

# TRANSFORMER ARCHITECTURE

Vincent Guigue,  
inspiré des supports de Benjamin Piwowarski & Thomas Gérald



# INTRODUCTION



# Aggregation problem

Sequence :  $\mathbf{s} = \{s_1, \dots, s_L\}$

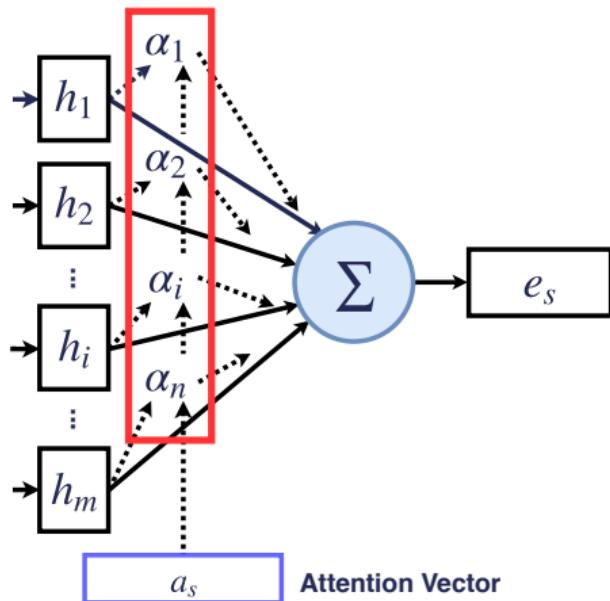
Representation learning :  $s_i \rightarrow \mathbf{e}_i \in \mathbb{R}^Z$

Decision on  $\mathbf{s} \Rightarrow$  Representation / mapping  $R(\mathbf{s}) : \mathbb{R}^{L \times Z} \mapsto \mathbb{R}^d$

+ Decision function  $f(R(\mathbf{s})) : \mathbb{R}^d \mapsto \mathcal{Y}$

- RNN + last hidden layer (ext: Bi-RNN)
- CNN/RNN + pooling
- CNN/RNN + attention = learnt weighted sum

# Attention & aggregation vs Diffusion process

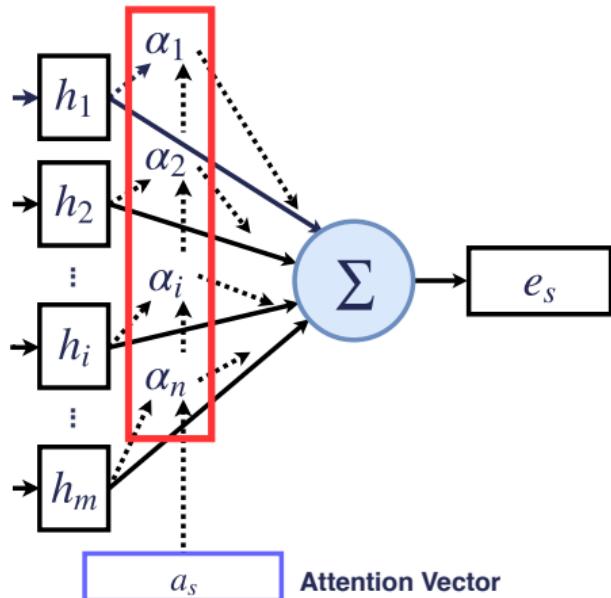


- Learning to weight localized representation
- Many efficient applications [2014-2018]
- Few parameters
- Attention vector = Query vector
- How to code it in pytorch?

Practical Session:

- Q1 Preliminary tests: uniform attention
- Q2 Classical global attention

# Attention & aggregation vs Diffusion process



- Learning to weight localized representation
- Many efficient applications [2014-2018]
- Few parameters
- Attention vector = Query vector
- How to code it in pytorch?

Q3 ⇒ Reconstruction of **all embedding** by recombination = diffusion process



# Inspiration from memory networks (question answering)

## Task:

Sam walks into the kitchen.  
Sam picks up an apple.  
Sam walks into the bedroom.  
Sam drops the apple.

Q: Where is the apple?  
A. Bedroom

Brian is a lion.  
Julius is a lion.  
Julius is white.  
Bernhard is green.

Q: What color is Brian?  
A. White

Mary journeyed to the den.  
Mary went back to the kitchen.  
John journeyed to the bedroom.  
Mary discarded the milk.

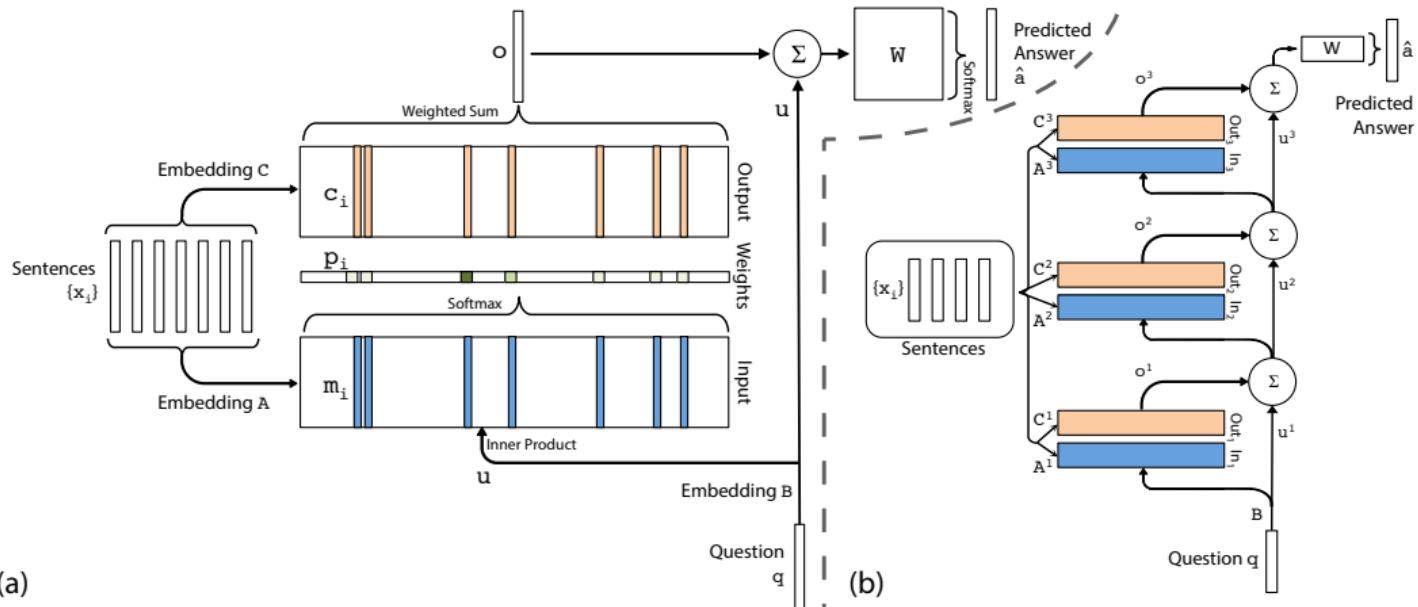
Q: Where was the milk before the den?  
A. Hallway

Babi dataset

## How to anchor question answering in Knowledge Bases?

# Inspiration from memory networks (question answering)

How to anchor question answering in Knowledge Bases?



