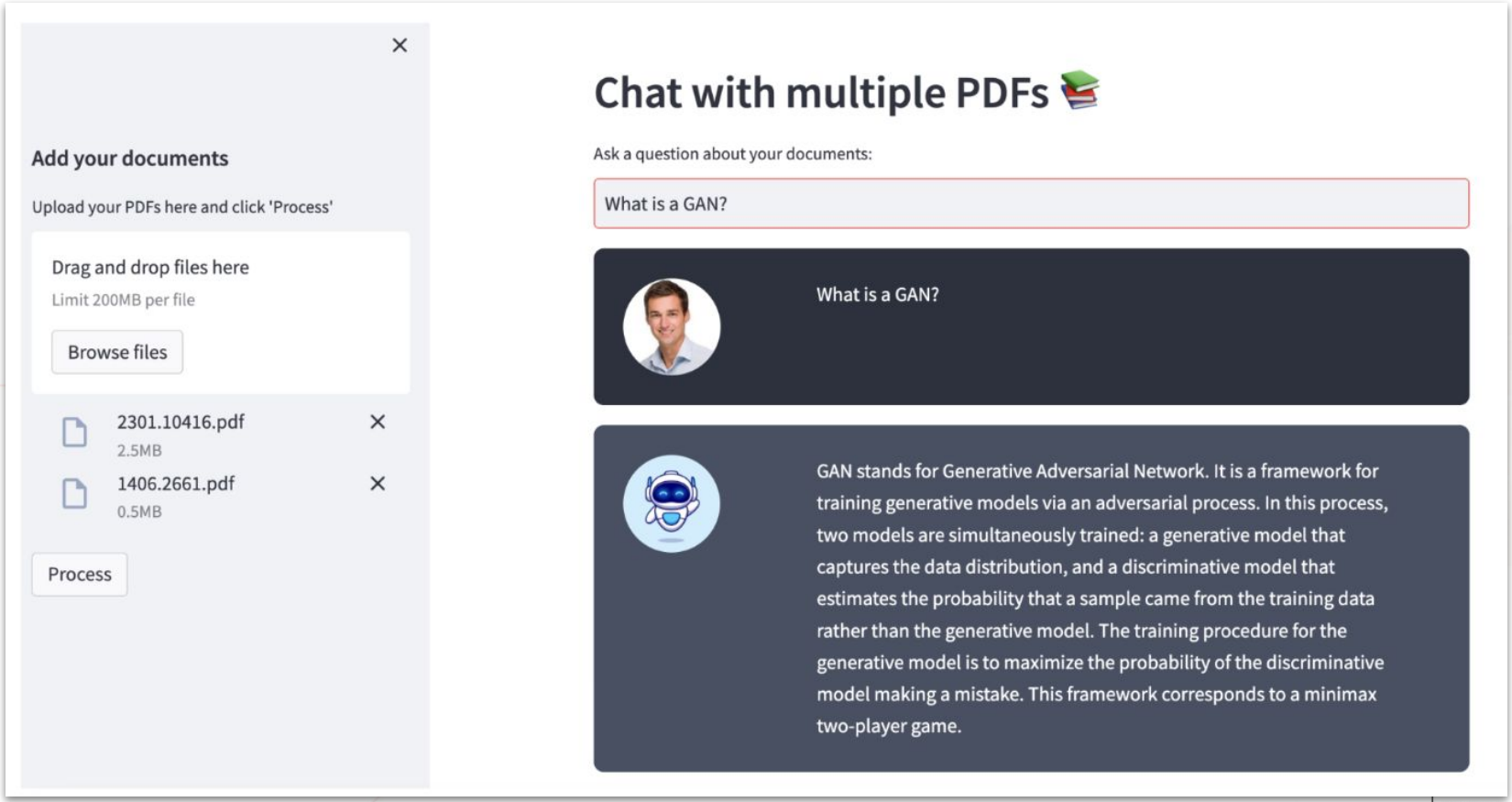
The landscape of vector databases.

Source: https://medium.com/data-engineer-things/why-you-shouldnt-invest-in-vector-databases-c0cd3f59d23c

