Natural language processing
&
language modelling

Content

How do we model:

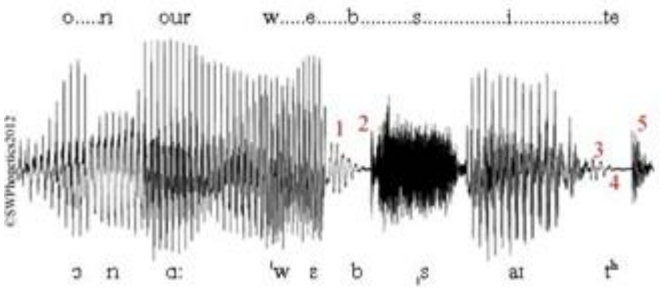
- Language
- Words
- Sequences
- Context

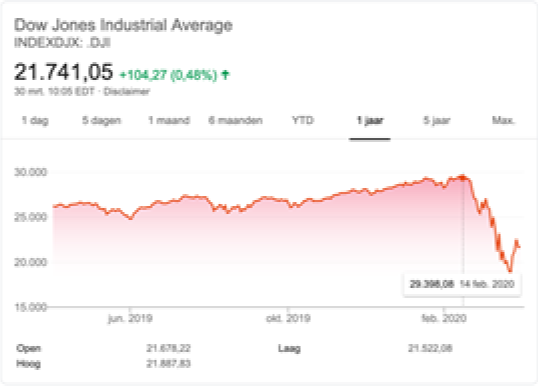
Transformers

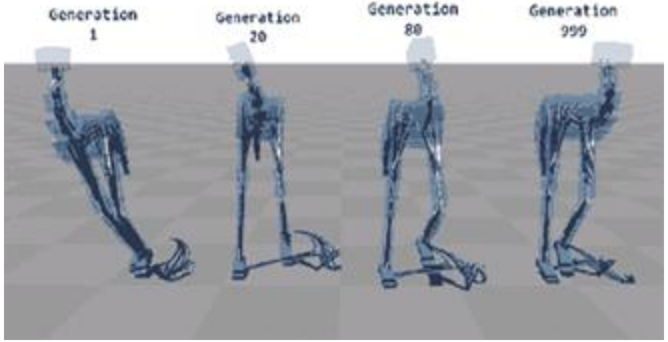
LLMs

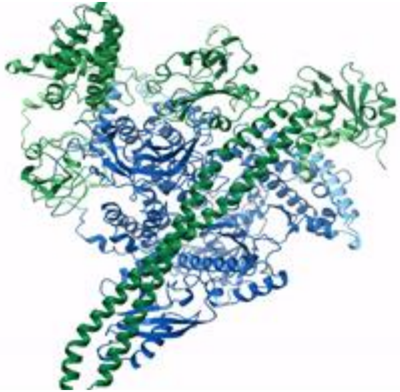
Sequential data

Universe seems to be inherently sequential





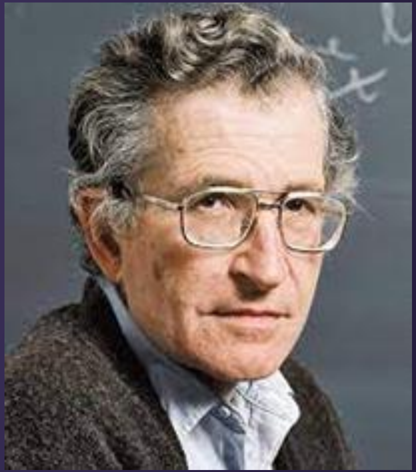




How can we model this?

Is there one (sequential) model to rule them all?

How does modelling language differ from other sequences?



> *Language is a process of free creation; its laws and principles are **fixed**, but the manner in which the principles of generation are used is free and **infinitely varied.***
>
> ~ Noam Chomsky

Sequential data

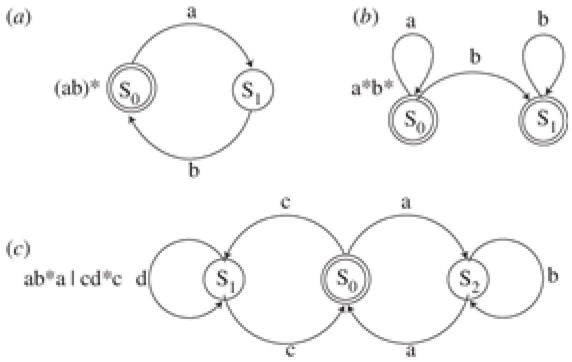
Modelling *natural language*

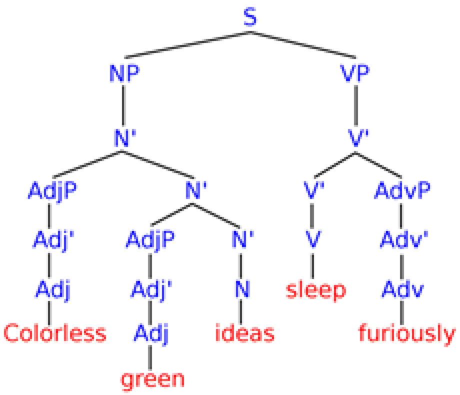
We have the innate ability for language (Universal Grammar) Syntactic Structures

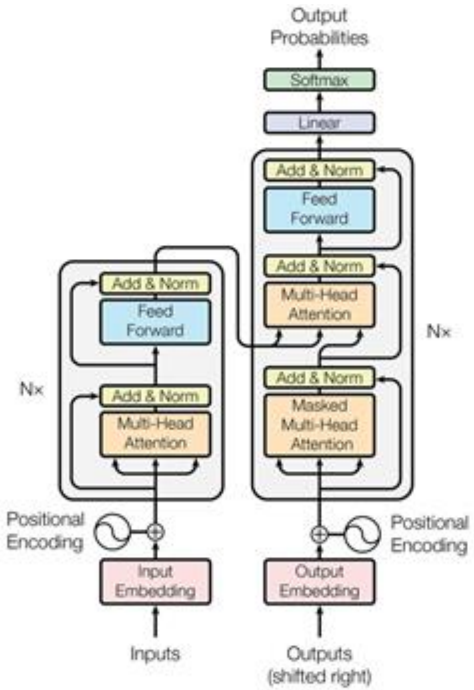
(Chomsky, N. 1957)

Different ways to model language:

- Finite State Automata
- Generative Grammars
- Deep Learning + Data
- ….

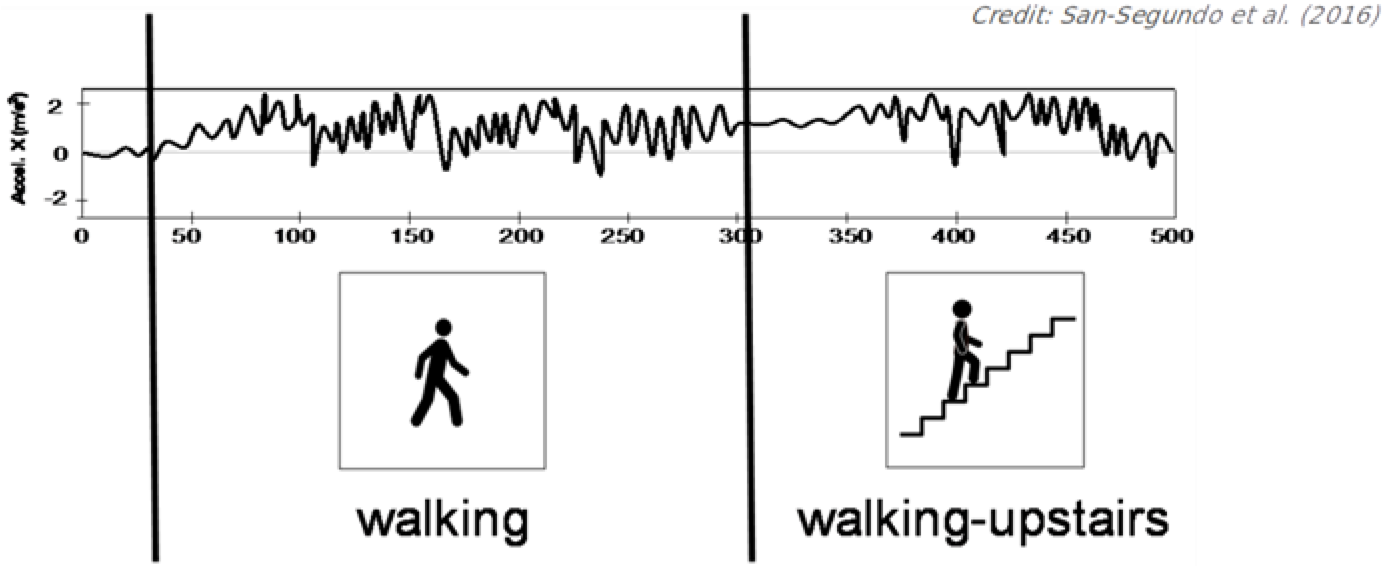






SURF

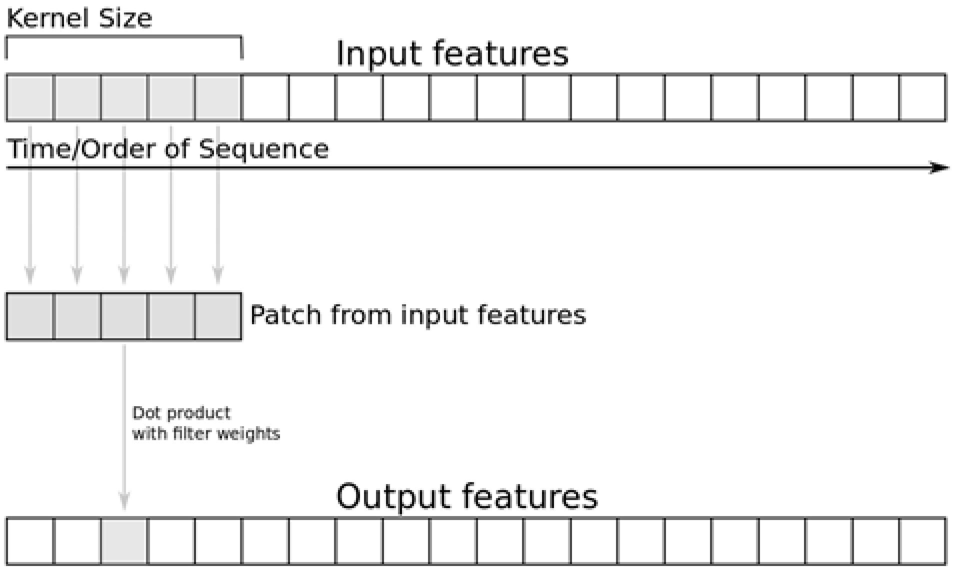
Human Activity Recognition



Human Activity Recognition

Can we use a CNN?