# Inspiration from memory networks (question answering)

## How to anchor question answering in Knowledge Bases?

<b>Story (1: 1 supporting fact)</b>	<b>Support</b>	<b>Hop 1</b>	<b>Hop 2</b>	<b>Hop 3</b>
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
<b>Where is John? Answer: bathroom Prediction: bathroom</b>				

<b>Story (2: 2 supporting facts)</b>	<b>Support</b>	<b>Hop 1</b>	<b>Hop 2</b>	<b>Hop 3</b>
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
<b>Where is the milk? Answer: hallway Prediction: hallway</b>				

<b>Story (16: basic induction)</b>	<b>Support</b>	<b>Hop 1</b>	<b>Hop 2</b>	<b>Hop 3</b>
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
<b>What color is Greg? Answer: yellow Prediction: yellow</b>				

<b>Story (18: size reasoning)</b>	<b>Support</b>	<b>Hop 1</b>	<b>Hop 2</b>	<b>Hop 3</b>
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
<b>Does the suitcase fit in the chocolate? Answer: no Prediction: no</b>				



# Conclusion

Current limitations with RNN:

- Long dependencies
- Global message representation
  - Continuous indexing
- Targeted dependencies
  - Co-reference resolution

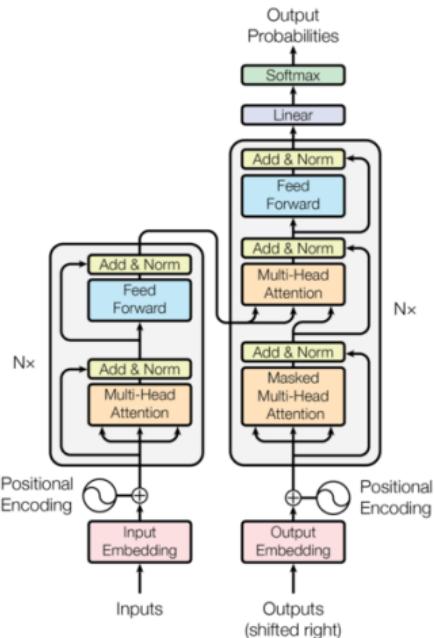
# ARCHITECTURE



# Transformer generality

Several sources to understand Transformers [easier than the original article]

- J. Alammar 2018 <http://jalammar.github.io/illustrated-transformer/>
- Alexander Rush 2018 <http://nlp.seas.harvard.edu/2018/04/03/attention.html>
- P. Bloem 2019 <http://www.peterbloem.nl/blog/transformers>

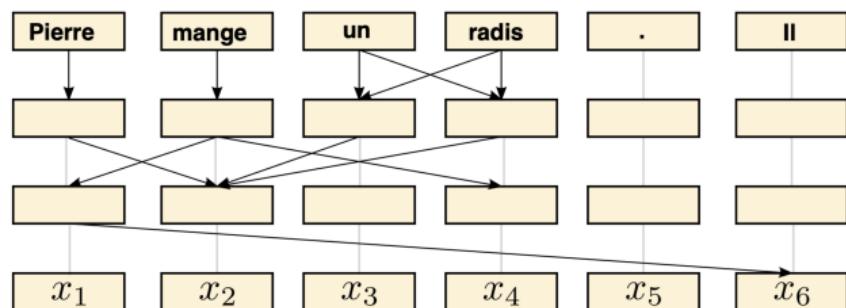


Overcoming classical pictures  
⇒ re-implementing a transformer block

# Subword representation

General idea: representation diffusion + learning  
the diffusion

Scale choice



Character  
+ (very)  
small  
vocabulary  
- no  
semantics

Word  
+ Semantics  
++  
- Huge  
vocabulary  
- Unknown  
words

[book] Sennrich, R. et al. 2016. Neural Machine Translation of Rare Words with Subword Units.

les chats sont des félidés

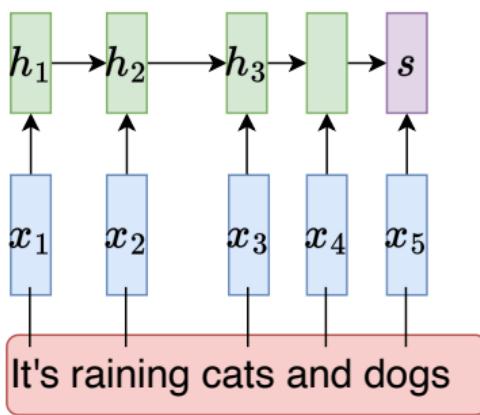
est transformé en

[CLS] \_les \_chats \_sont \_des \_f éli dés [SEP]

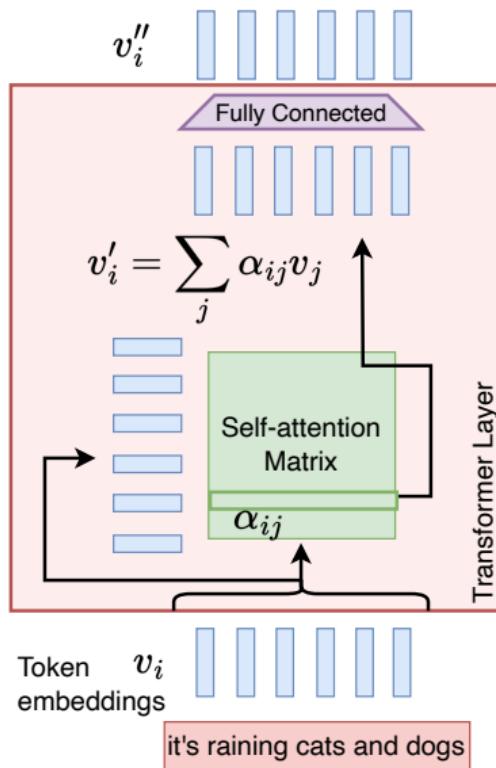
# Transformer architecture: state-of-the-art aggregation

