

Intro to Neural Networks

BA865 – Mohannad Elhamod

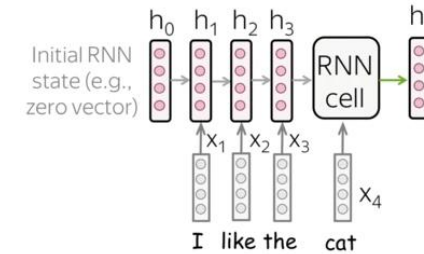
Transformers

Last time on BA865...

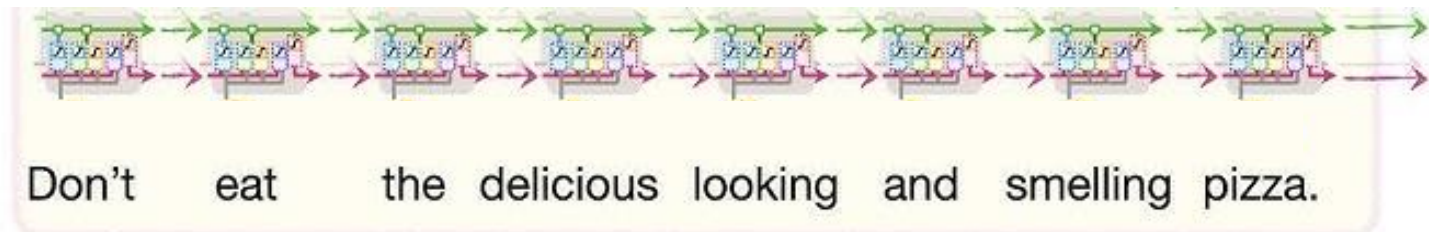
Catastrophic forgetting... !

- We started with having one memorization path...
- There were some attempts to solve the issue by adding more paths (e.g., LSTM added long and short term paths).
- Still a struggle to learn long sequences...

[Lena-voita](#)



Text: I like the **cat** on a mat <eos>
 ↑
 we are here
 not read yet



[StatQuest](#)

LSTM

An Embedding Per Token?

Instead of having embedding(s) that represent entire sequences, how about we...

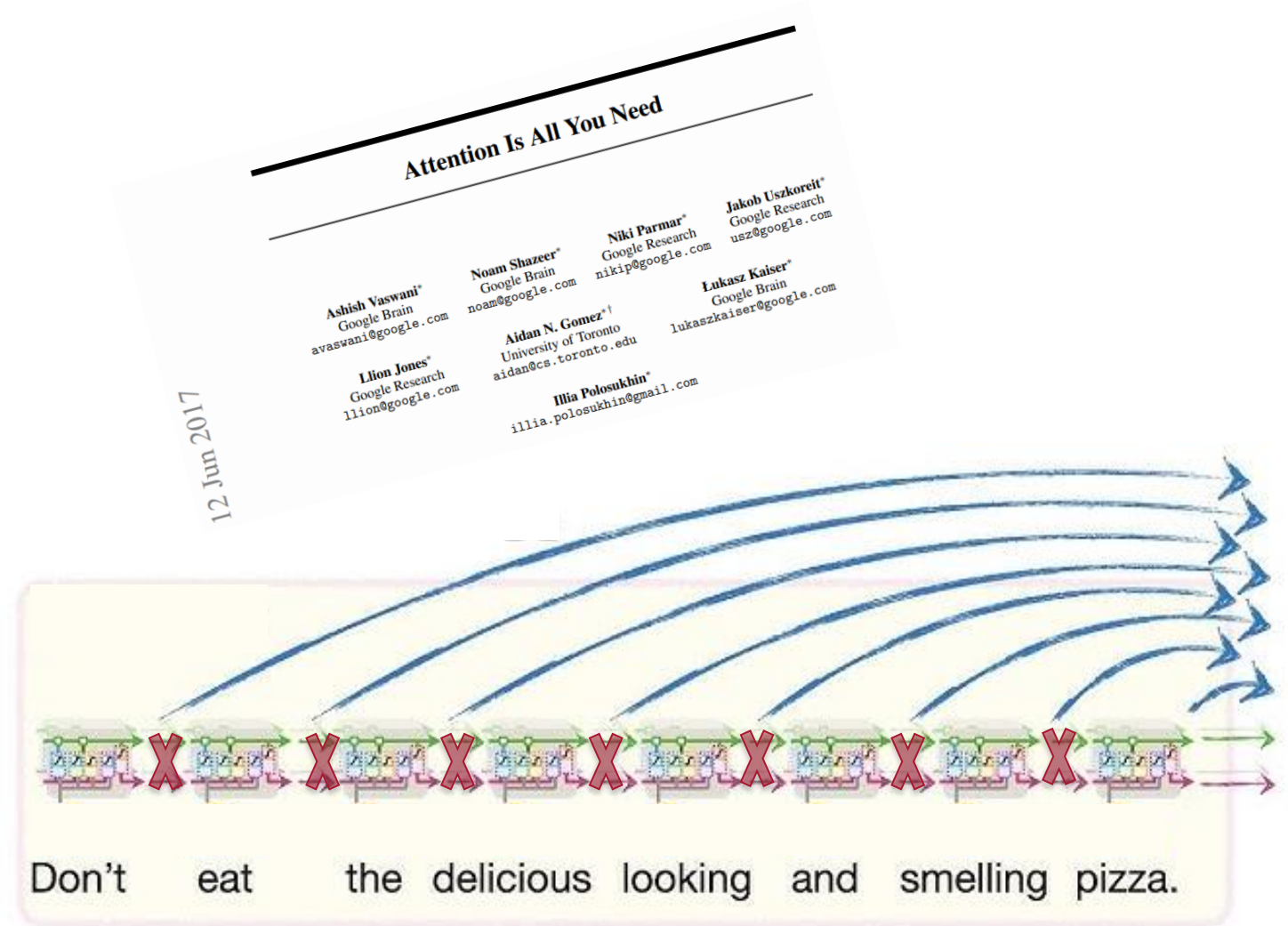
- Learn different embeddings for different tokens.
- Learn the relationship(s) (or similarity) between these tokens to represent the sequence (e.g., the sentence).