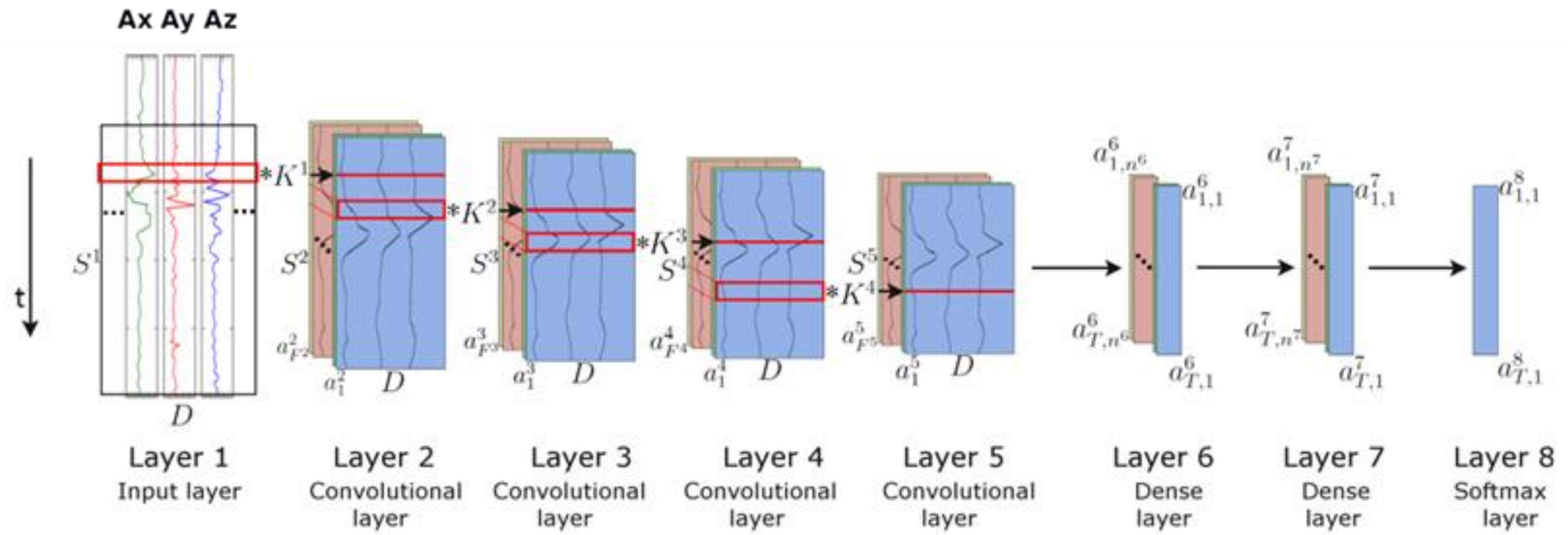
CNNs work here because they take **nearby** tokens into account



Human Activity Recognition

Can we use a CNN?

CNNs work here because they take **nearby** tokens into account



SURF

How to model language?

Language model objective:

📊 Model the **probability** of the next symbol given the previous context by creating a probability distribution over language

➡️ Language models like ChatGPT model language *left to right:* one word at a time

$$P(w_1, w_2, \ldots w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ldots P(w_n \mid w_1, \ldots w_{n-1})$$

How to model language?

Language model objective:

📊 Model the **probability** of the next symbol given the previous context by creating a probability distribution over language

➡️ Language models like ChatGPT model language *left to right:* one word at a time

$$P(w_1, w_2, \ldots w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ldots P(w_n \mid w_1, \ldots w_{n-1})$$

The cat

How to model language?

Use CNNs for modelling language?

CNN's context size is small and does **not *remember*** past context

Language is very noisy and heavily (long-distance) context dependent

$$P(w_1, w_2, \ \dots \ w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ \dots \ P(w_n \mid w_1, \ \dots \ w_{n-1})$$

The cat

Neural representations

High Performance Machine Learning Group

SURF

How do we represent words?

Language is **discrete**!

> “One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

one
cat
ate
tuna
chicken
weird
human
a

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

one	0
cat	1
ate	2
tuna	3
chicken	4
weird	5
human	6
a	7

SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

one
cat
ate
tuna
chicken
weird
human
a

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

$$[0.122, 1.54, -0.193]$$

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Easy to **compose** and easy to **vectorize**

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

Embedding

one
cat
ate
tuna
chicken
weird
human
a

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

$[0.122, 1.54, -0.193]$

$\vdots$

$[0.567, 0.32, 1.243]$

How do we represent words?

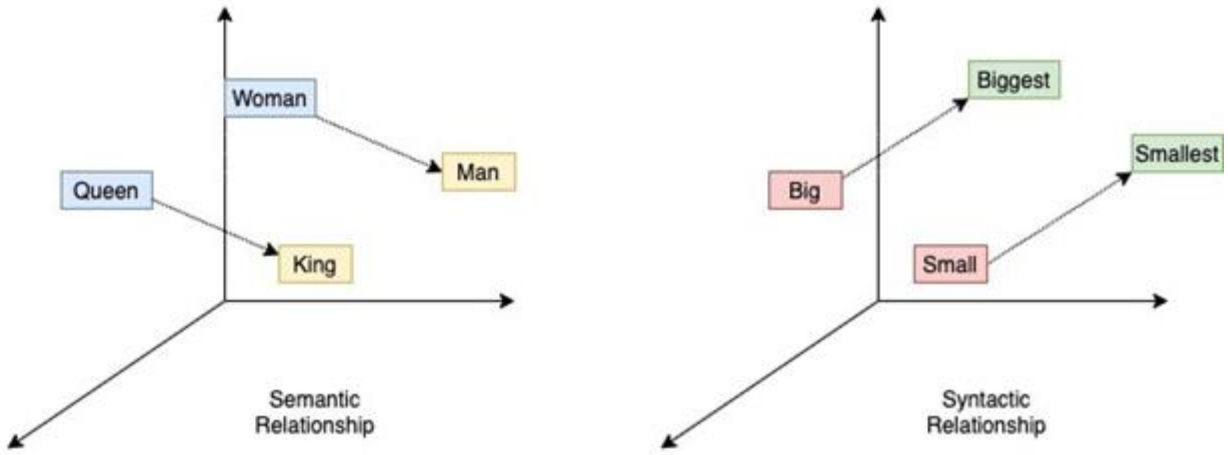
- Most basic: one hot encoding [0, 0, 1, 0, ….., 0, 0]
  ❌ No meaningful distance between vectors

- Word2Vec

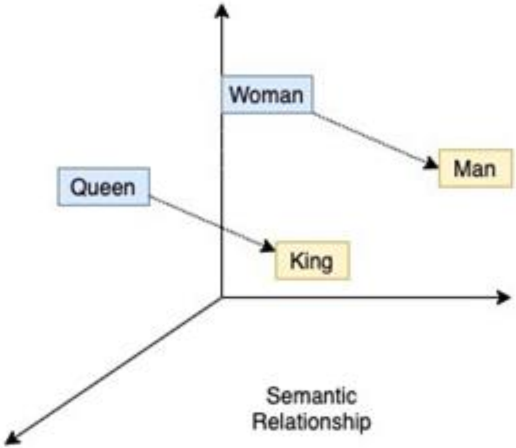
Distributional hypothesis

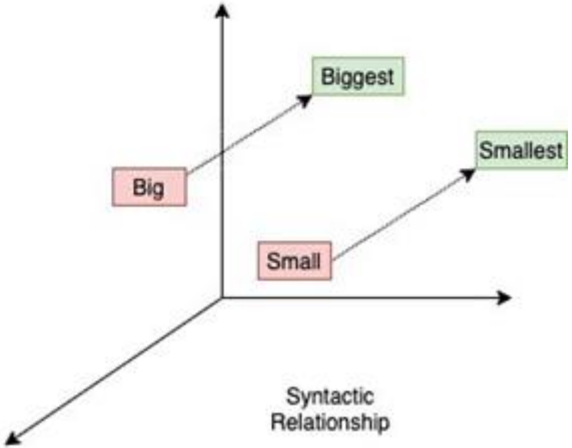
- The **context** surrounding a given word provides information about its meaning
- Words are similar if they share similar **contexts** (semantic + syntactic)
- Semantic similarity ≈ distributional similarity

How do we represent words?



How do we represent words?





King - Man + Woman = Queen

Each word has a numerical vector representation learnt by a language model

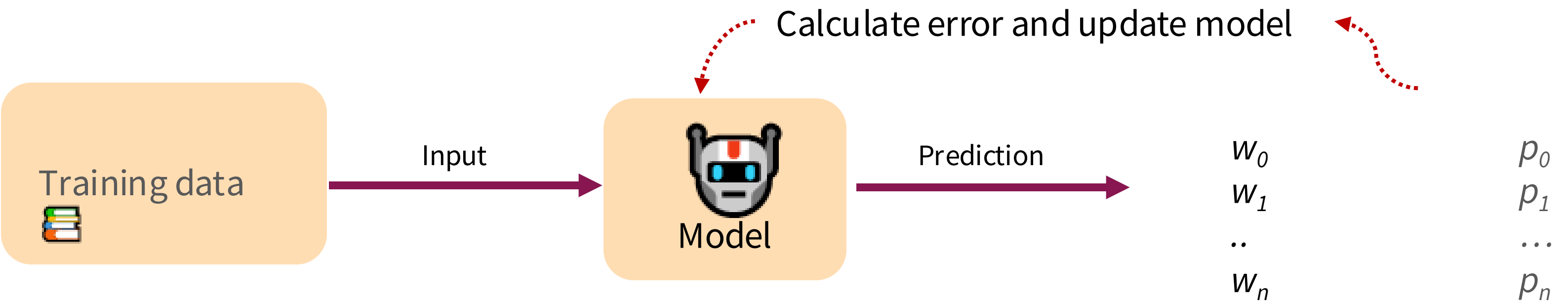
How do we represent words?

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Mikolov et al. 2013. *Efficient Estimation of Word Representations in Vector Space*

How do we model sequences?

Unsupervised learning



How do we model sequences?