## Recurrent Neural Network:

$$h_{t+1} = h_t W_1 + x_{t+1} W_2$$



## Transformer:



Attention is all you need, [Vaswani et al. NeurIPS 2017](#)

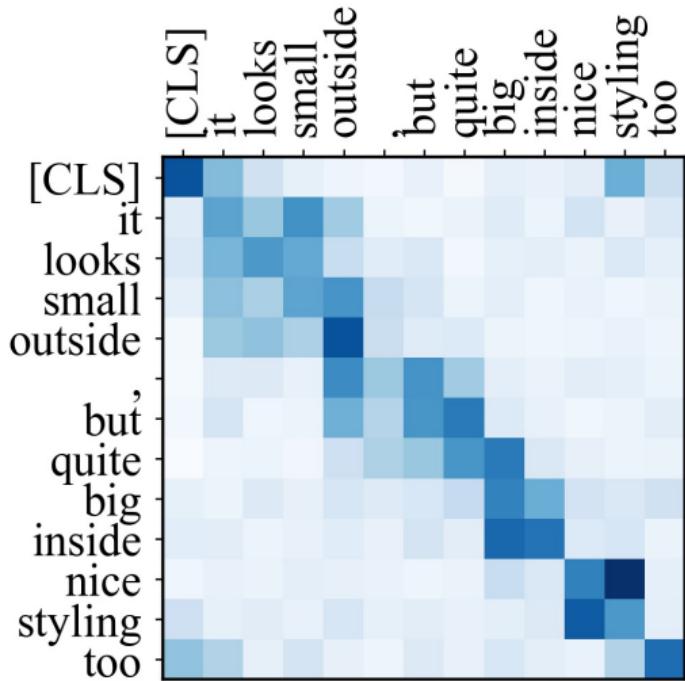
Sequence to Sequence Learning with Neural Networks, [Sutskever et al. NeurIPS 2014](#)

# Attention block: General idea

- 1 Computing self-attention between tokens
- 2 Normalizing attention (softmax)
- 3 Exploiting attention to recombine token representation

⇒ How to compute interactions  
(=self attention)

⇒ New token representation  
(=contextual representation)



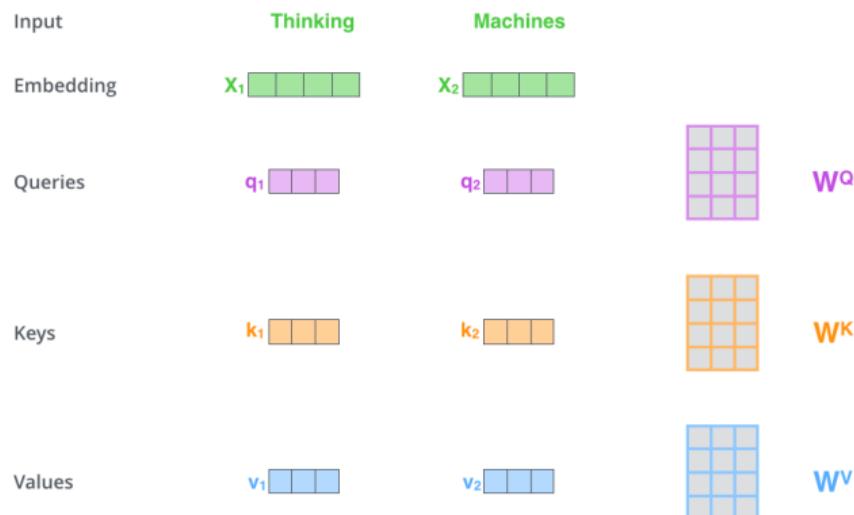


# Attention block

Interactions through vectors derived from the embedding

- 1 Compute Keys, Queries, Values by linear mapping

**Note:** here are the main parameters of the transformer architecture

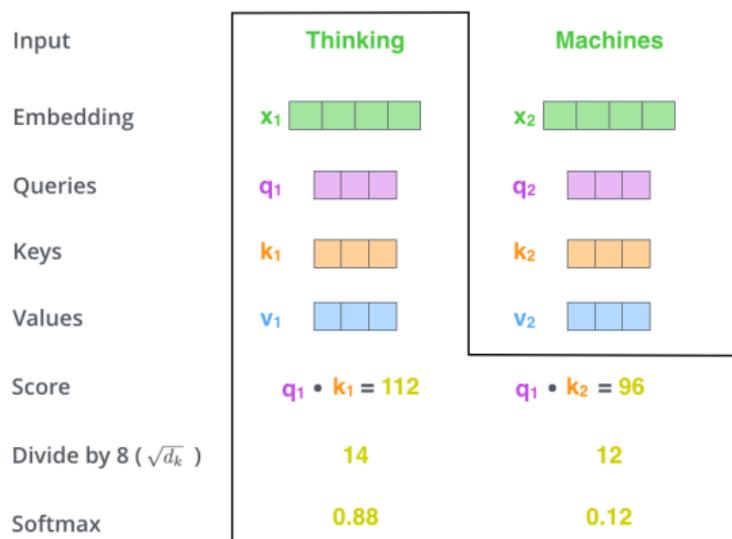




# Attention block

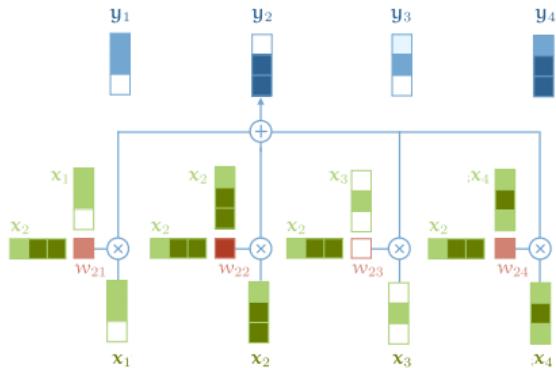
## Computing the attention

- 1 Inner product between keys/queries
- 2 Division by key/query size
  - Softmax too sensitive to large values
  - Temperature
- 3 Softmax  $\Rightarrow$  over the relevant dimension