[StatQuest](#)

Attention!

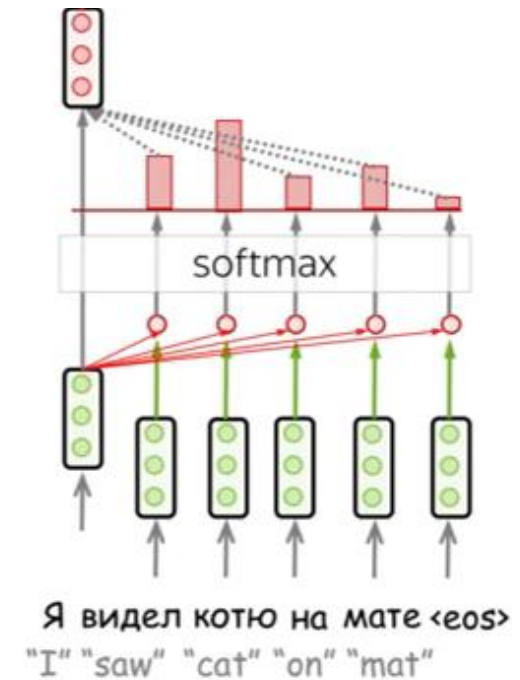
- So, let's not bother with memories that represent entire sequences anymore...
- We are only interested in the attention between tokens.



[StatQuest](#)

Attention!

- How do we capture similarity?
 - Dot product (i.e., cosine similarity)
 - Based on the similarities between the tokens, we can create new embeddings!
 - How is this implemented?



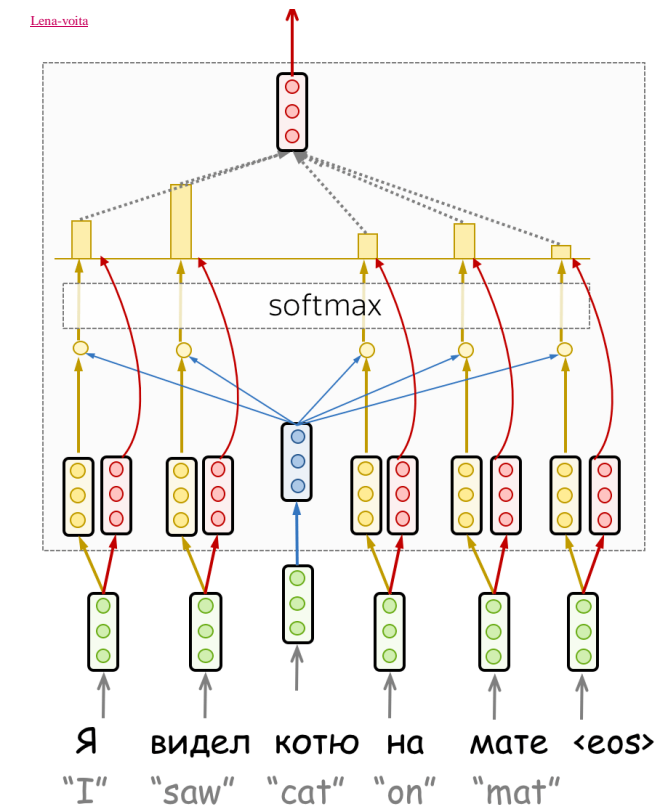
[Lena-voita](#)