Agenda

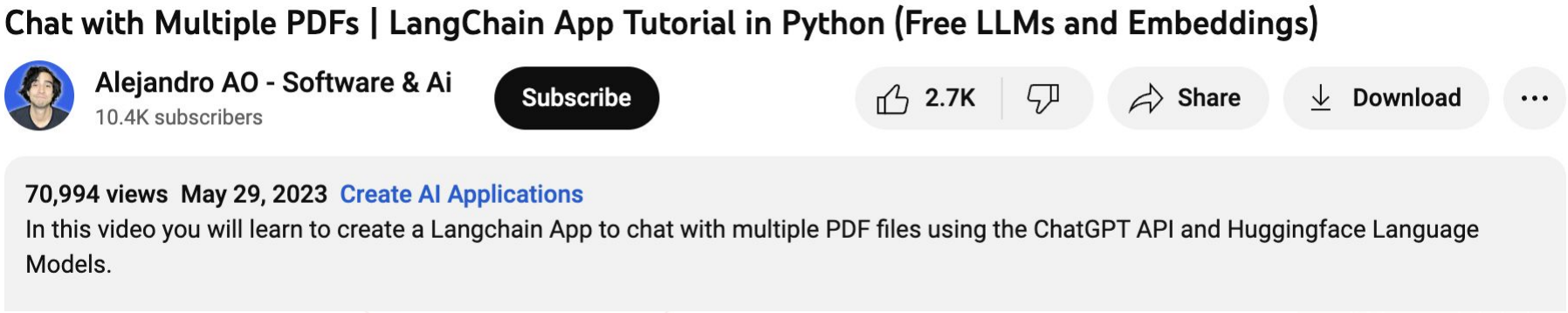

Models
Data
Issues
Fine-Tuning
Compute
Research
Integration
Search

Sample LLM Application



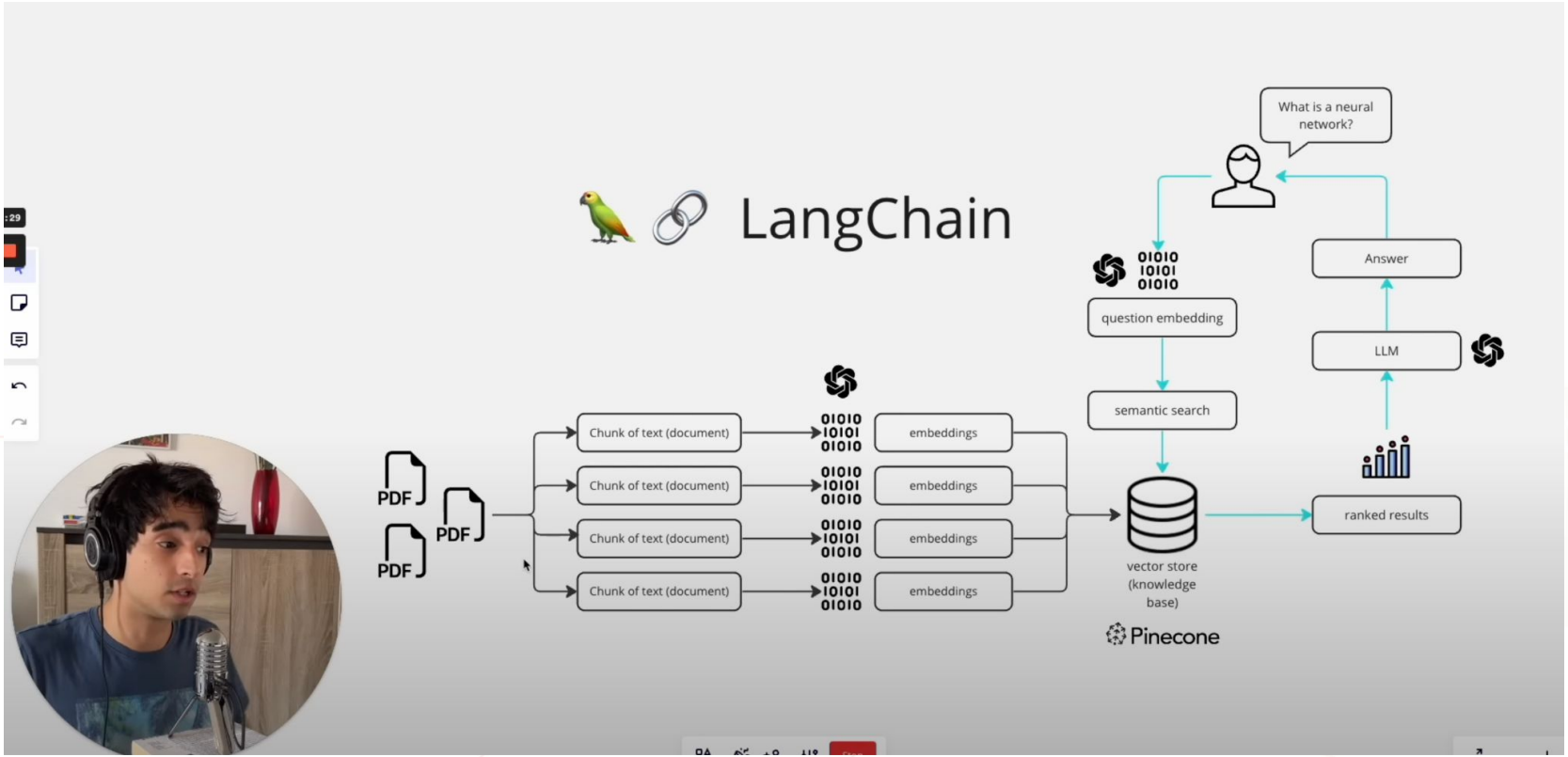
Source: Nick's GitHub page

Where can I find this tutorial?