Unsupervised learning

Training data

The monkey ate that banana because it was…

Input

Model

Prediction

Yellow	*0.8*
Hungry	*0.1*
There	*0.1*

Calculate error and update model

True output

SURF

The Transformer

High Performance Machine Learning Group

SURF

Attention

Weighted average over hidden states

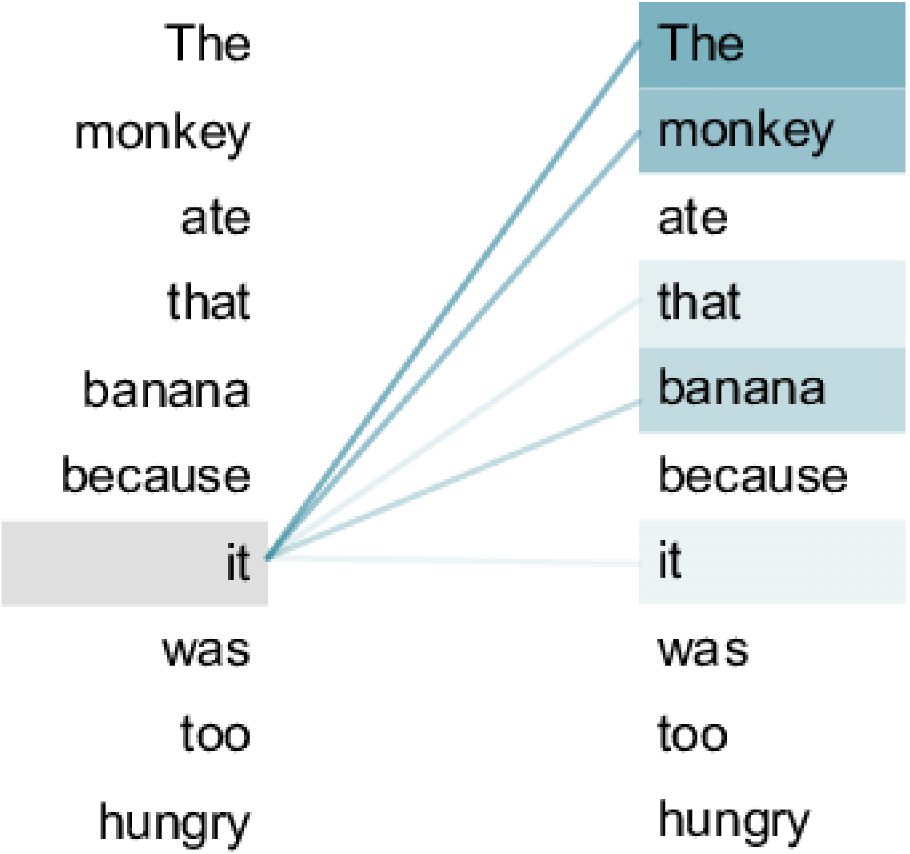


Image source

Attention is all you need: self-attention

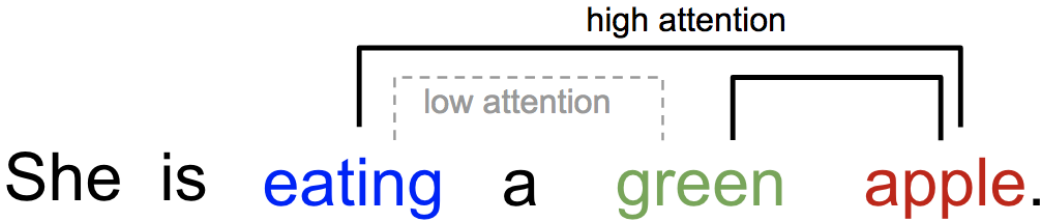


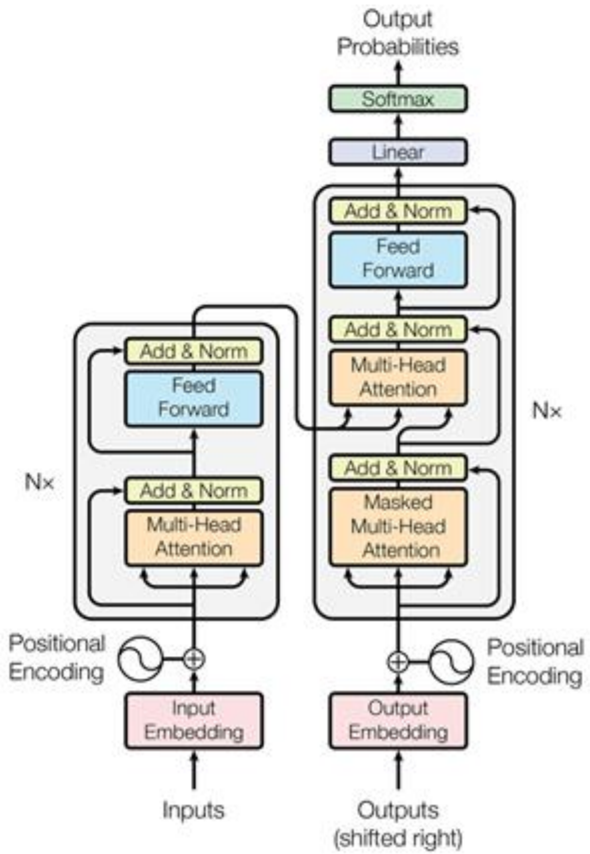
Image source

Transformers

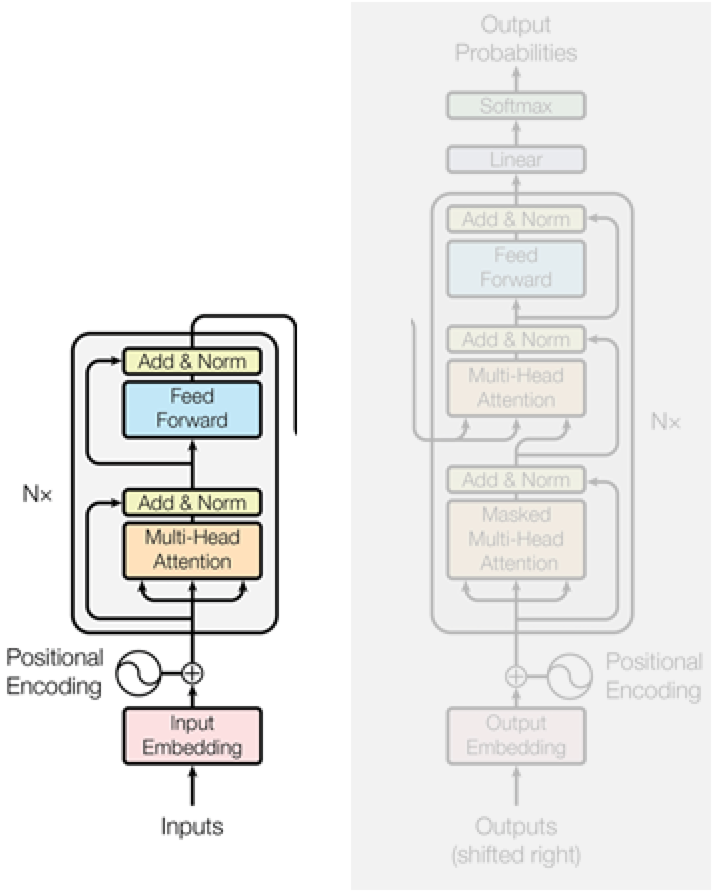
What is the **importance** of every element in the input sequence with respect to **itself?**

Every word directly connected to every other word

More parallelization!