# Attention block

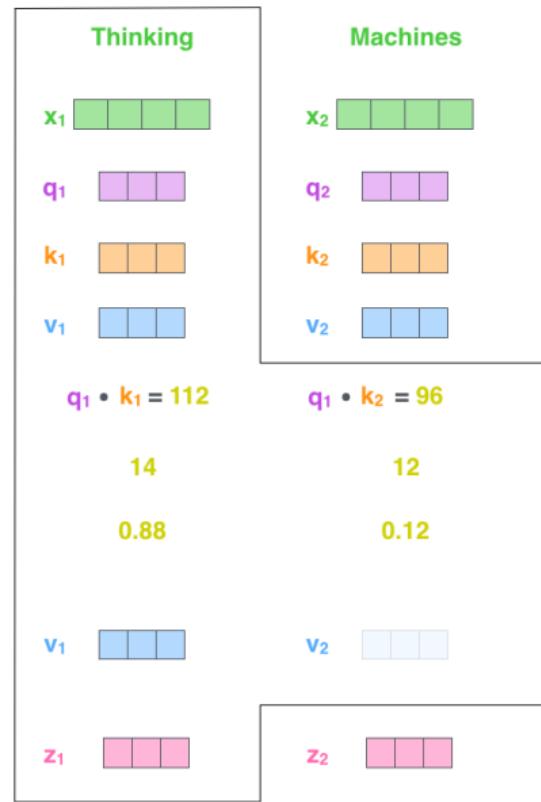
New values:



$$v_i^{\ell+1} = \sum \alpha_k v_k^\ell$$

How to come back to the embedding space?  
 $\text{value} \mapsto \text{embedding}$

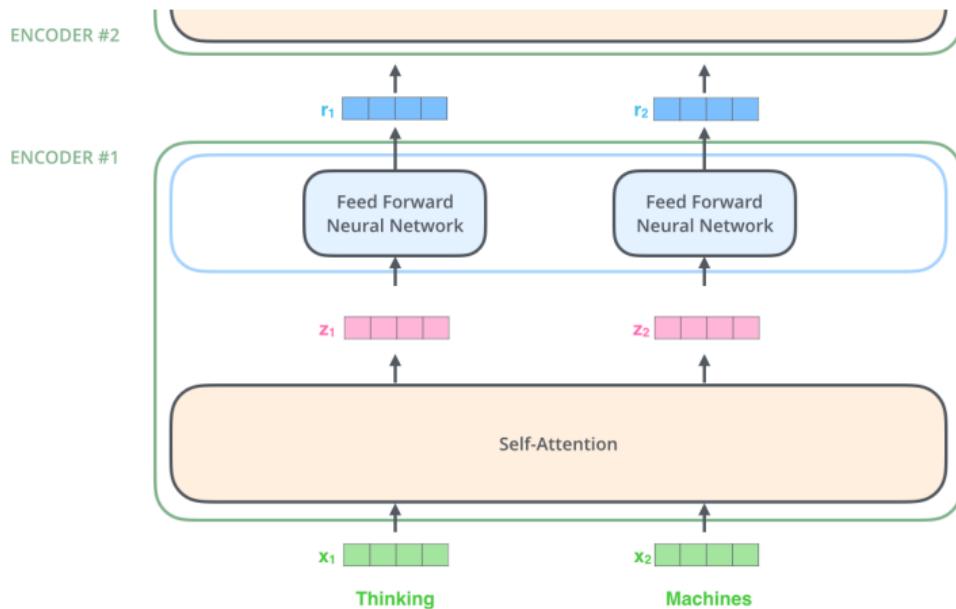
Input  
Embedding  
Queries  
Keys  
Values  
Score  
Divide by 8 ( $\sqrt{d_k}$ )  
Softmax  
Softmax X Value  
Sum





# Attention block

Use a fully connected to map to embedding space



# Parallelism $\Rightarrow$ Batch of data

All tokens at once !

$$\begin{matrix} \mathbf{X} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{W}^Q \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} = \begin{matrix} \mathbf{Q} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

$$\begin{matrix} \mathbf{X} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{W}^K \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} = \begin{matrix} \mathbf{K} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

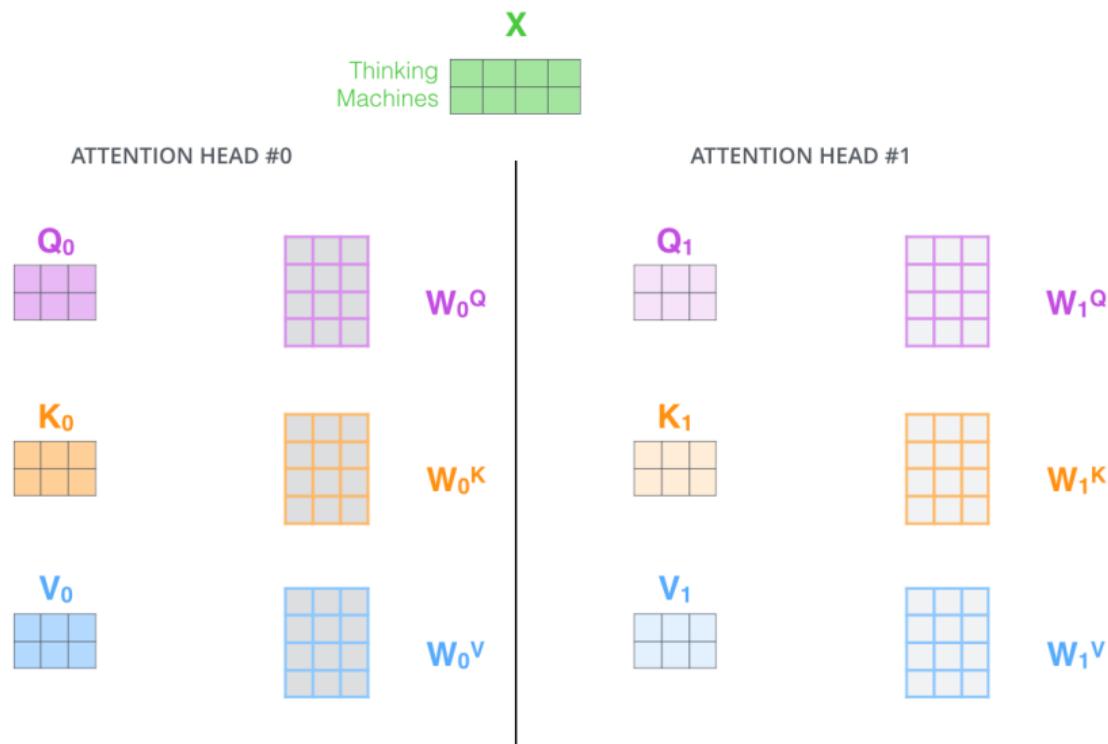
$$\begin{matrix} \mathbf{X} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{W}^V \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} = \begin{matrix} \mathbf{V} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

$$\text{softmax} \left( \frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}} \right) = \mathbf{z}$$
$$\begin{matrix} \mathbf{Q} \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{K}^T \\ \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \quad \mathbf{v}$$



# Multiple heads : dimension compatibility challenge

Why only **one** interaction?