Attention!

- The objective is to learn more sophisticated embeddings that capture the semantics of the sentence.
- **Query: Source of attention (e.g., I am looking for an adjective)...**
- **Key: Target of attention (e.g., I am your adjective)...**
 - Their dot product gives the **attention scores**.
- **Value : used as an embedding weighted by these scores.**

$$\text{Attention}(q, k, v) = \overbrace{\text{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right)}^{\text{Attention weights}} v$$

from
to
vector dimensionality of K, V



Attention!

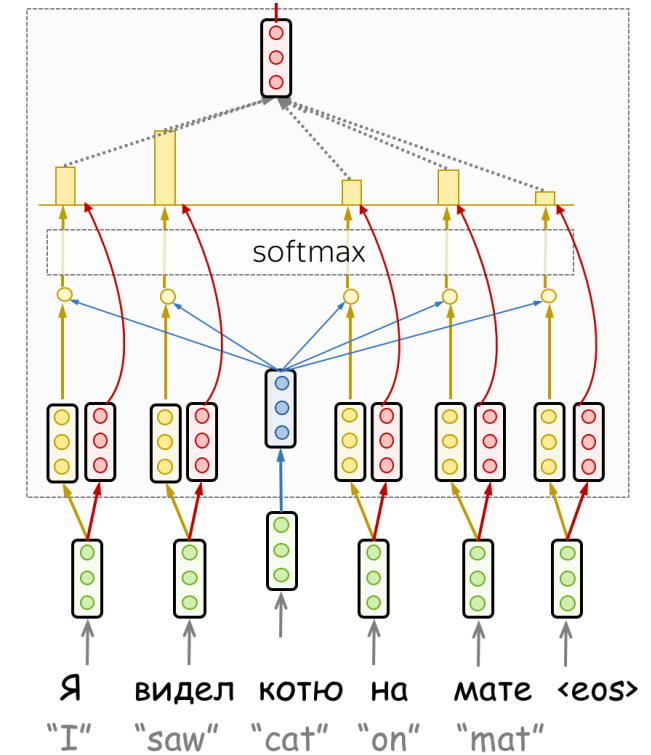
- How are these key, value, and query calculated?
- This is called an *“attention head”*.

$$\begin{bmatrix} W_Q \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{blue} \\ \text{blue} \\ \text{blue} \end{bmatrix}$$

$$\begin{bmatrix} W_K \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{yellow} \\ \text{yellow} \\ \text{yellow} \end{bmatrix}$$