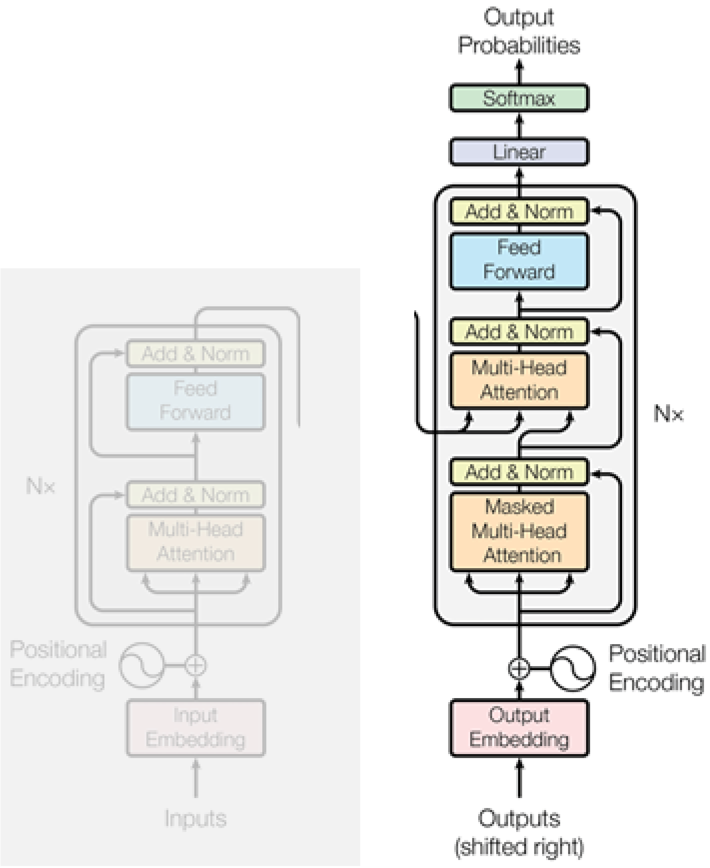
Dissection of the Transformer



Encoder

Understanding the input

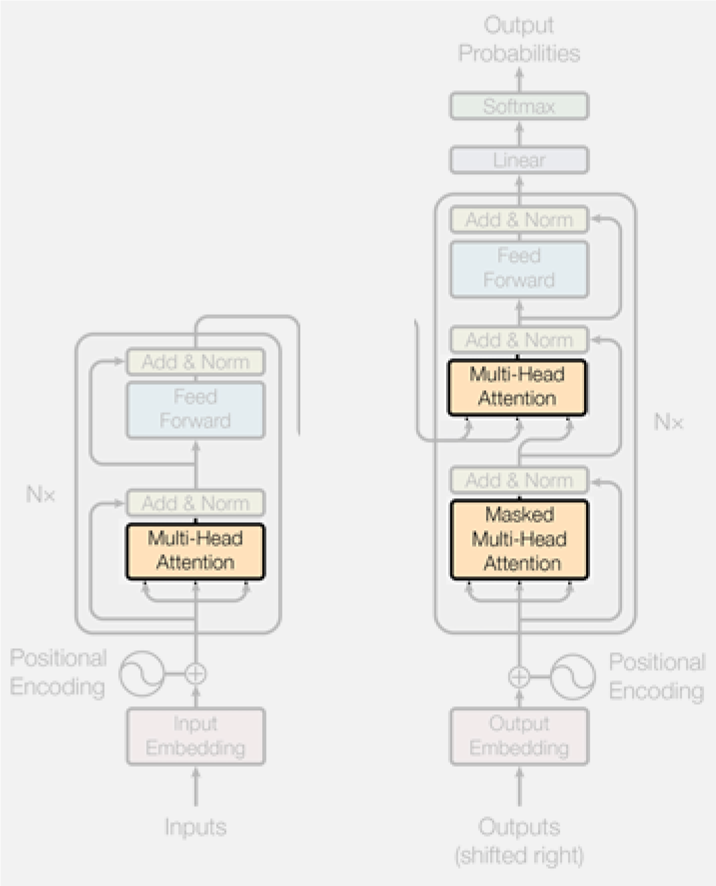
Dissection of the Transformer

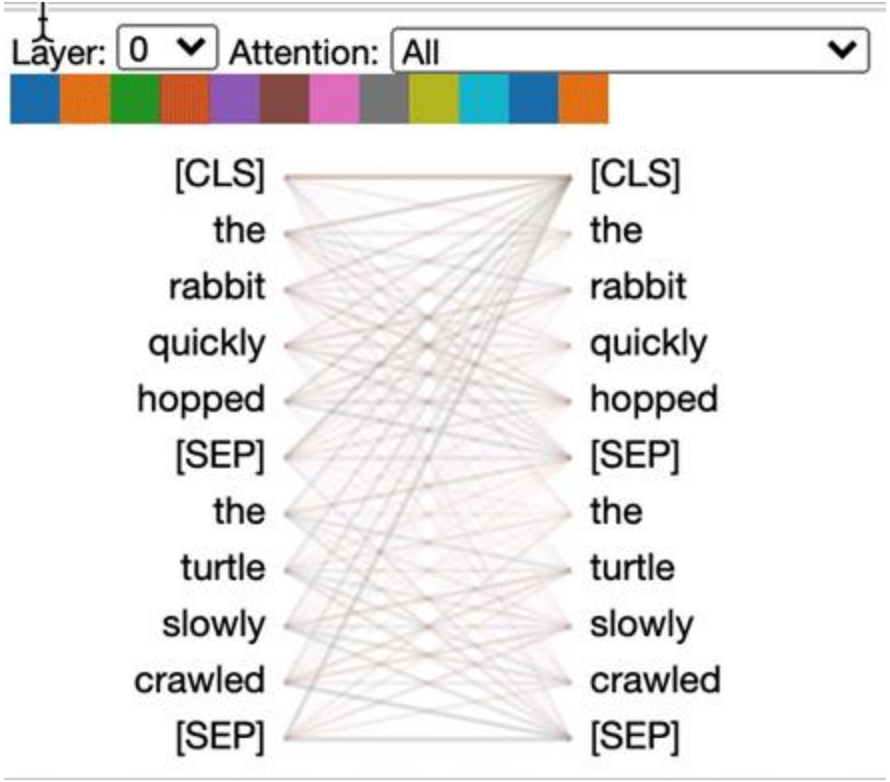


Decoder

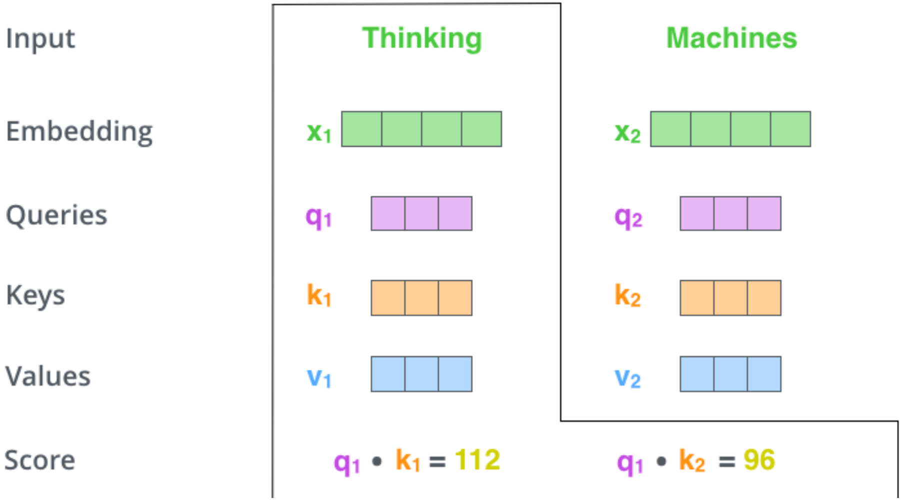
Generating the output

Dissection of the Transformer





Self Attention



SURF

Image source

Self Attention

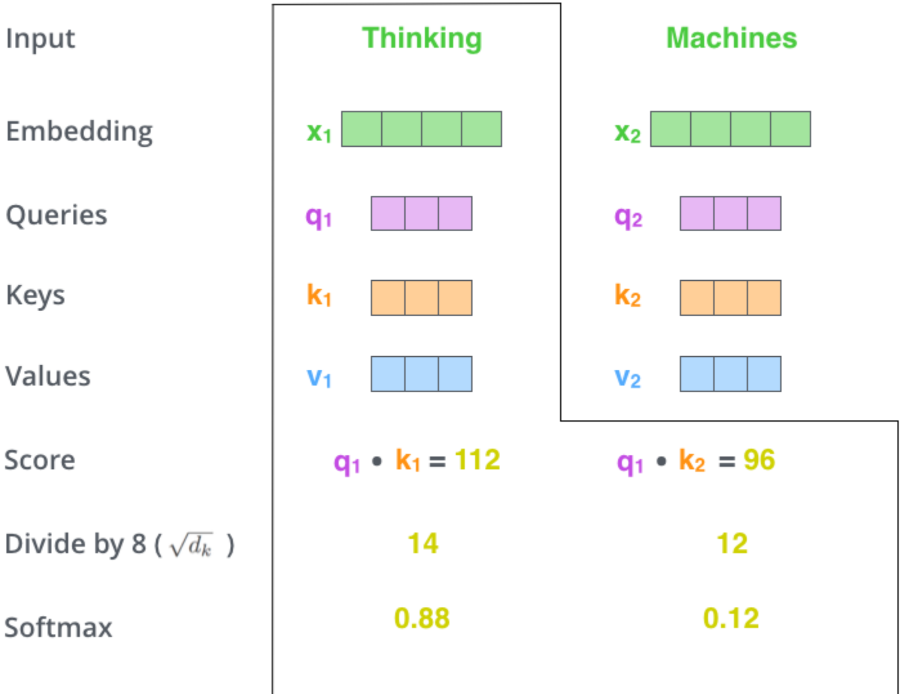


Image source

Self Attention

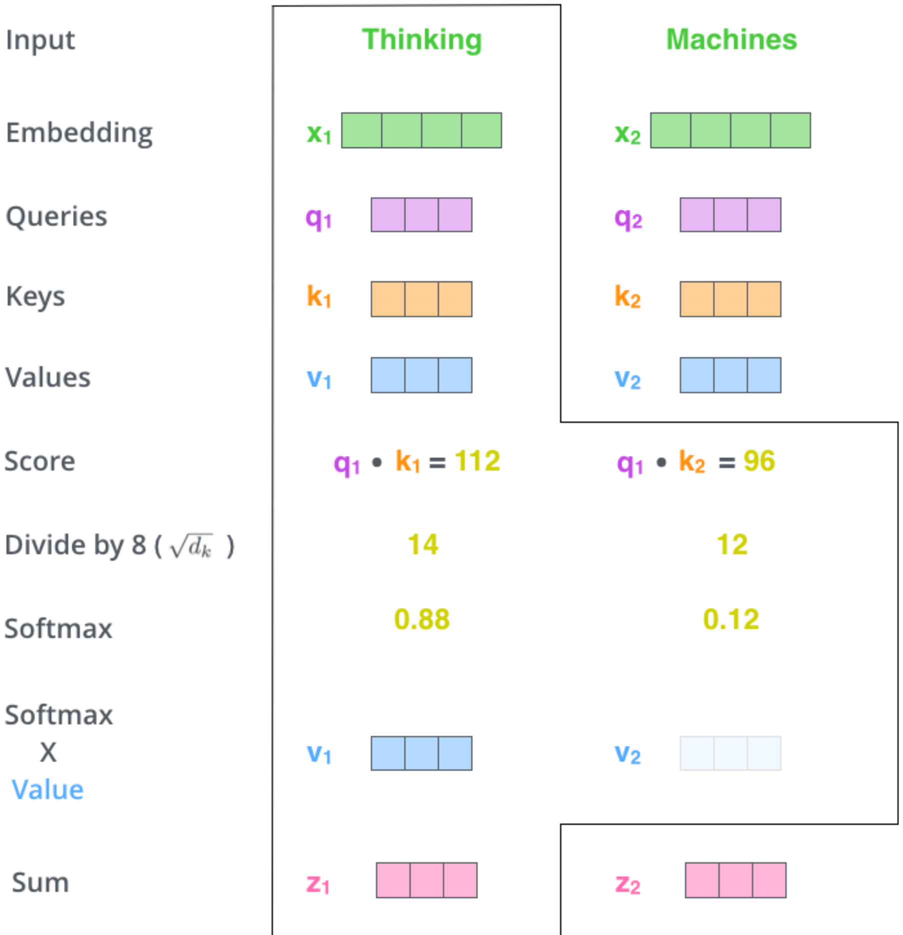
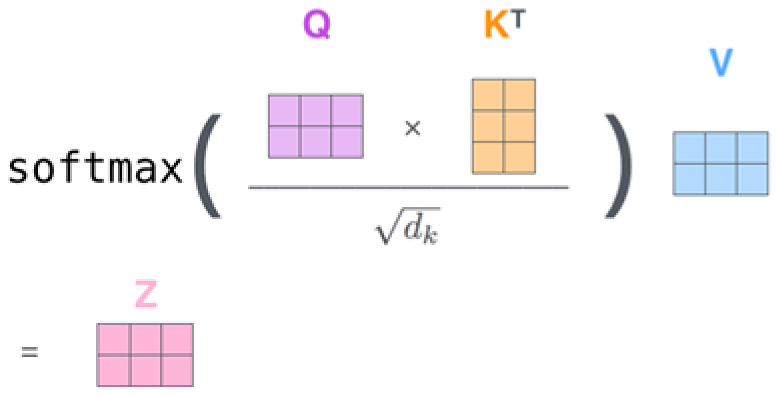


Image source

Self Attention



$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

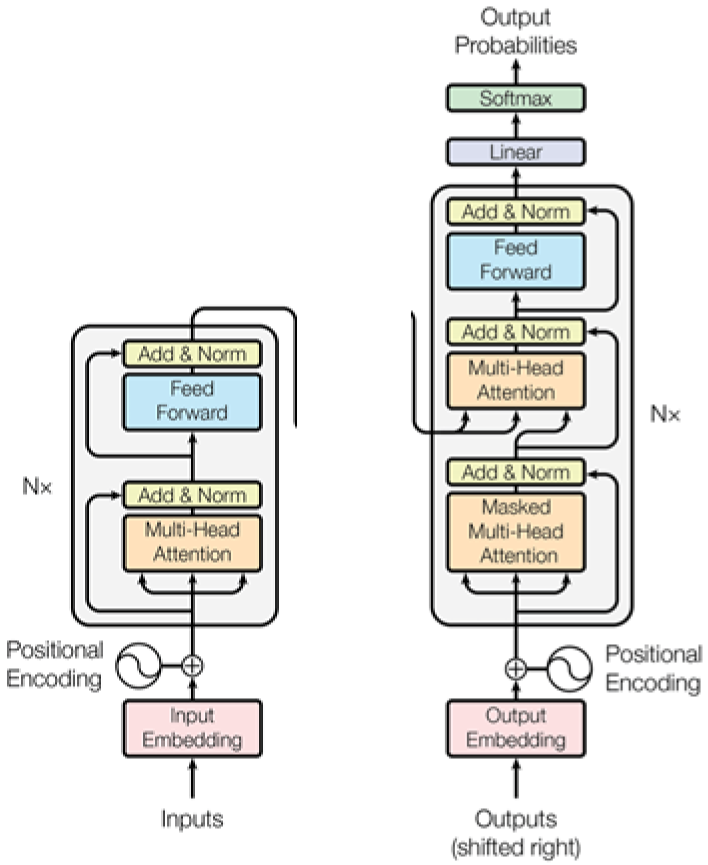
$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix} \qquad \frac{e^{x_i}}{\sum_{j=0}^{K} e^{x_j}}$$

Self Attention

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix} \qquad \frac{e^{x_i}}{\sum_{j=0}^{K} e^{x_j}} \qquad \begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix}$$