# Multiple heads : dimension compatibility challenge

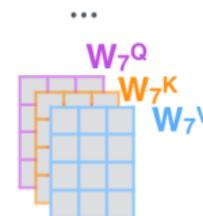
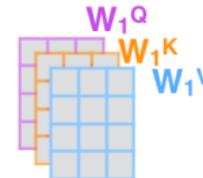
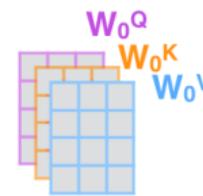
Why only **one** interaction?

1) This is our input sentence\*  
each word\*

Thinking  
Machines



2) We embed each word\*  
3) Split into 8 heads.  
We multiply **X** or **R** with weight matrices



4) Calculate attention using the resulting Q/K/V matrices



5) Concatenate the resulting Z matrices, then multiply with weight matrix **W<sup>O</sup>** to produce the output of the layer



\* In all encoders other than #0, we don't need embedding.  
We start directly with the output of the encoder right below this one





# Multiple heads : dimension compatibility challenge

Why only **one** interaction?

Typically, values are concatenated then mapped to rebuild embeddings

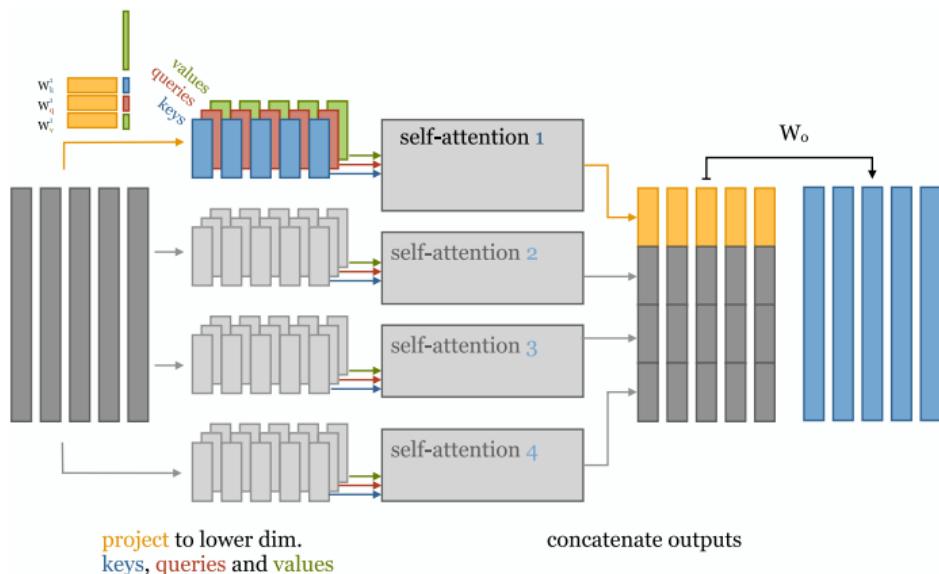
BERT

- Embedding : 768
- key/query/values : 64
- ... & 12 heads

$$\Rightarrow 12 \times 64 = 768$$

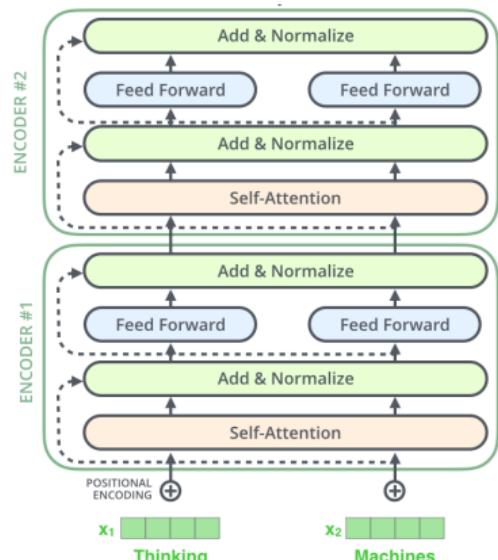
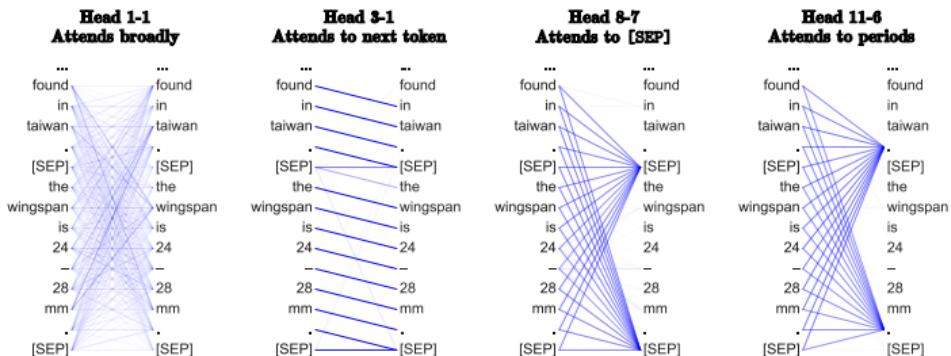
ChatGPT:

- Embedding : 12288
- key/query/values : 128
- 96 heads



# Multiple layers

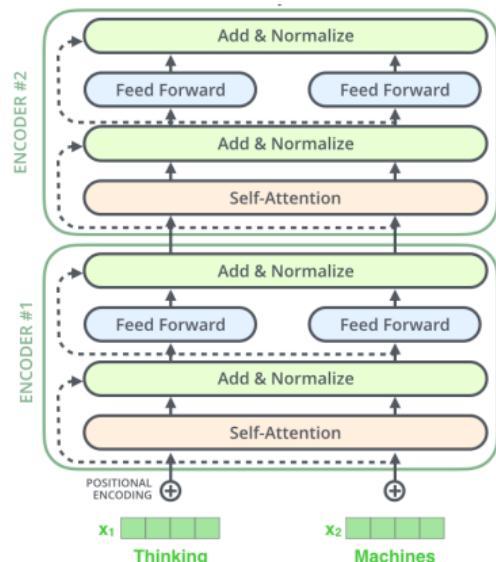
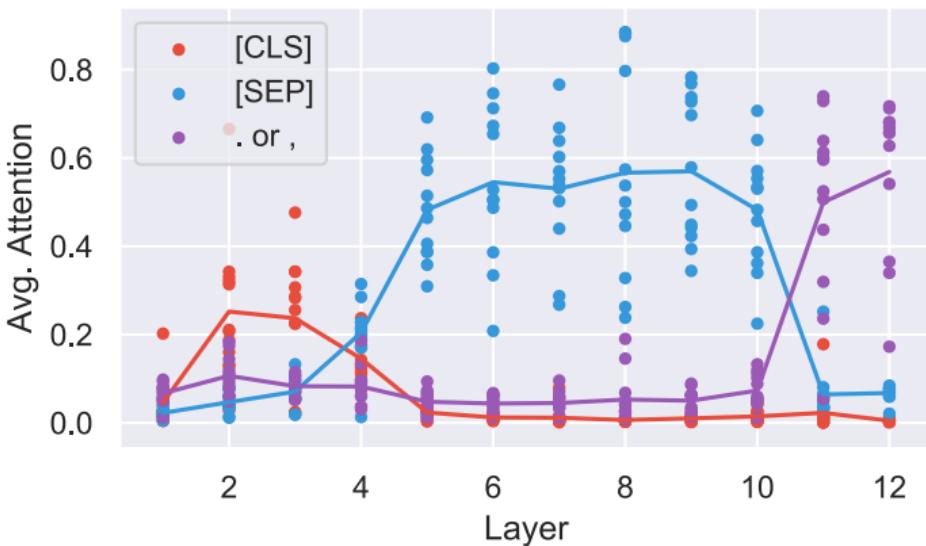
- Have a look on the skip connexion
- Diffusion is made step-by-step
- Bertology shows that different relations are extracted at different scales





# Multiple layers

- Have a look on the skip connexion
- Diffusion is made step-by-step
- Bertology shows that different relations are extracted at different scales

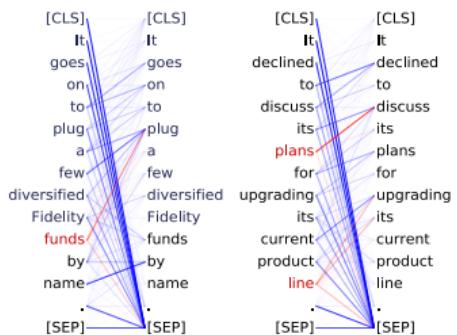


# Multiple layers

- Have a look on the skip connexion
- Diffusion is made step-by-step
- Bertology shows that different relations are extracted at different scales

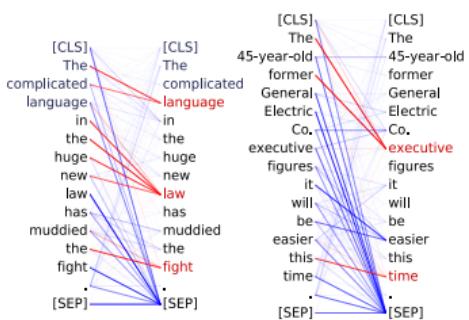
## Head 8-10

- Direct objects attend to their verbs
- 86.8% accuracy at the dobj relation

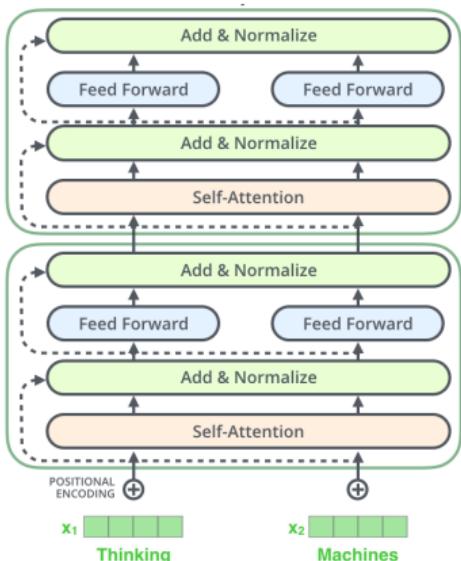


## Head 8-11

- Noun modifiers (e.g., determiners) attend to their noun
- 94.3% accuracy at the det relation



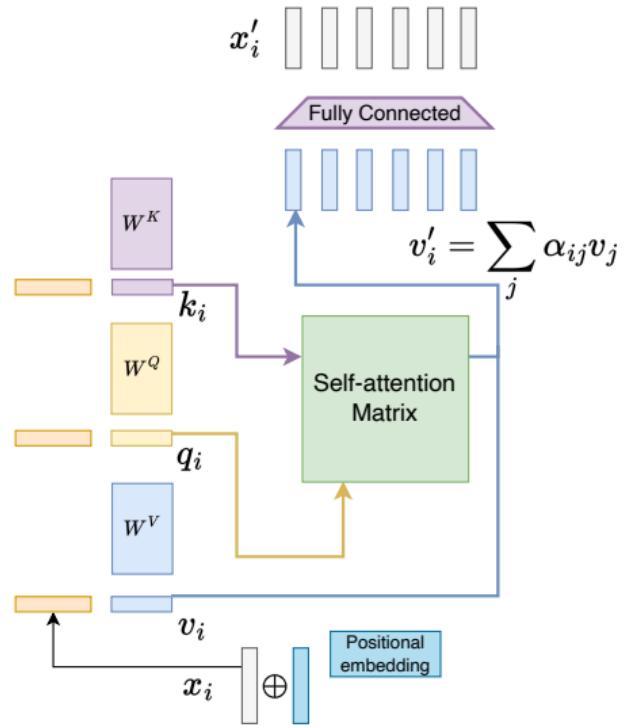
ENCODER #1  
ENCODER #2





# Position coding

Until now, Transformer does not take position into account !!



the cat sat on the mat



# Position coding

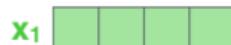
Until now, Transformer does not take position into account !!