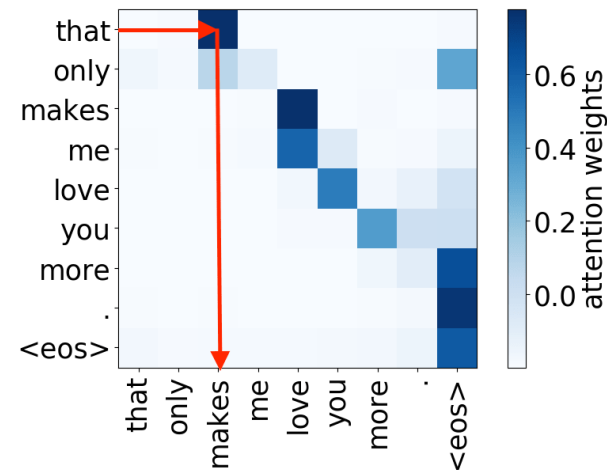
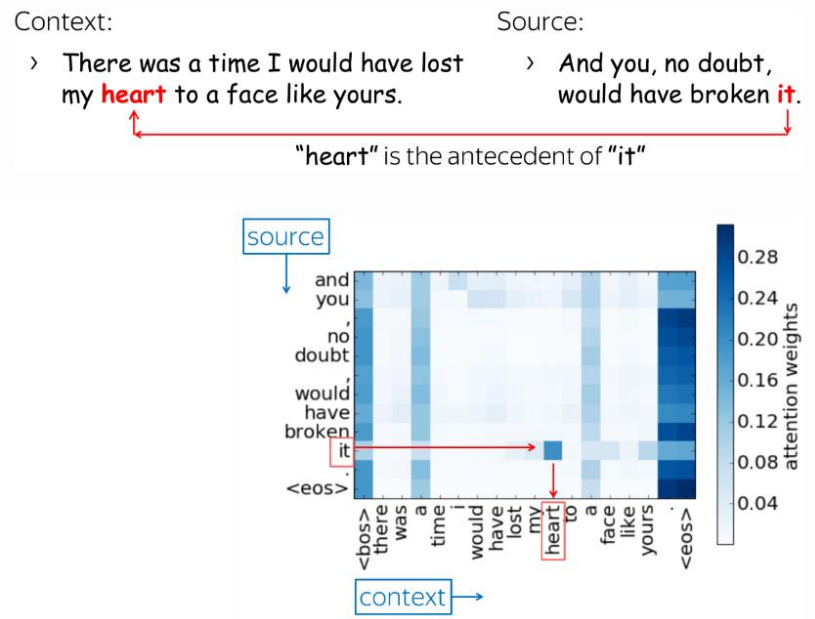
$$\begin{bmatrix} W_V \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{red} \\ \text{red} \\ \text{red} \end{bmatrix}$$

Lena-voita

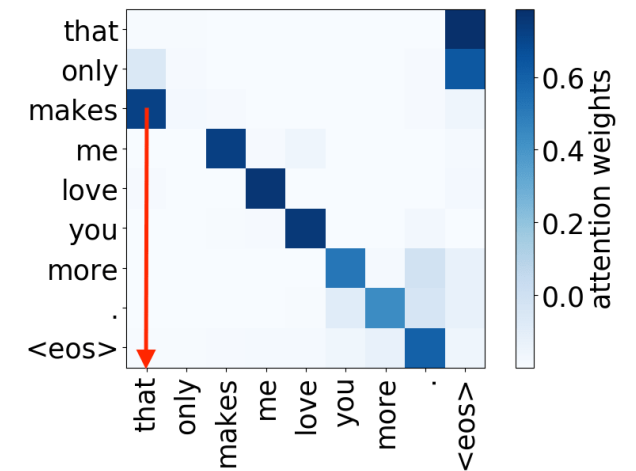


Attention!

- Examples



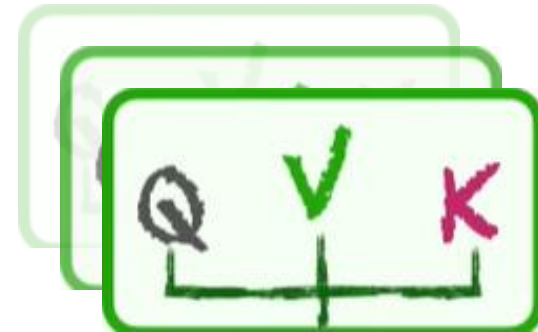
Subject -> verb



Verb -> subject

Attention!

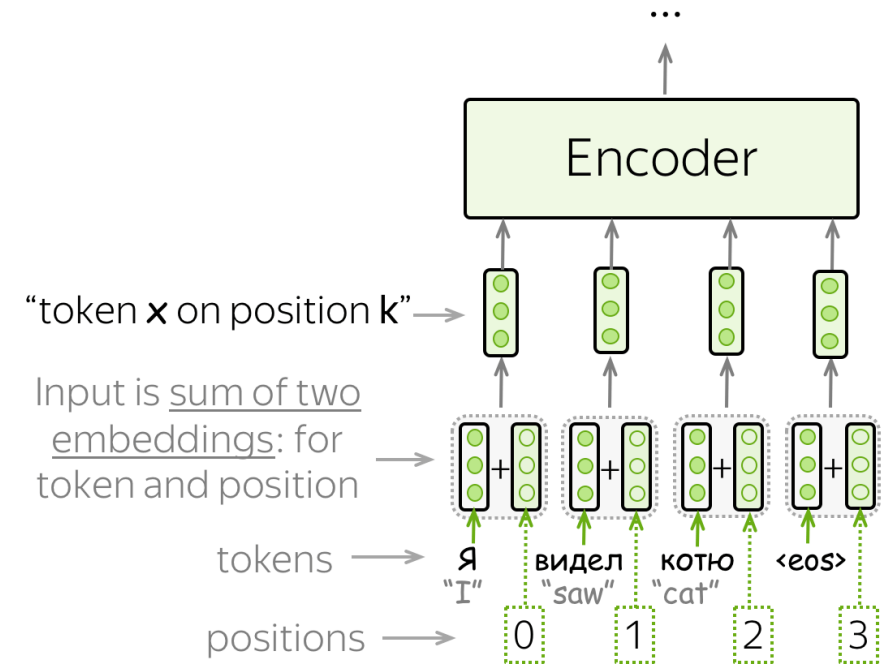
- If one head can learn some relationship in the sequence...
- multiple heads can learn multiple relationships.