Dissection of the Transformer



Layer Normalization

Skip Connections

Positional Encoding

Byte Pair Encodings

Layer Stacking

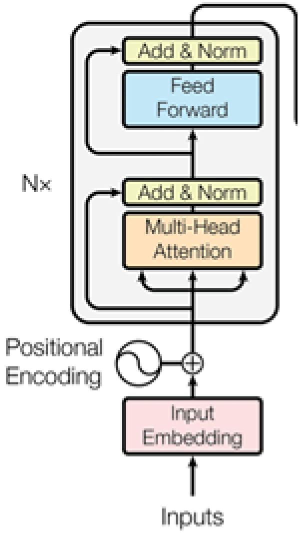
Cross Attention

Model types

Encoder

BERT

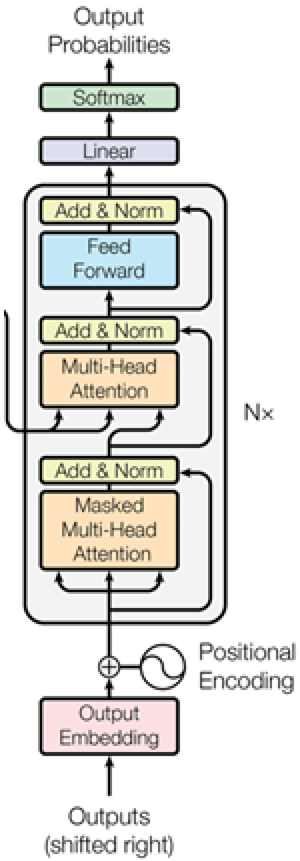
Sequence classification



Decoder

GPT

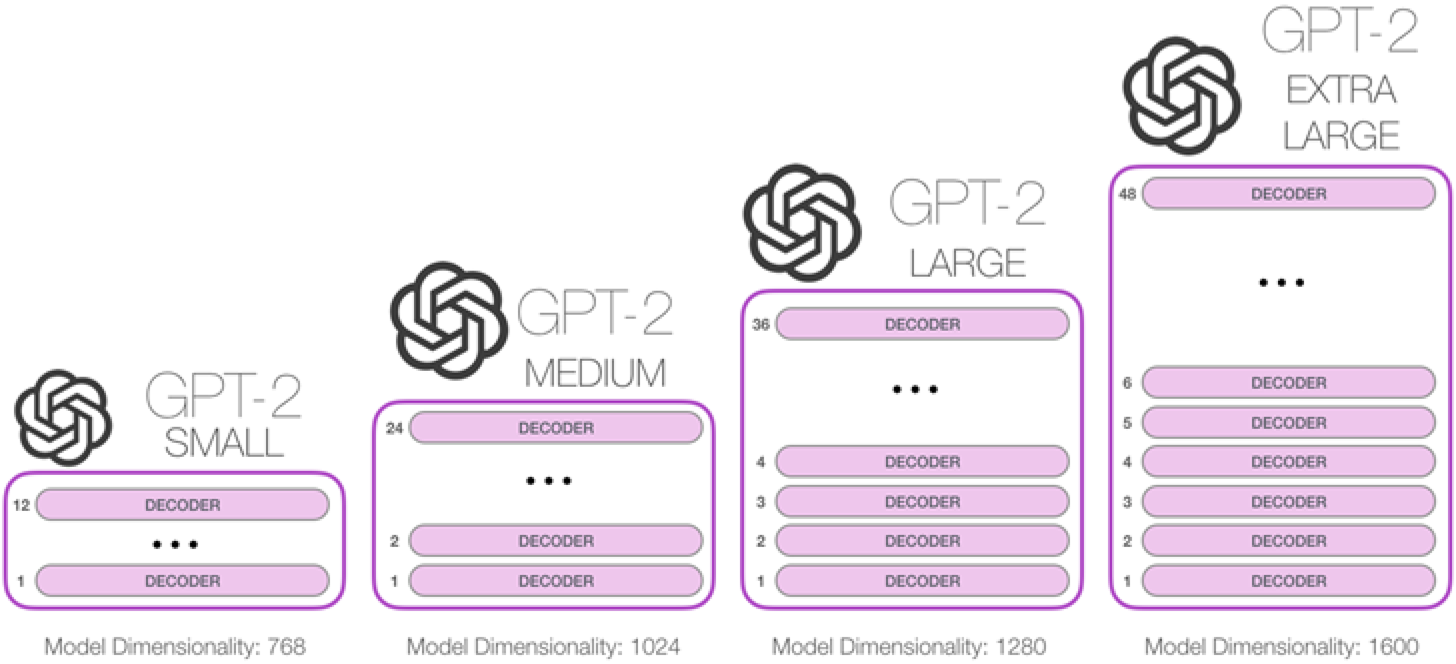
Text generation



Encoder-decoder: T5

Sequence-to-sequence e.g. machine translation

Generative Pre-Trained Transformers



From language models to chatbots

GPT: language model
ChatGPT: *aligned* language model

Alignment: what type of response is desirable? How can we make model behaviour be *aligned* with human norms and values?

Alignment and instruction finetuning

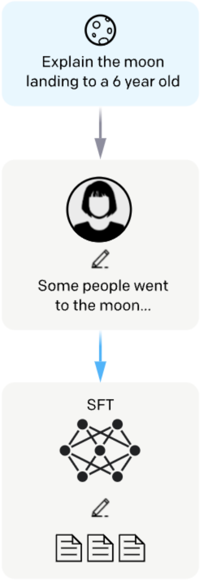
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



SURF

Alignment and instruction finetuning

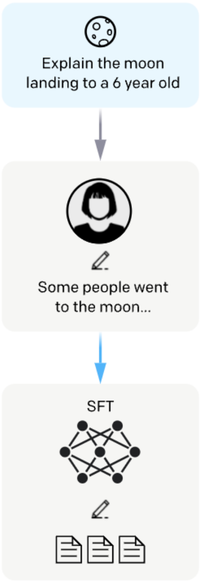
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



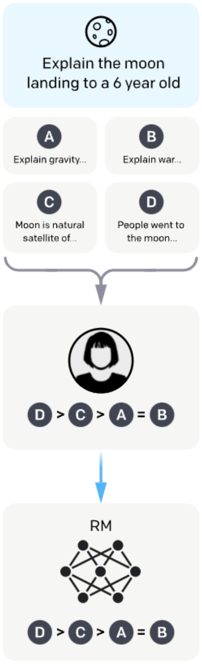
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Alignment and instruction finetuning

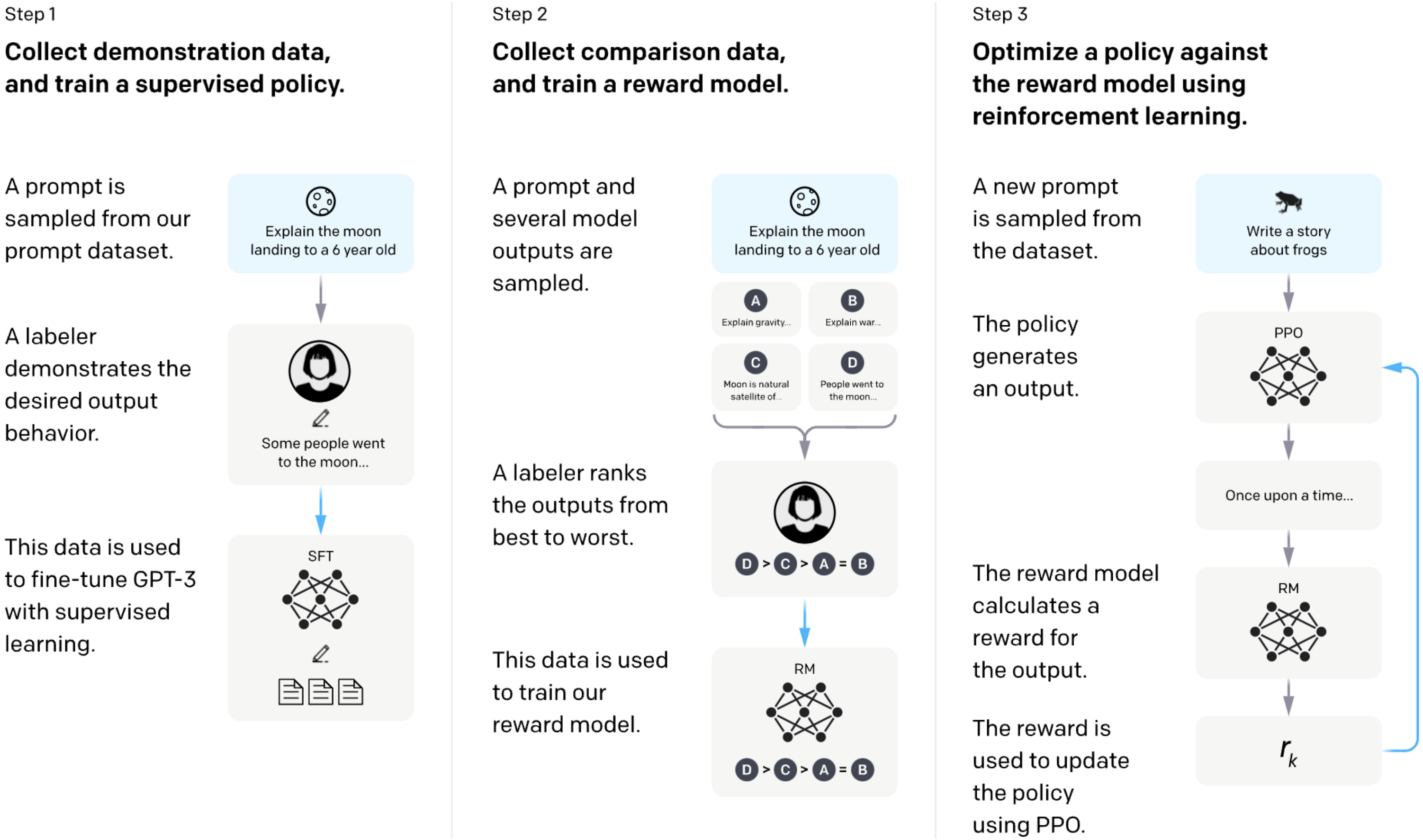
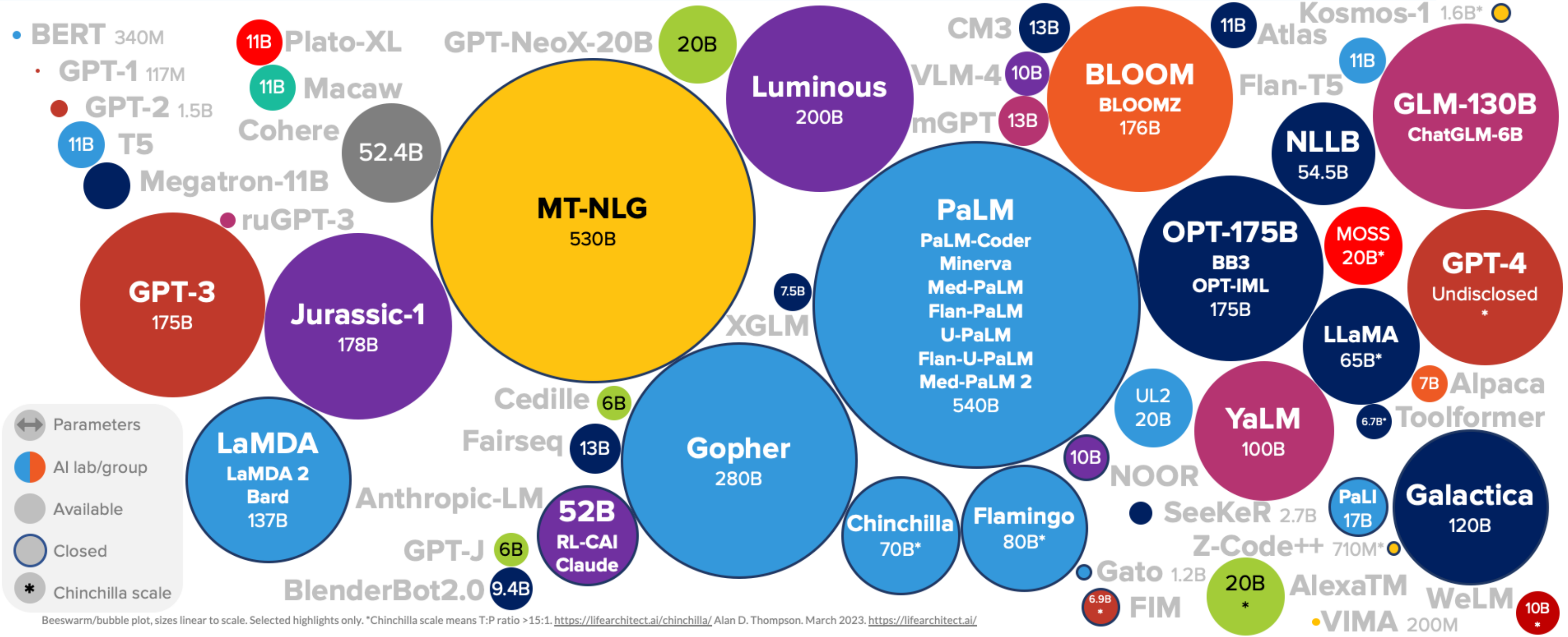