POSITIONAL  
ENCODING

0	0	1	1
---	---	---	---

+

EMBEDDINGS



0.84	0.0001	0.54	1
------	--------	------	---

+

0.91	0.0002	-0.42	1
------	--------	-------	---

+

INPUT

Je

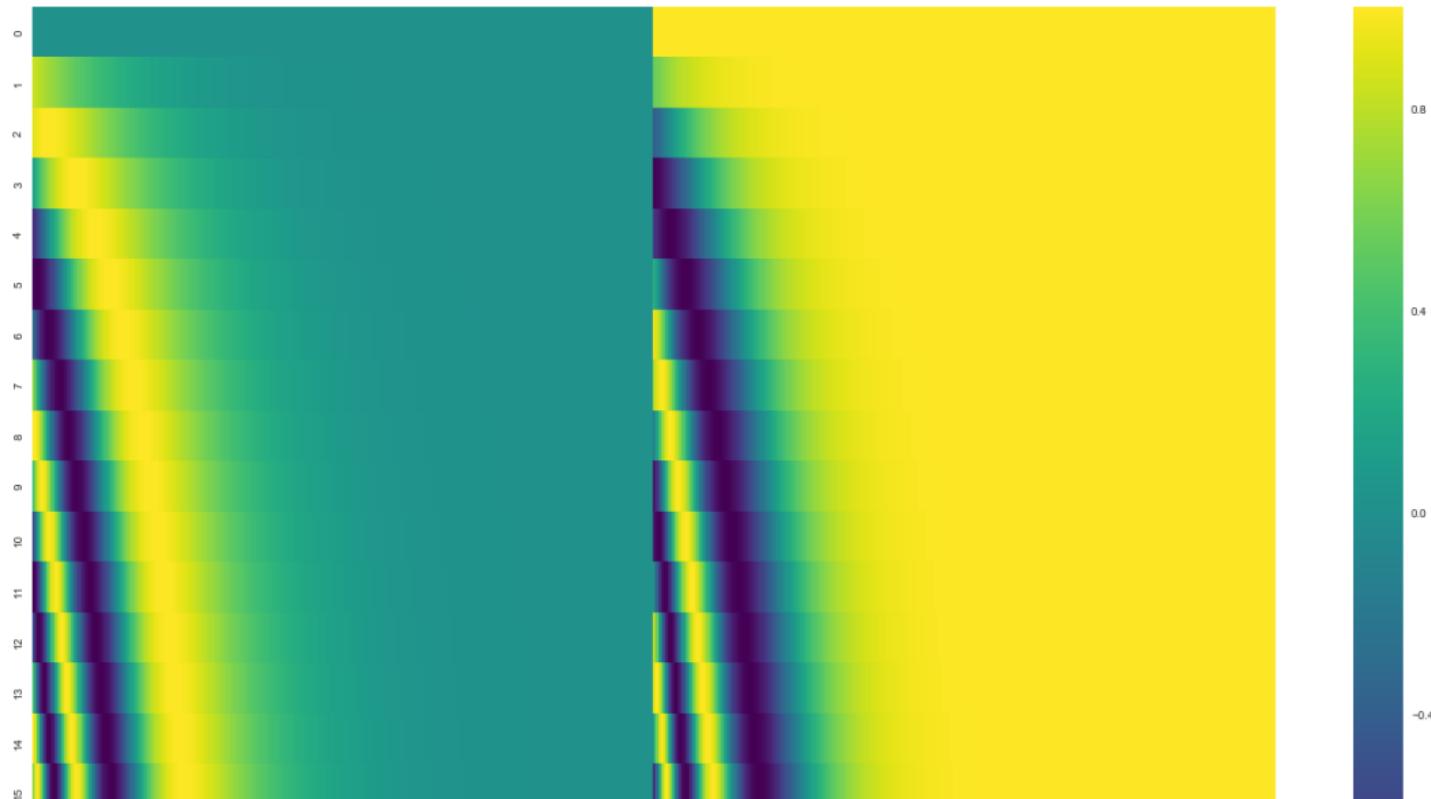
suis

étudiant



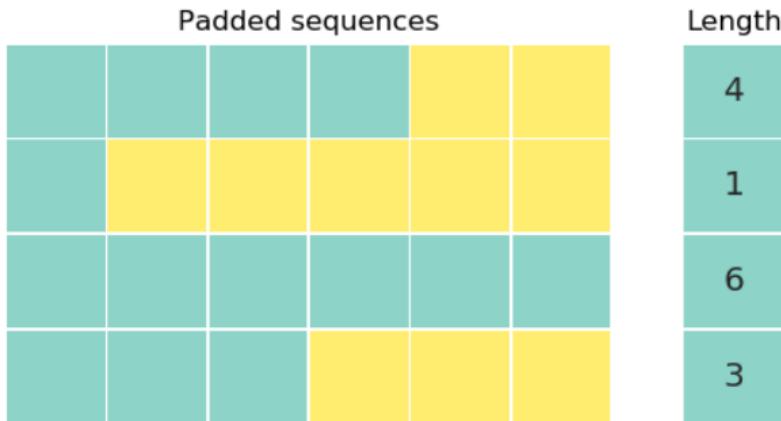
# Position coding

Until now, Transformer does not take position into account !!



# Masking attention computation

Think about **batches** :



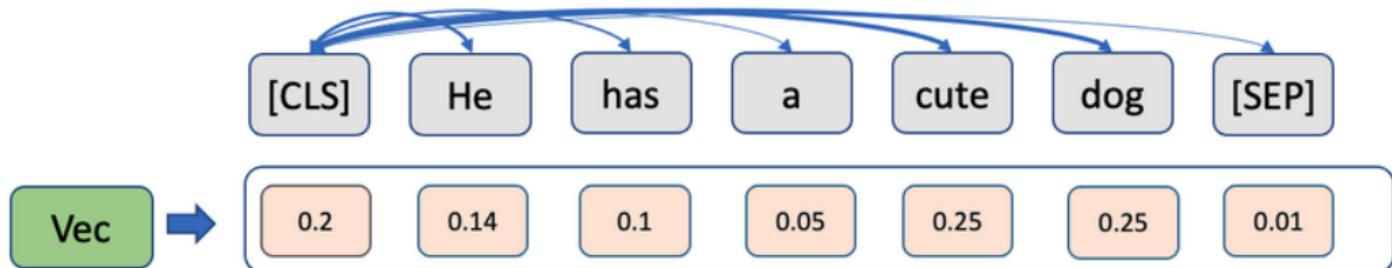
- Lengths are different
- Attention for PAD token =  $-\infty \Rightarrow \alpha_{PAD} = 0$