Multi-head attention

Attention!

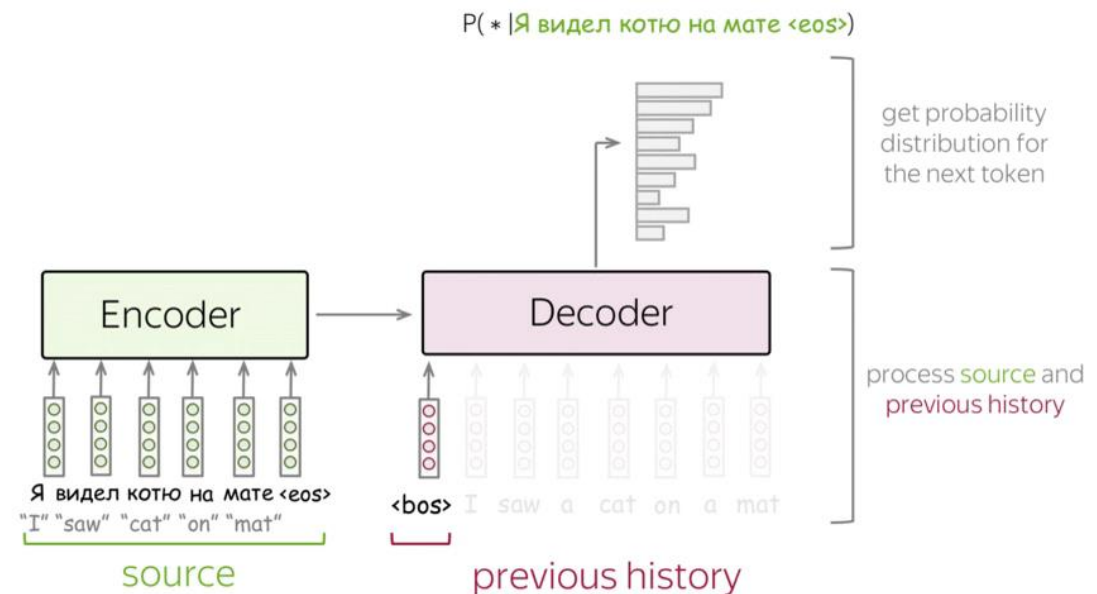
- Note that the word embeddings are now learned as part of model training.
- Since the order of tokens in the sequence matters, we add a “*positional encoding*” to the word embedding.



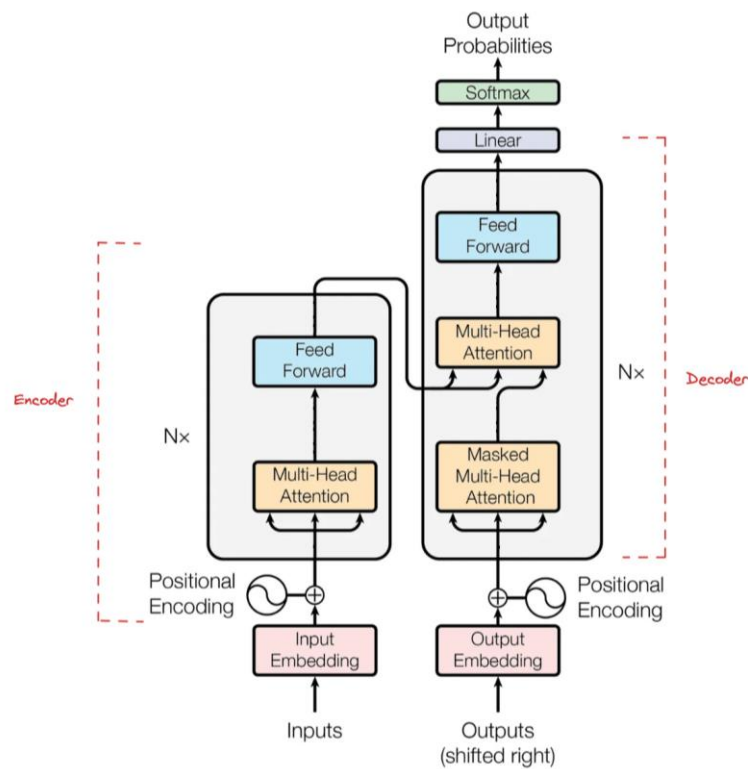
[Lena-volta](#)