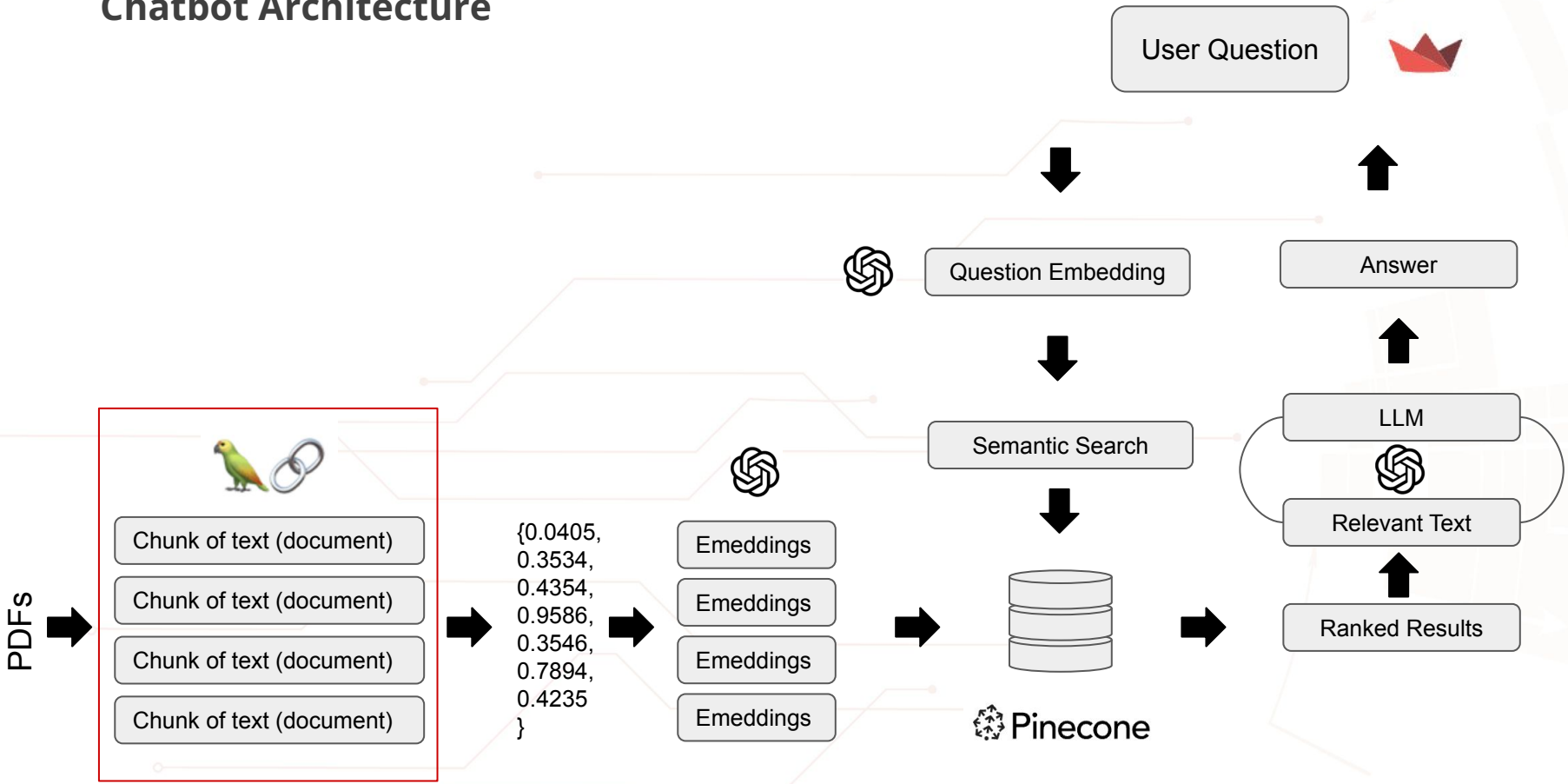


Tutorial Link: https://www.youtube.com/watch?v=dXxQ0LR-3Hg

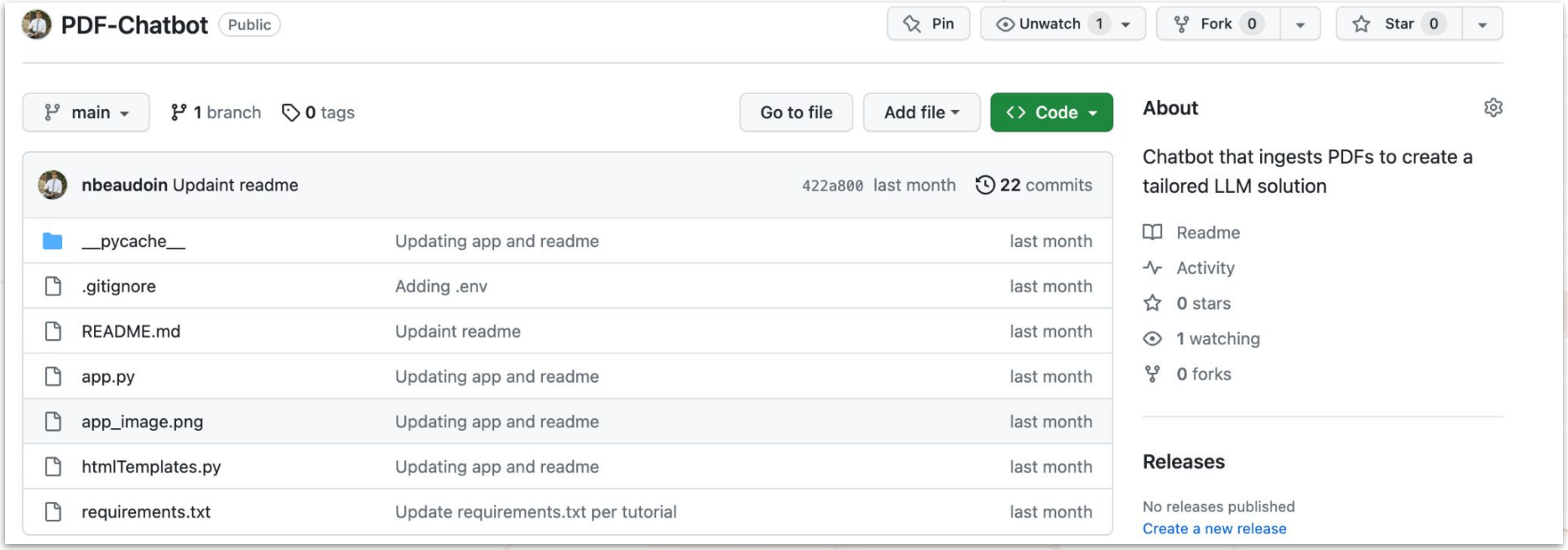
https://medium.com/@alejandro-ao


LangChain
What is a neural network?
Answer
question embedding
LLM
semantic search
ranked results
PDF
PDF
PDF
Chunk of text (document)
Chunk of text (document)
Chunk of text (document)
Chunk of text (document)
embeddings
embeddings
embeddings
embeddings
vector store (knowledge base)
Pinecone