Image source

Scaling Transformers

High Performance Machine Learning Group

Cambrian Explosion of LMs



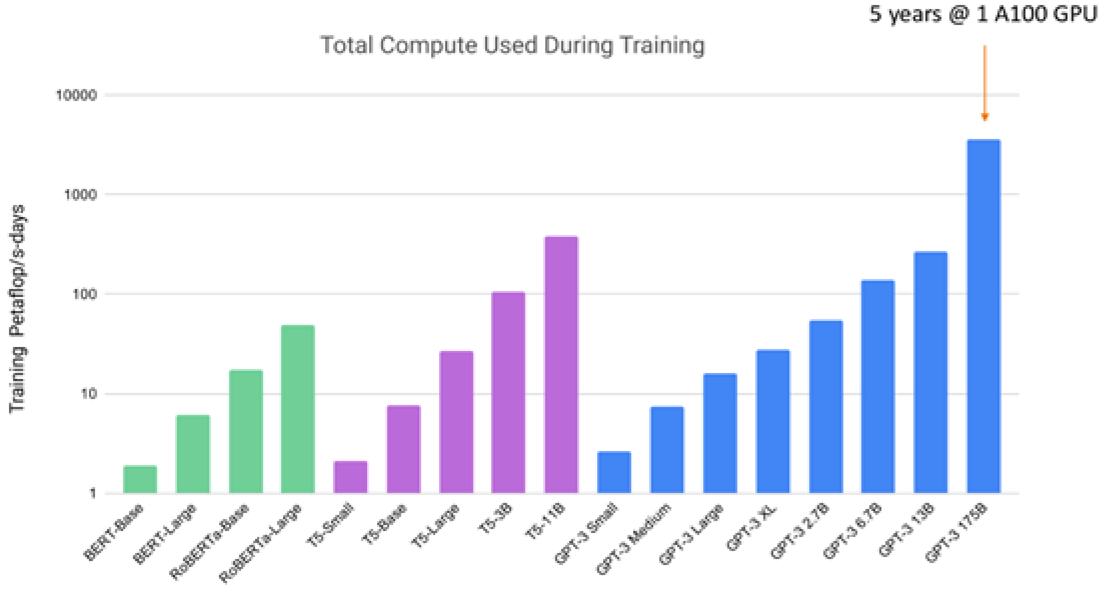
In comparison: the “big” model in original attention paper had only 213M parameters

Image source

Transformer Explosion

GPT-3 Trained on ~350 Billion tokens

GPT-3 has 175B parameters!



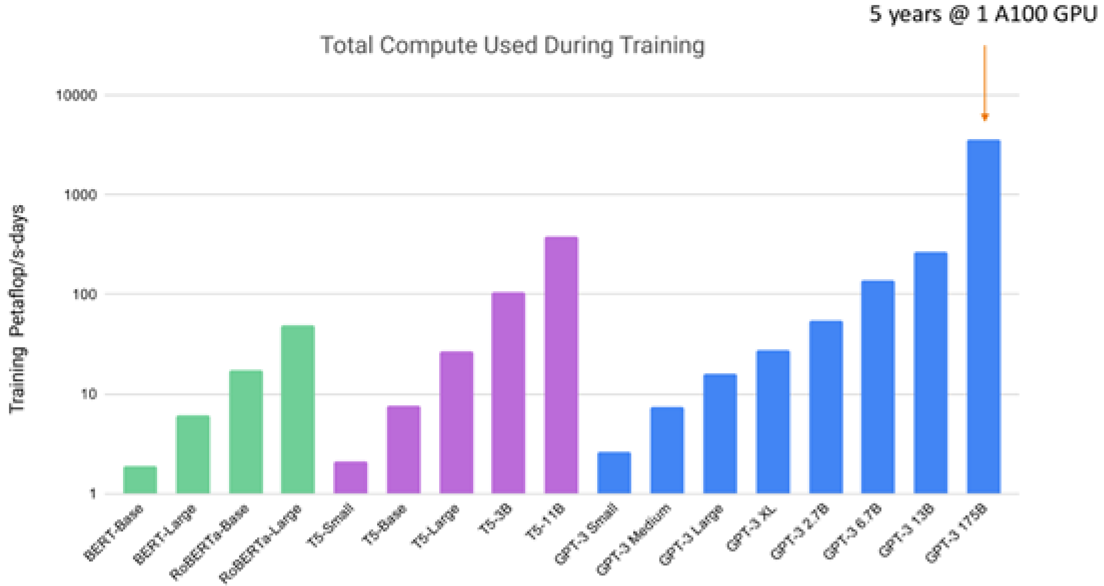
Transformer Explosion

GPT-3 Trained on ~350 Billion tokens

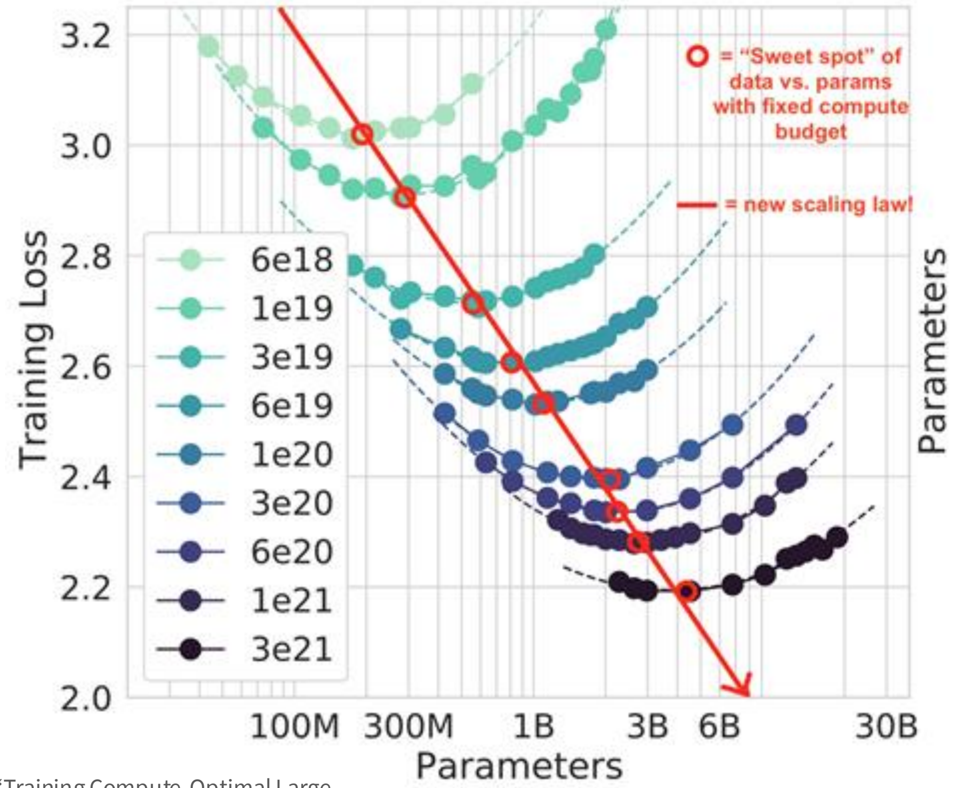
GPT-3 has 175B parameters!

GPT4: in the range of
1.8T parameters, trained on >1T tokens.

3 months at 20.000 A100s…

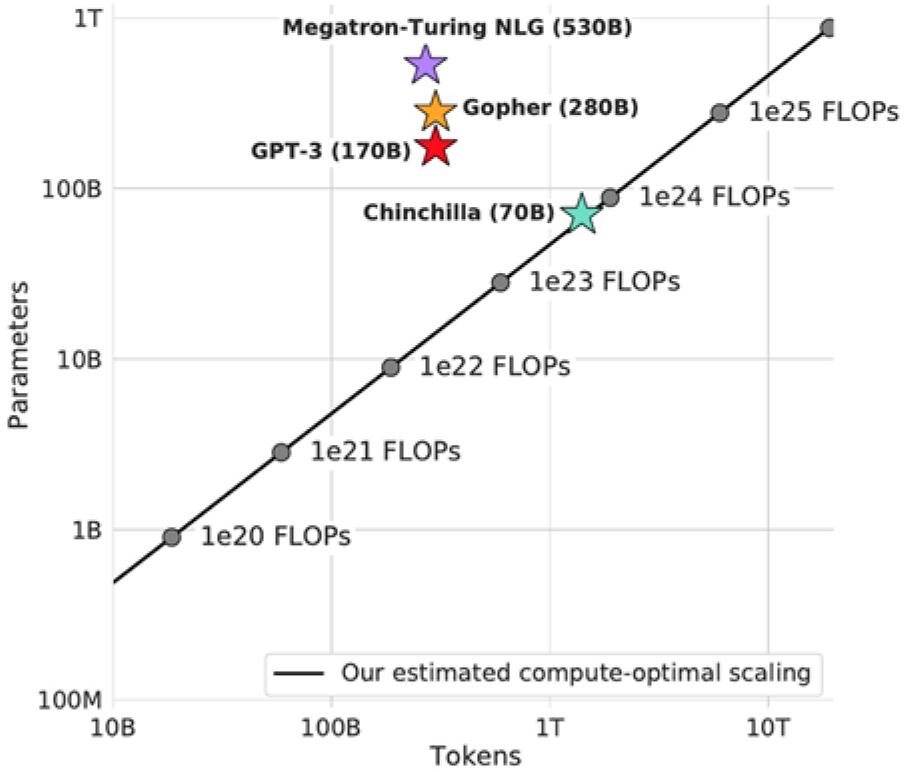


Chinchilla Scaling Laws



Hoffmann, Jordan et al. "Training Compute-Optimal Large Language Models." *ArXiv* abs/2203.15556 (2022): n. pag.