Attention in an Encoder-Decoder Framework

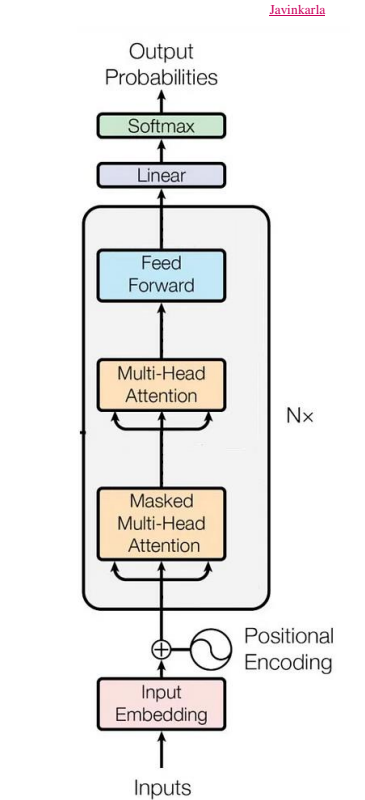
- One can think of the attention head(s) as the layer(s) in an encoder.
- Instead of having an embedding of the entire sequence (e.g., RNN), now we have an embedding per token.
- A decoder can be learned to do other tasks (e.g., translation, text-to-audio, etc.).



The Transformer is Born!



Original Encoder-Decoder version



GPT (Decoder only) version

The Transformer is Born!