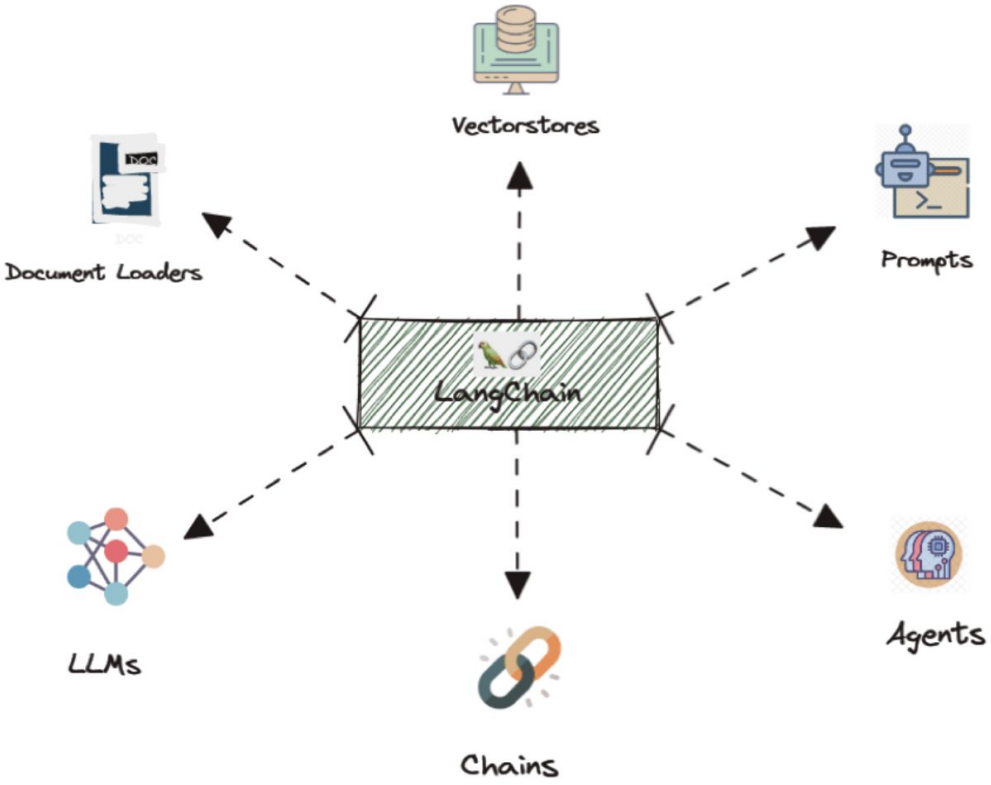
Chatbot Architecture

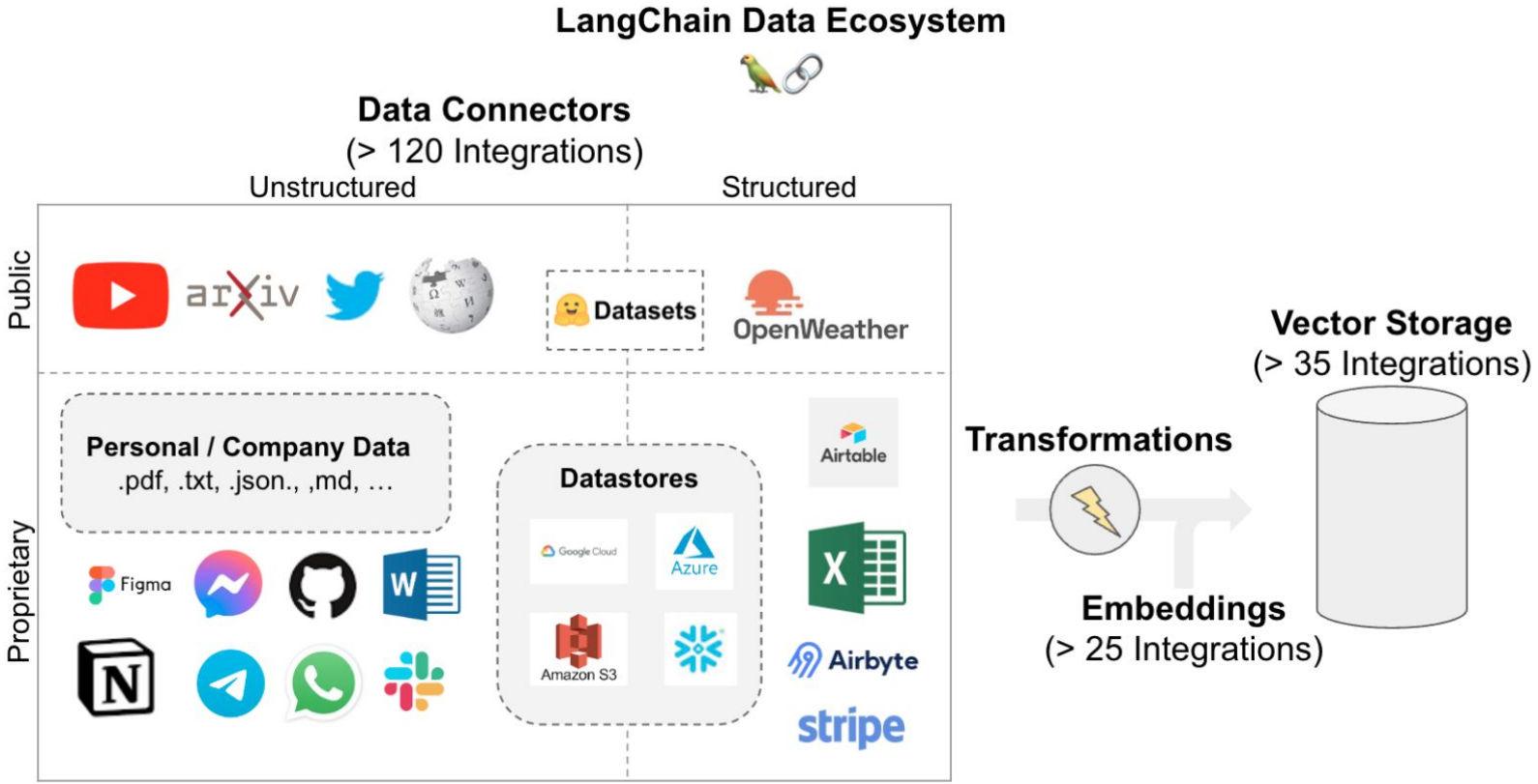


Original source: https://medium.com/@alejandro-ao:


PDF-Chatbot
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 branch
0 tags
Go to file
Add file
Code
nbeaudoin Updaint readme
422a800 last month
22 commits
__pycache__ Updating app and readme last month
.gitignore Adding .env last month
README.md Updaint readme last month
app.py Updating app and readme last month
app_image.png Updating app and readme last month
htmlTemplates.py Updating app and readme last month
requirements.txt Update requirements.txt per tutorial last month
About
Chatbot that ingests PDFs to create a tailored LLM solution
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release