Chinchilla Scaling Laws



Hoffmann, Jordan et al. "Training Compute-Optimal Large Language Models." *ArXiv* abs/2203.15556 (2022): n. pag.

Finetuning LLMs at scale

High Performance Machine Learning Group

SURF

From training to finetuning

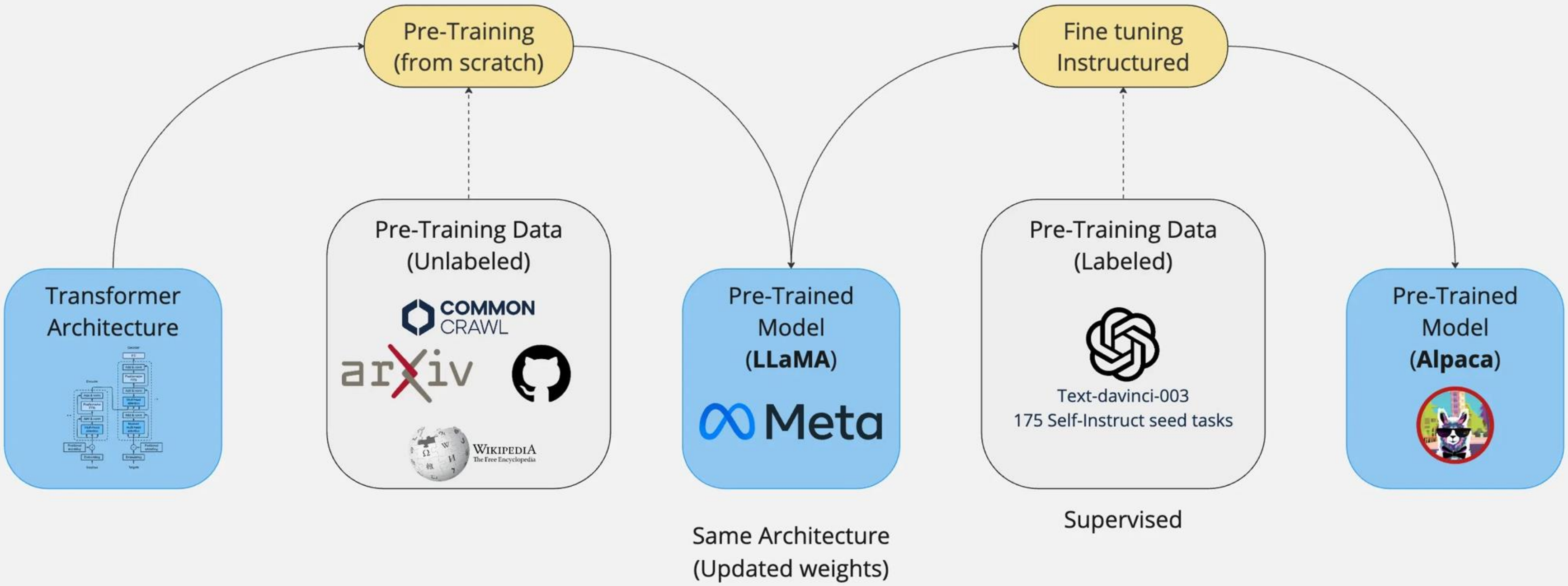


Image source

From training to finetuning

Current era of LLMS: few parties with enough data and compute to train large models

What can you do without requiring 1000 H100 GPUs? ☺

From training to finetuning

Current era of LLMS: few parties with enough data and compute to train large models

What can you do without requiring 1000 H100 GPUs? ☺

No training

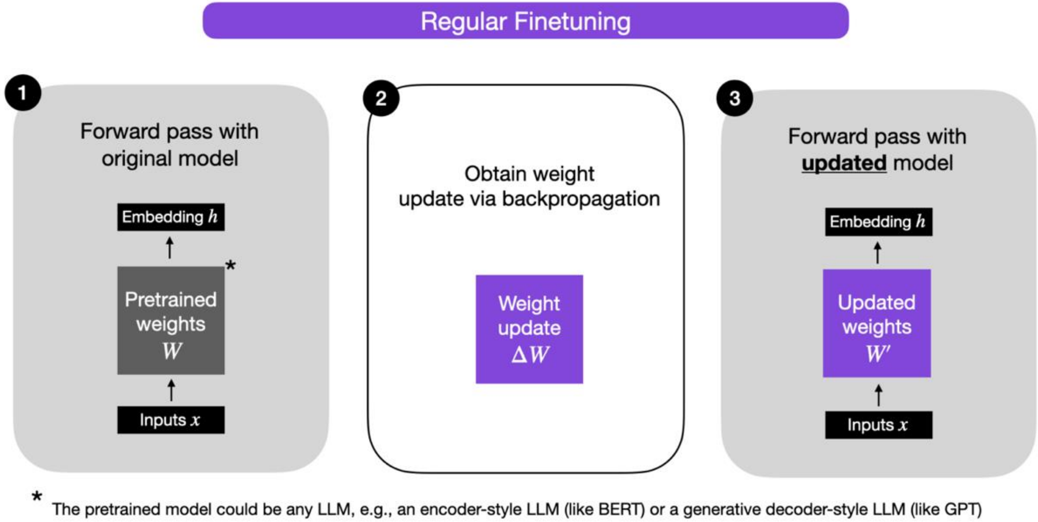
Use foundation models as they are:
- In-context learning
- Retrieval augmented generation

Finetuning

Transfer learning: finetune model to adapt to specific task

Efficient parameter optimization

Often not necessary to change **all** parameters



[Image source](#)

Efficient parameter optimization

Often not necessary to change **all** parameters

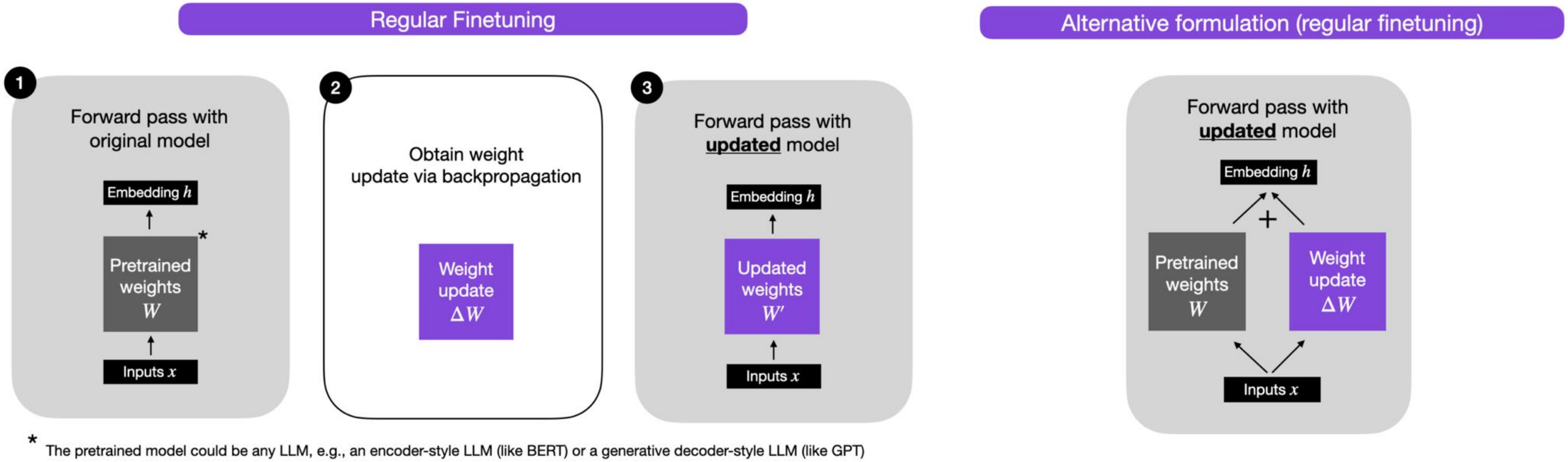
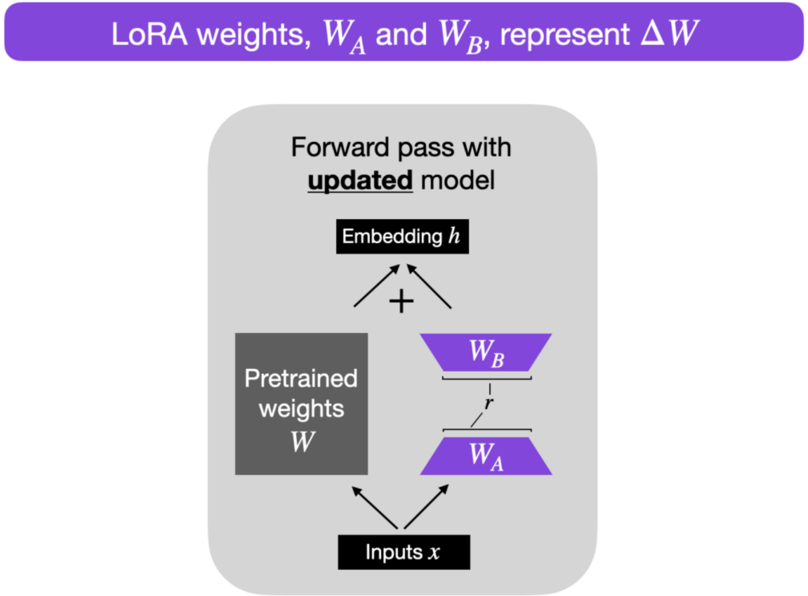


Image source

Efficient parameter optimization

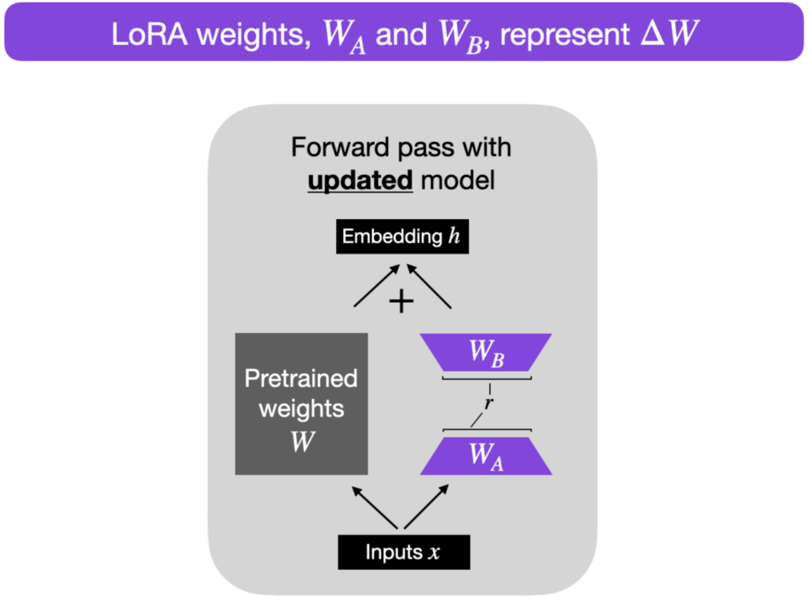
Often not necessary to change **all** parameters



[Image source]()

Efficient parameter optimization

Often not necessary to change **all** parameters



Reduces parameters in vram by about 4x!