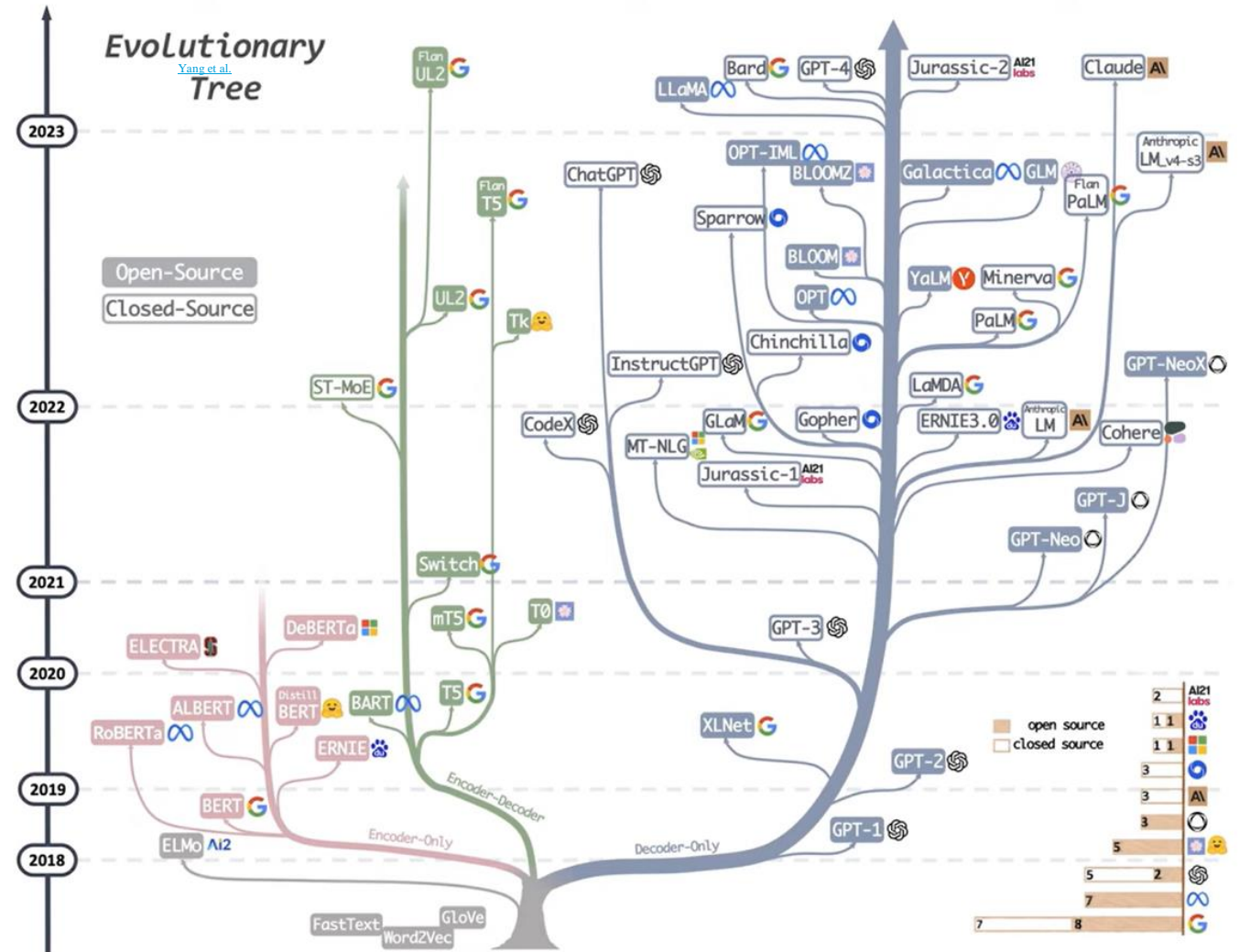
Model	Examples	Tasks
Encoder	ALBERT, BERT, DistilBERT, ELECTRA, RoBERTa	Sentence classification, named entity recognition, extractive question answering
Decoder	CTRL, GPT, GPT-2, Transformer XL	Text generation
Encoder-decoder	BART, T5, Marian, mBART	Summarization, translation, generative question answering

[Javinkara](#)

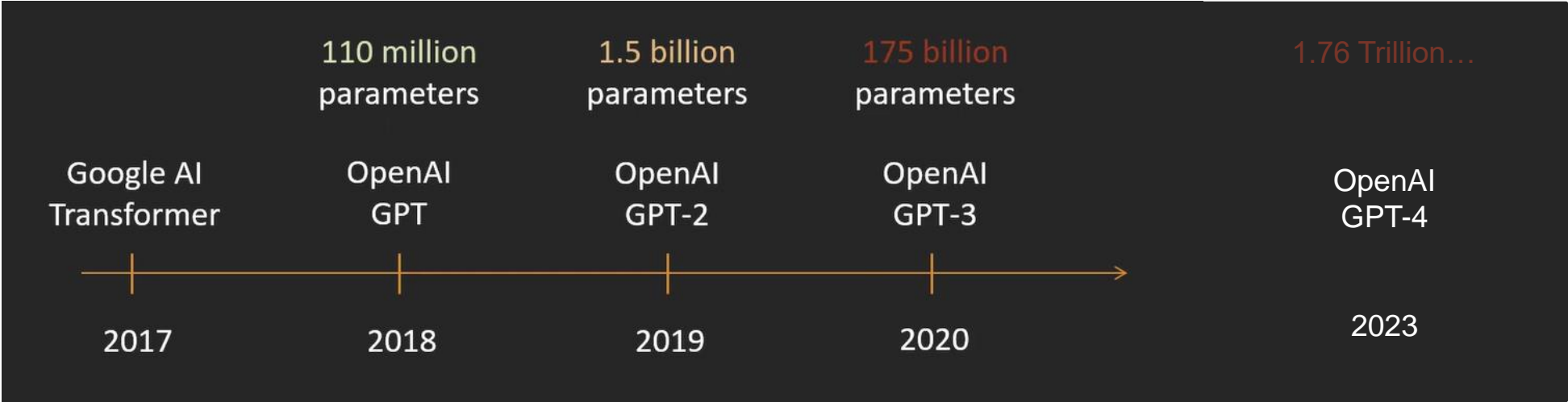
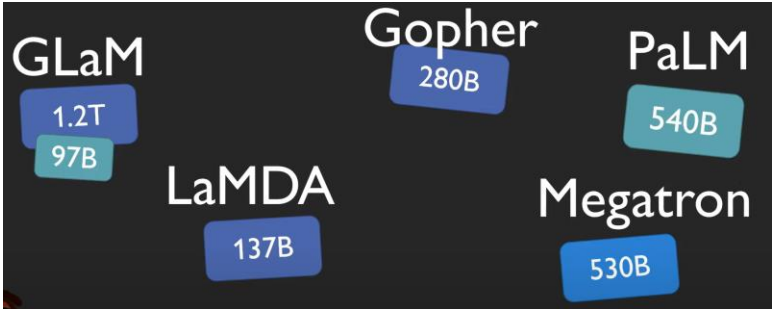
Why so many?!