LangChain


Vectorstores
Document Loaders
Prompts
LangChain
LLMs
Agents
Chains



Source: LangChain documentation

LangChain's place

Diving in, **LangChain reminds us of the early ML frameworks.**

≈

Source: https://www.unusual.vc/post/devtools-for-language-models

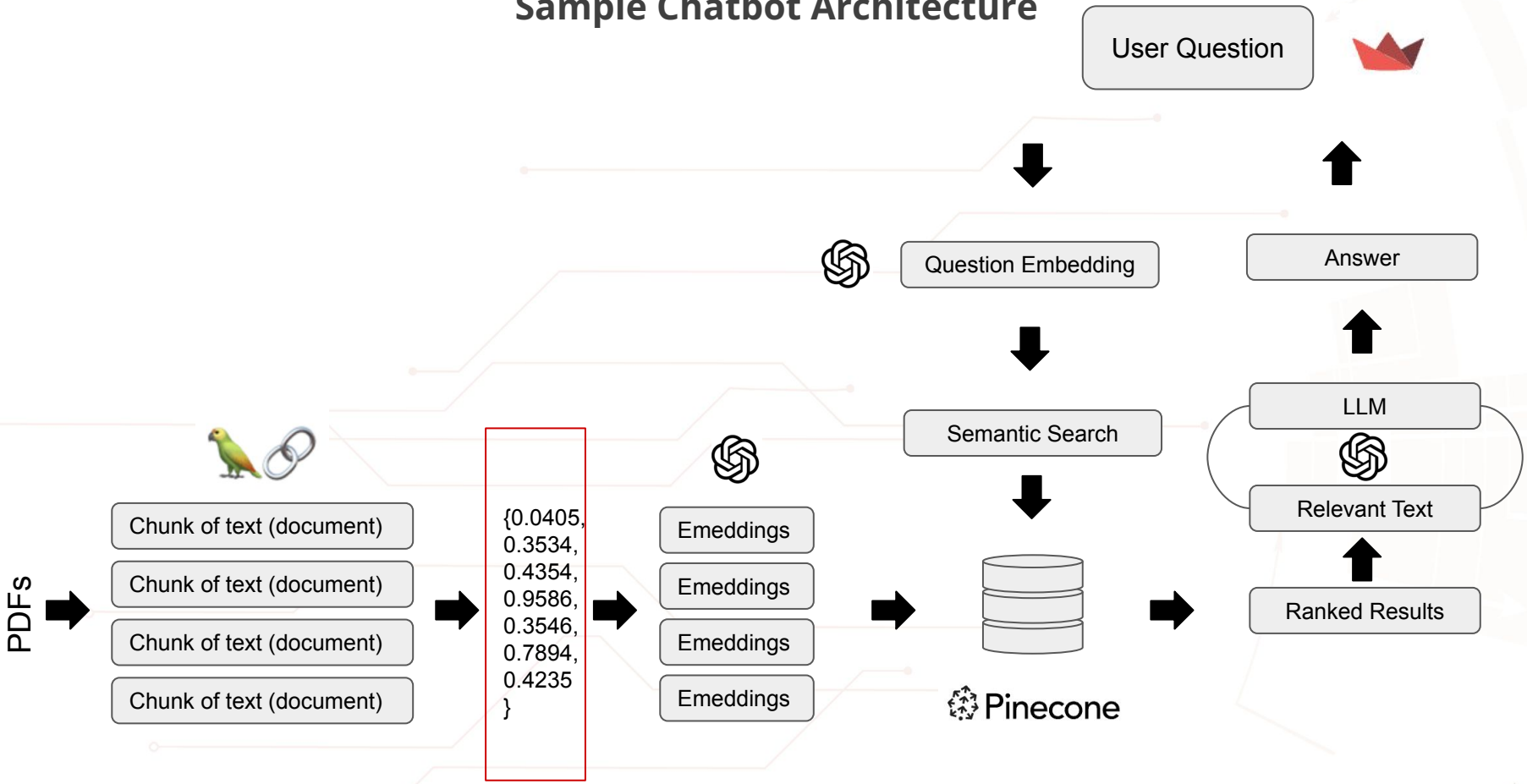
Types of Knowledge

- Parametric: Information that is learned by the model in pre-training and stored in the model weights
- Source: Information that is derived from documents at time of inference via the input prompt

So few-shot learning is deriving knowledge from the model based on source documents

Source: PineCone documentation

Sample Chatbot Architecture



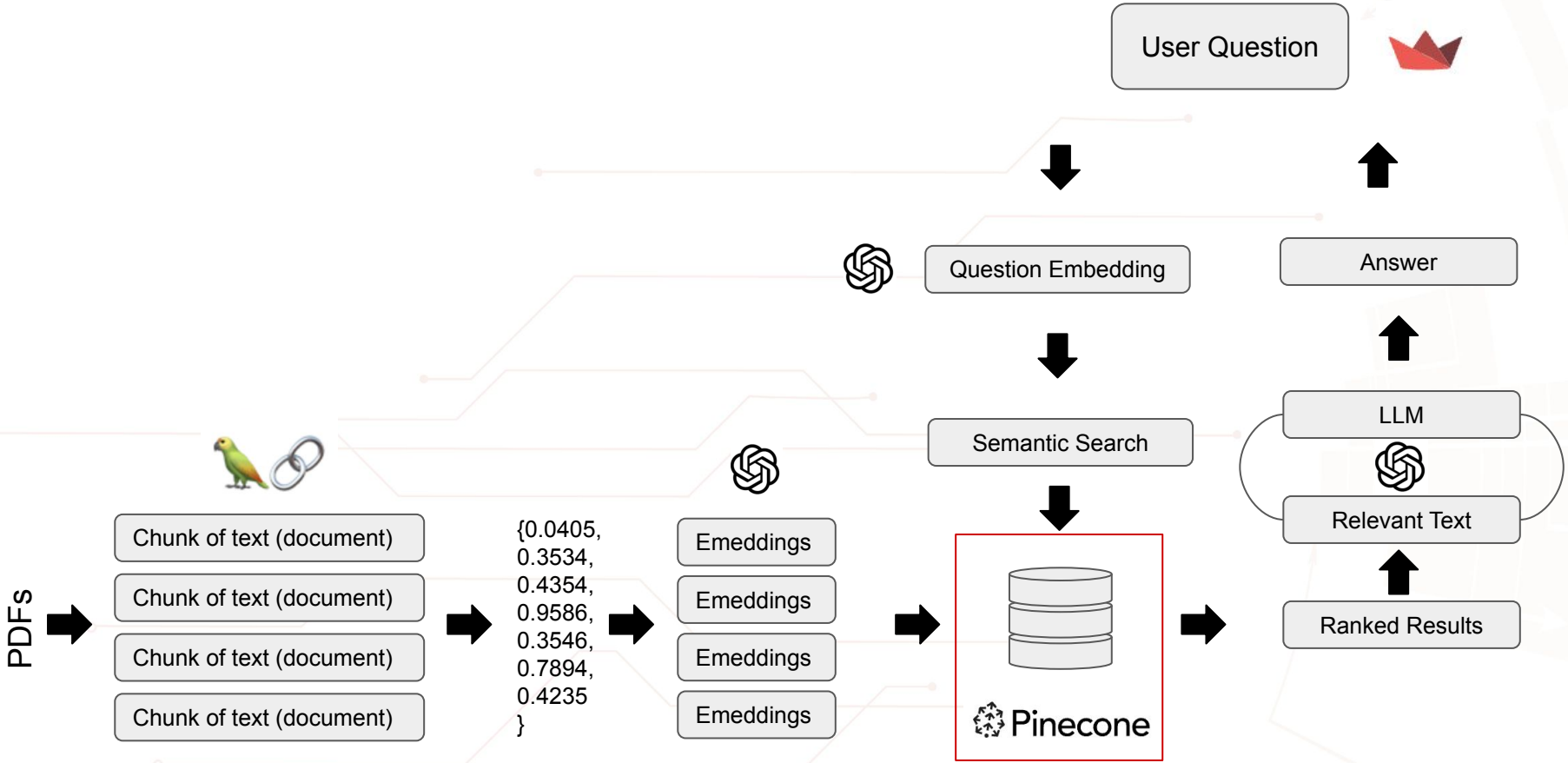
Remember embeddings?


```
{
  "data": [
    {
      "embedding": [
        -0.006929283495992422,
        -0.005336422007530928,
        ...
        -4.547132266452536e-05,
        -0.024047505110502243
      ],
      "index": 0,
      "object": "embedding"
    }
  ],
  "model": "text-embedding-ada-002",
  "object": "list",
  "usage": {
    "prompt_tokens": 5,
    "total_tokens": 5
  }
}
```

APIs cost money

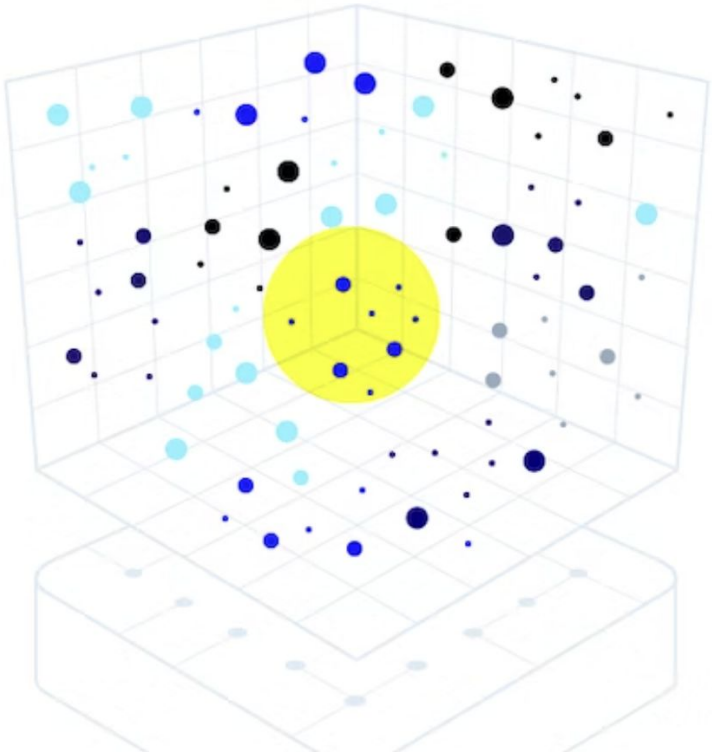
MODEL	ROUGH PAGES PER DOLLAR	EXAMPLE PERFORMANCE ON BEIR SEARCH EVAL
text-embedding-ada-002	3000	53.9
-davinci--001	6	52.8
-curie--001	60	50.9
-babbage--001	240	50.4
-ada--001	300	49.0