Example:
LLama 7B in FP:
4 (bytes) x 7 = 28GB (inference)

8 (bytes) x 7 = 56GB (training)

This can go down to 14GB with PEFT!

Image source

Considerations

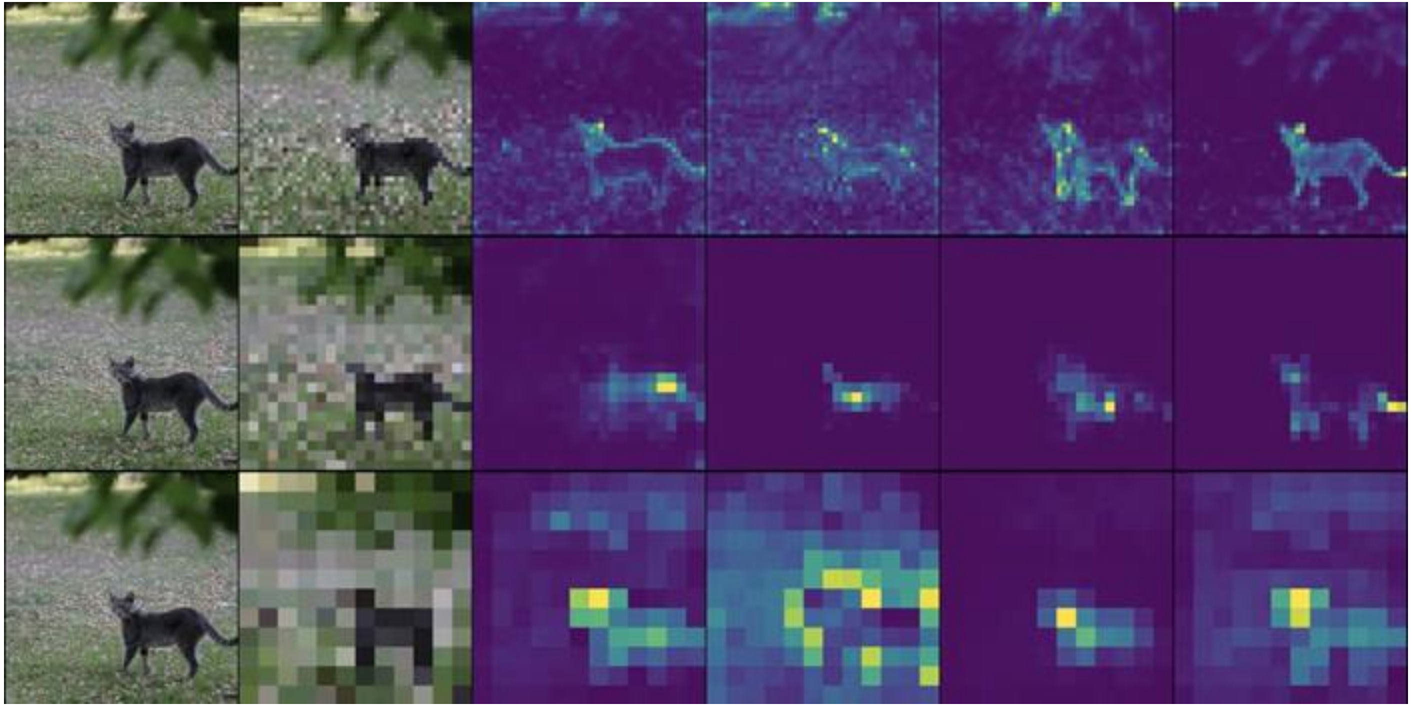
💬 Social bias
⚖️ Legal limitations
❗ Trust and reliability
Confident talking parrot
📢 Disinformation

🌳 Environmental impact

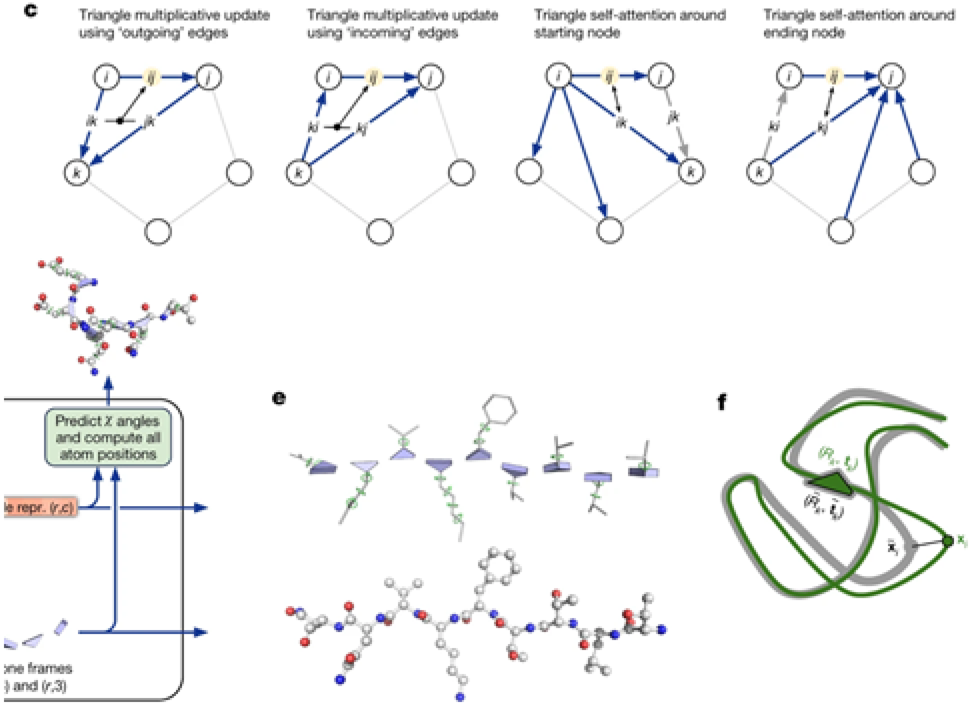
Transformers for other modalities

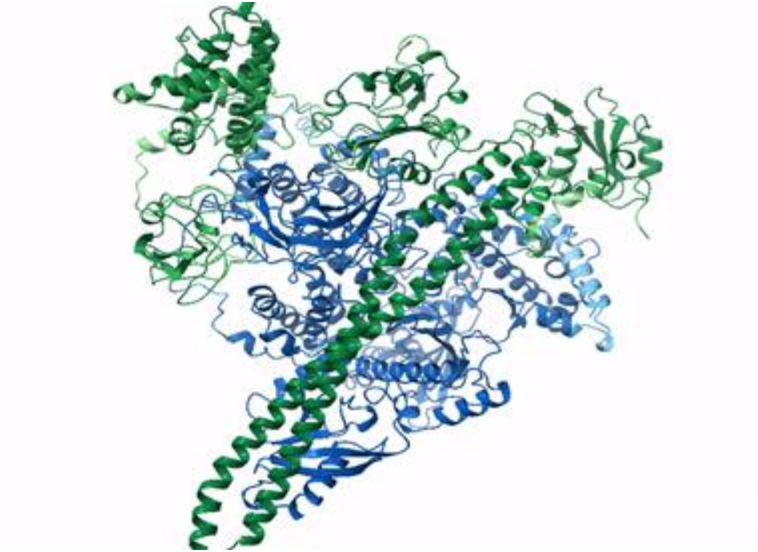
High Performance Machine Learning Group

Transformers for other modalities: ViT



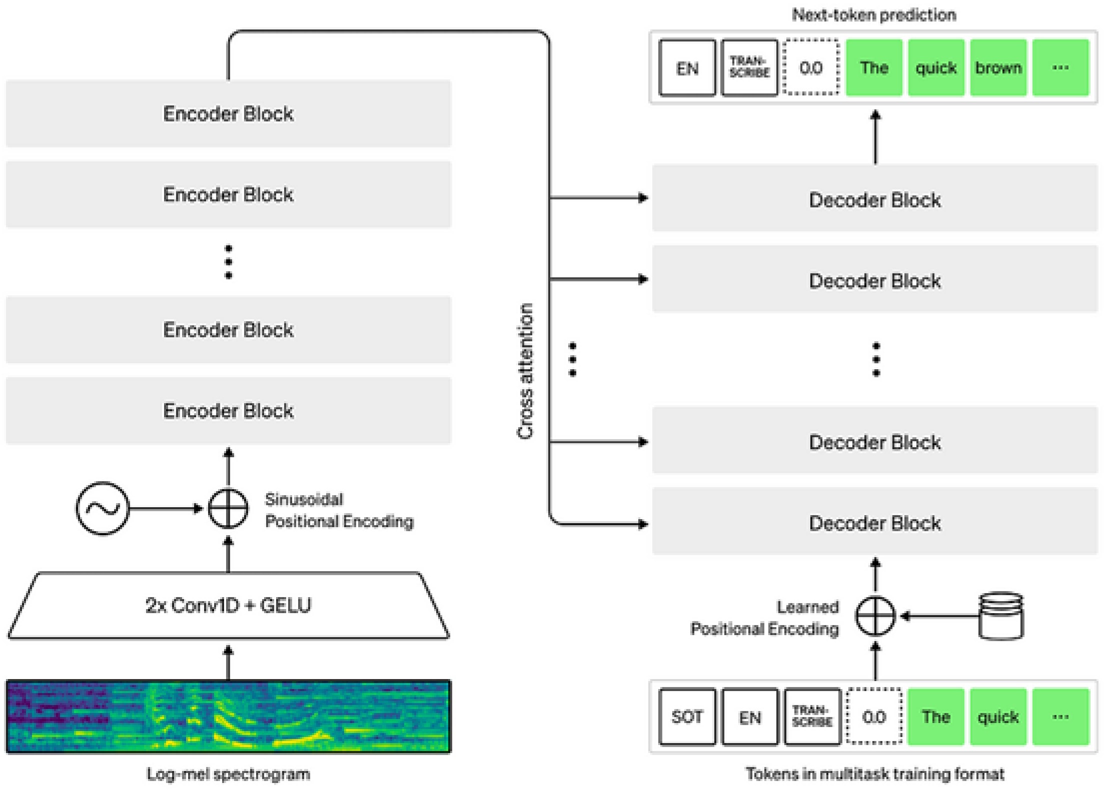
Transformers for other modalities: EvoFormer





https://www.nature.com/articles/s41586-021-03819-2

Transformers for other modalities: Whisper



https://openai.com/research/whisper

Want to learn more?

Use existing (efficient!) frameworks

vLLM

Hugging Face

unsloth

Make use of SURF guides for efficient finetuning:
https://servicedesk.surf.nl/wiki/display/WIKI/LLM+finetuning+on+Snellius

SURF

Fin

Introduction Series