Where do the differences come from?

- **Data.**
- **Model** type and size.
- **Hyperparameters** (context size, embedding size,...).
- **Training process** (the cost function, fine-tuning, human feedback, etc.).



The GPT evolution...



[AI Coffee Break with Letitia](#)

[Book Corpus](#)
[WebText](#)


1,038 books (around 74M sentences and 1G words) of 16 different sub-genres (e.g., Romance, Historical, Adventure, etc.)

[Common Crawl + ...](#)

Over 240 billion pages.
Petabytes of data.

[????](#)

The GPT evolution...

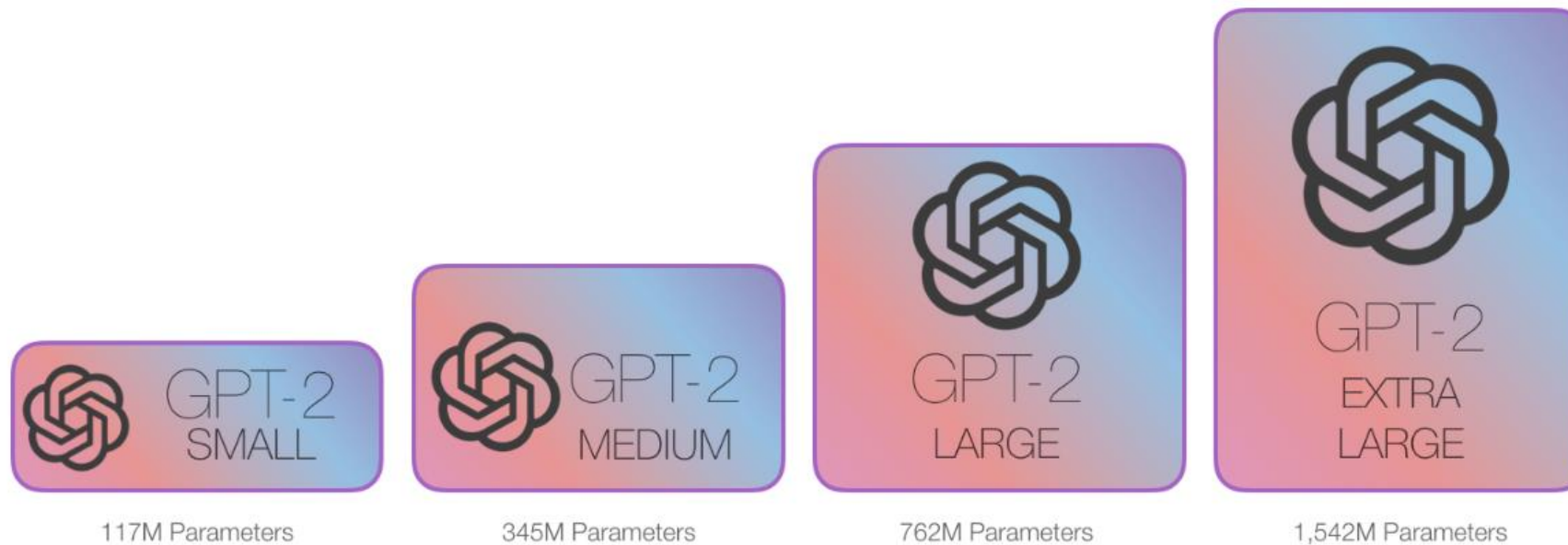


Link in the description below. 🖱️ Chowdhery et al. 2022

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

[AI Coffee Break with Letitia](#)

Different model sizes



[Jay Alamar](#)