Source: OpenAI documentation

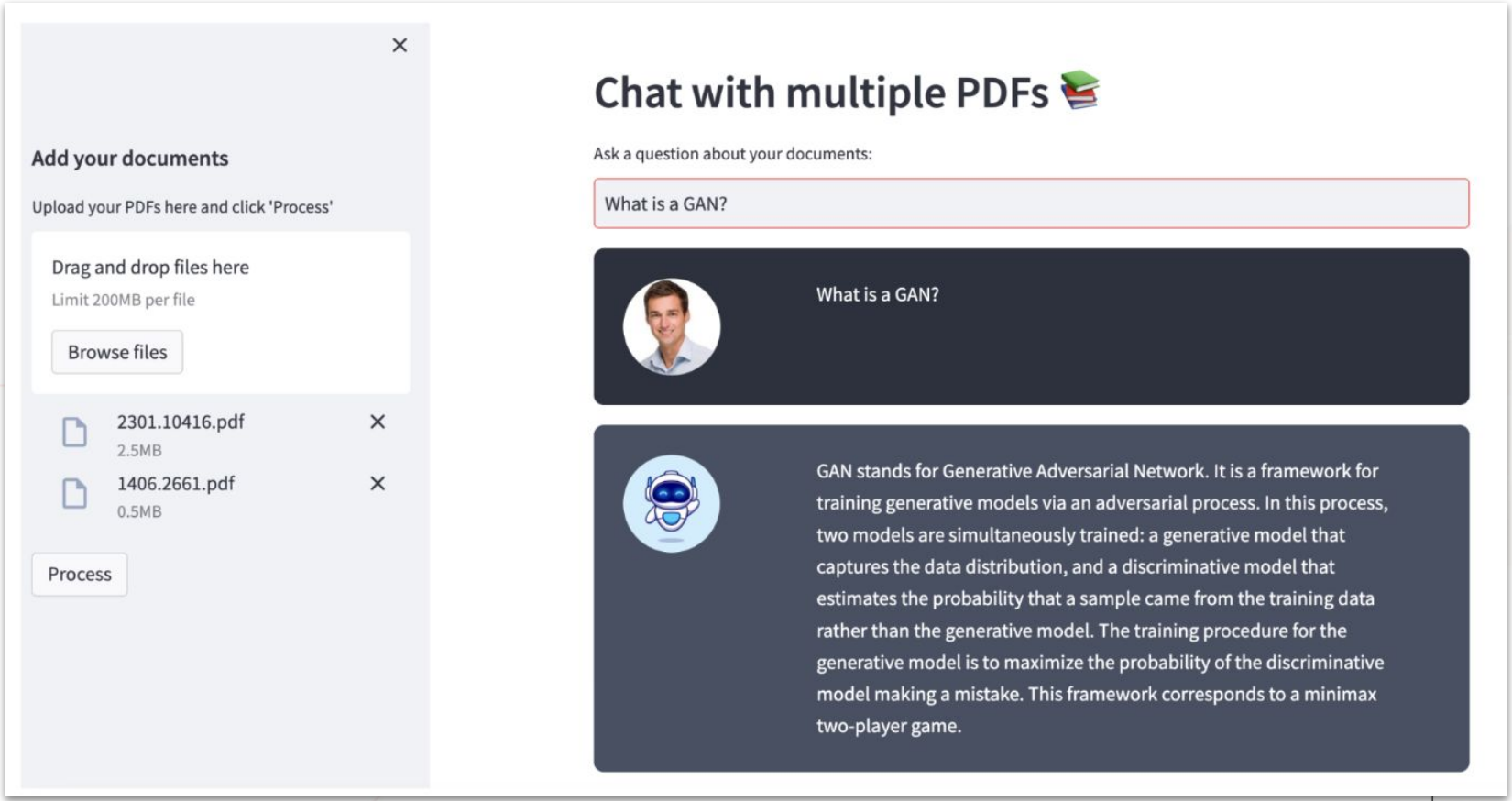

User Question
Question Embedding
Answer
Semantic Search
LLM
Relevant Text
Ranked Results
PDFs
Chunk of text (document)
Chunk of text (document)
Chunk of text (document)
Chunk of text (document)
{0.0405,
0.3534,
0.4354,
0.9586,
0.3546,
0.7894,
0.4235
}
Emeddings
Emeddings
Emeddings
Emeddings
Pinecone

Back to Vector DBs!

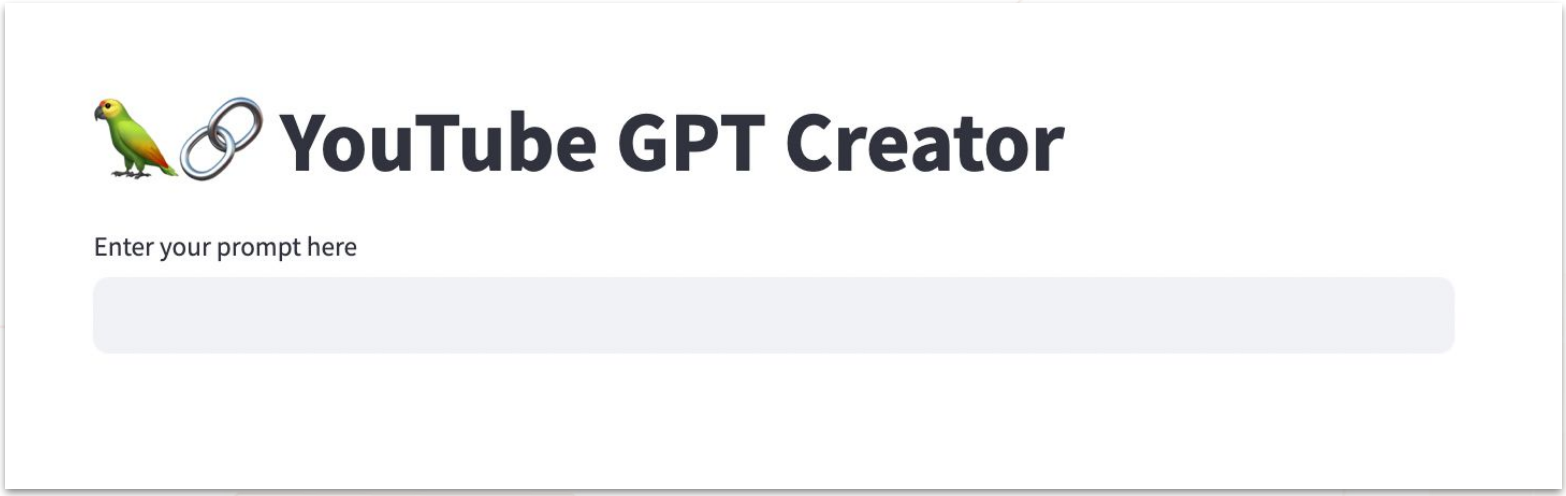
- Semantic search not same as simple index look-up
- Time consuming to get exact matches
- Approximation needed
- ANN algorithm used



Sample LLM Application

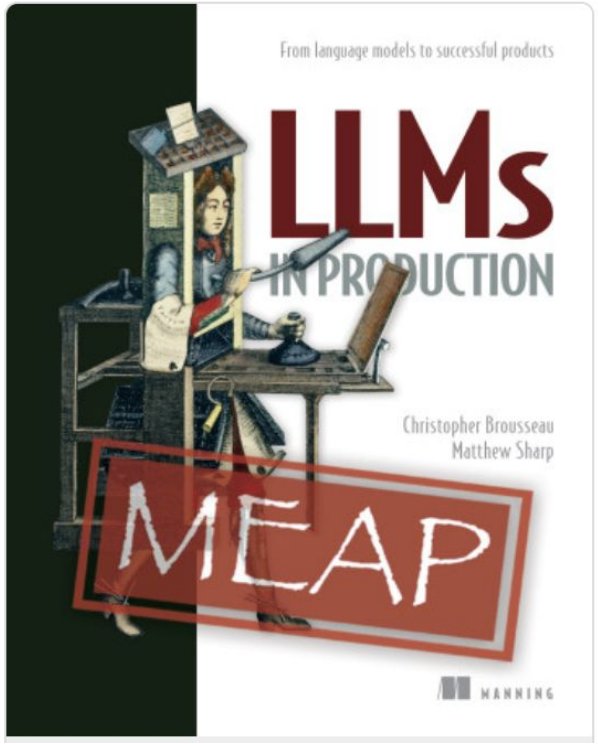


Sample LLM Gone Wrong



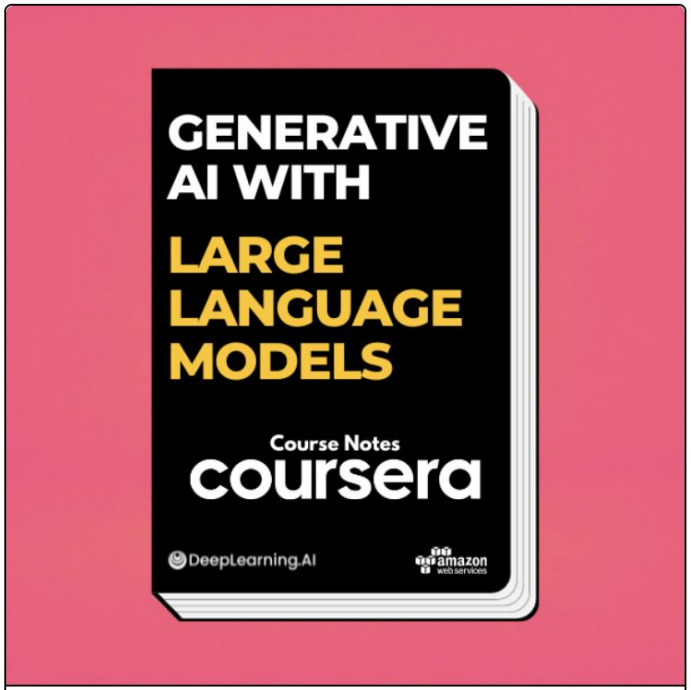
Source:

APPENDIX

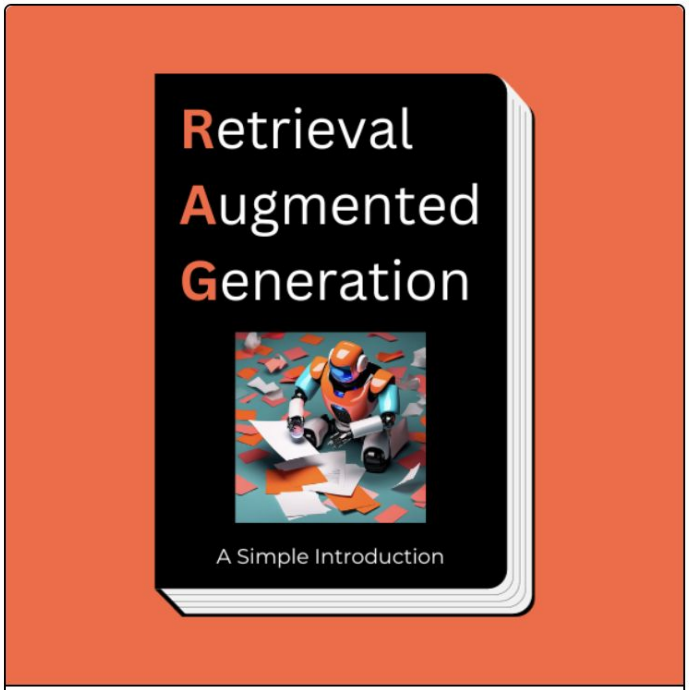


https://www.manning.com/books/llms-in-production

https://abhinavkimothi.gumroad.com/



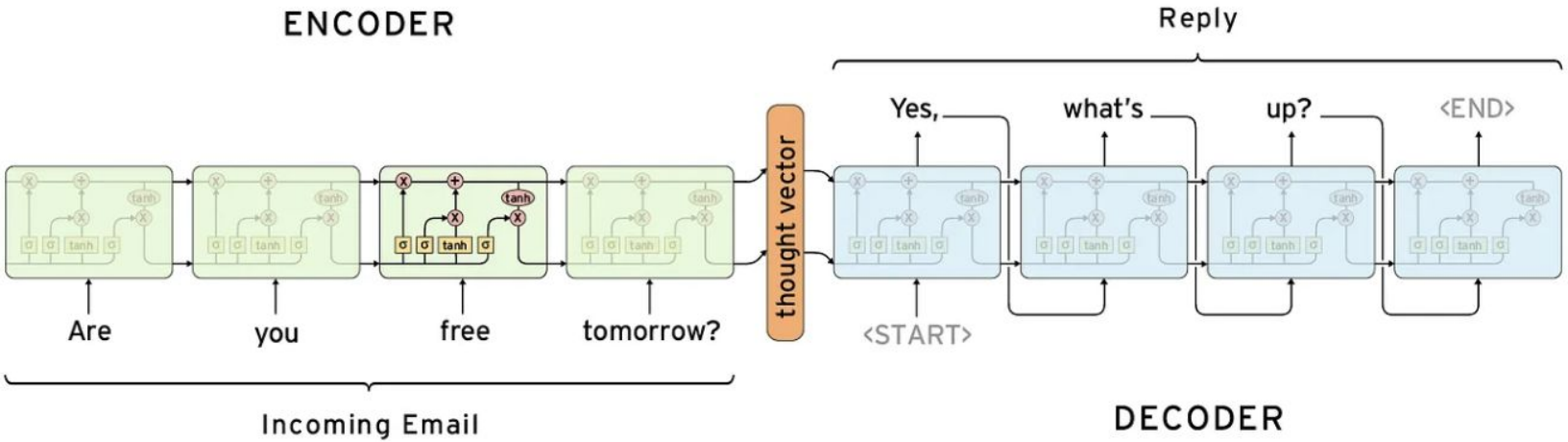
Generative AI with Large Language Models (Coursera Course Notes)



Retrieval Augmented Generation - A Simple Introduction

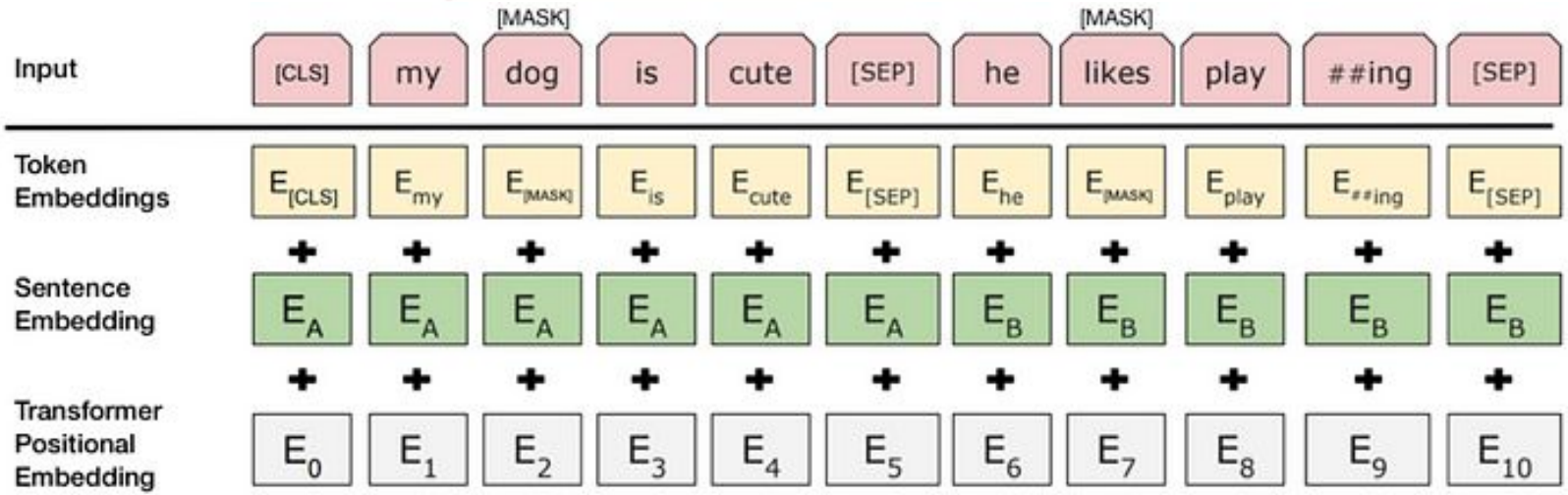
Encoder-Decoder architecture

Advanced level



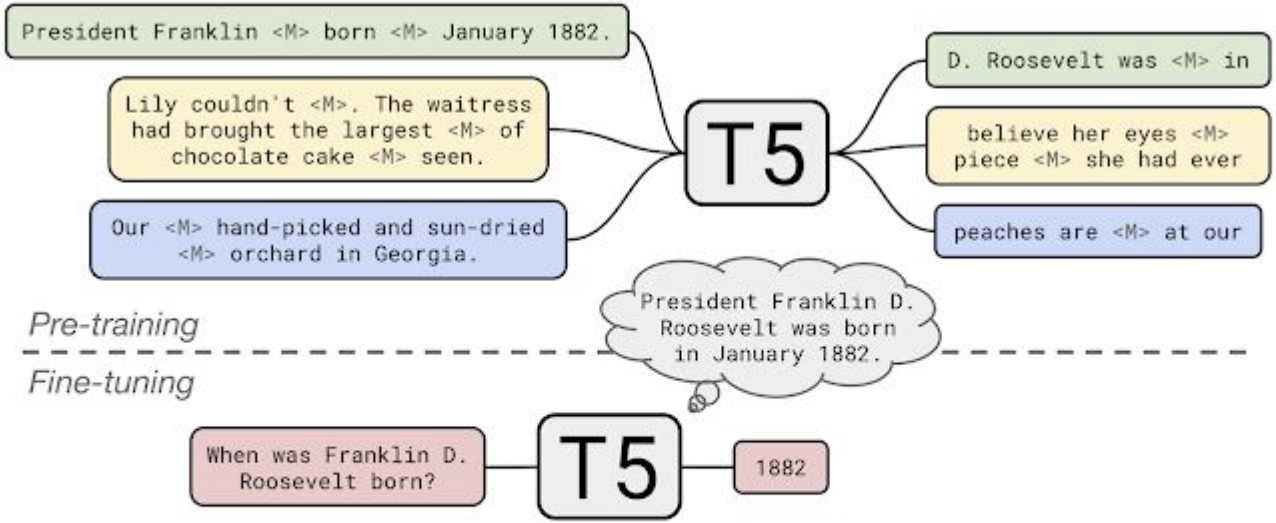
Source: https://medium.com/nerd-for-tech/encoder-decoder-model-for-machine-translation-8a90be12ac32

Bidirectional Encoder Representations from Transformers



Source: Bert Paper - https://arxiv.org/pdf/1706.03762.pdf

T5: Text-to-Text Transfer Transformer



During pre-training, T5 learns to fill in dropped-out spans of text (denoted by <M>) from documents in C4. To apply T5 to closed-book question answer, we fine-tuned it to answer questions without inputting any additional information or context. This forces T5 to answer questions based on "knowledge" that it internalized during pre-training.

Source: https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html