# Specific tokens

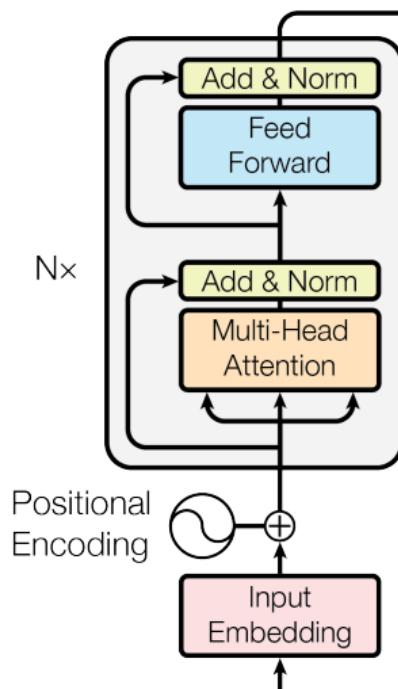
How to split texts or aggregate all informations? ⇒ Introducing specific tokens:

- CLS
- SEP



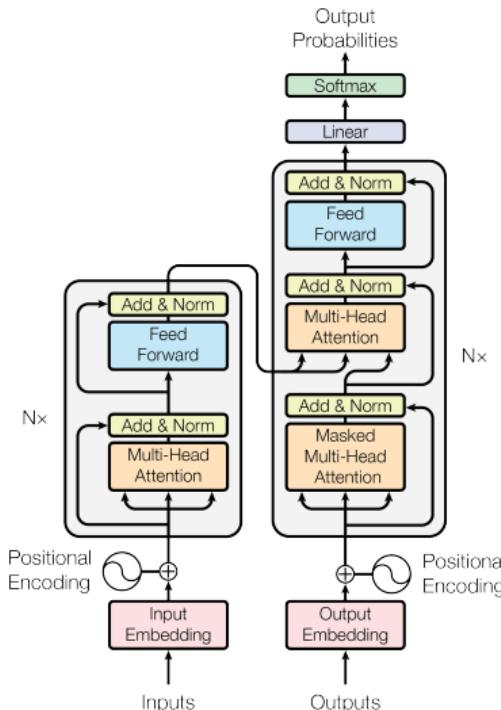
# Different architectures (and different use-cases)

- 1 Encoder-Decoder (original article, T5, ...)
- 2 Encoder Only (BERT)
- 3 Decoder Only (GPT)



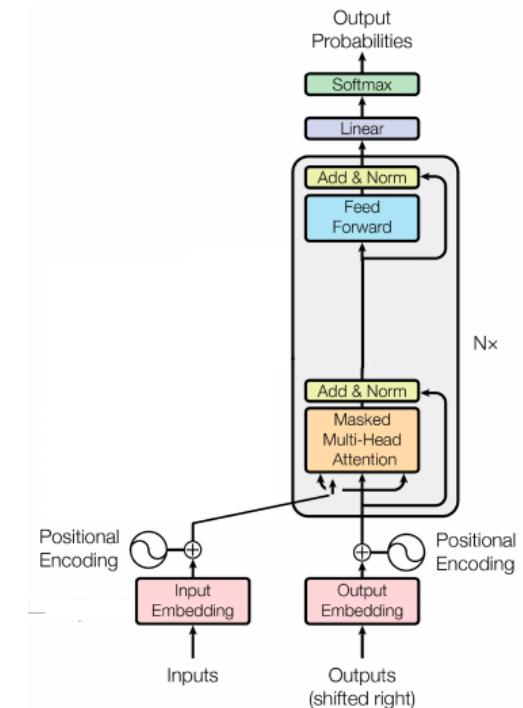
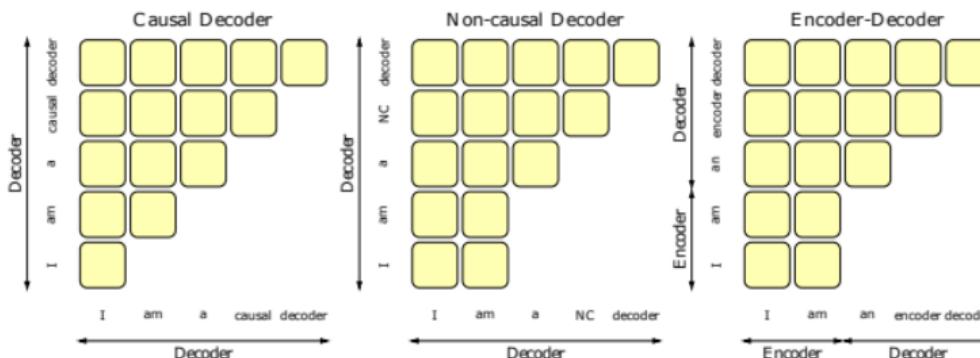
# Different architectures (and different use-cases)

- 1 Encoder-Decoder (original article, T5, ...)
- 2 Encoder Only (BERT)
- 3 Decoder Only (GPT)



# Different architectures (and different use-cases)

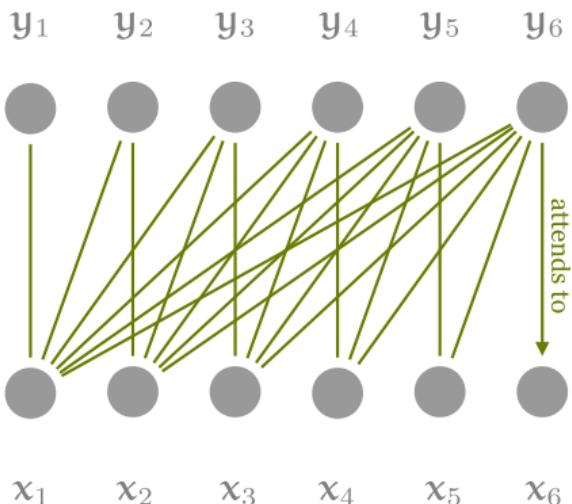
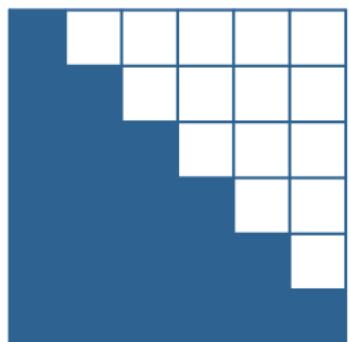
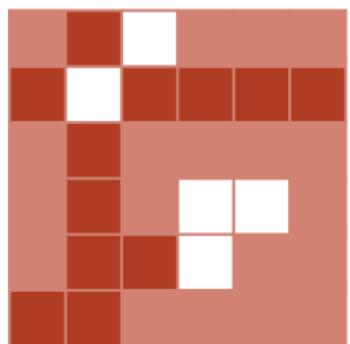
- 1 Encoder-Decoder (original article, T5, ...)
- 2 Encoder Only (BERT)
- 3 Decoder Only (GPT)





# Masking attention computation

Constraint on the attention to improve prediction



raw attention weights

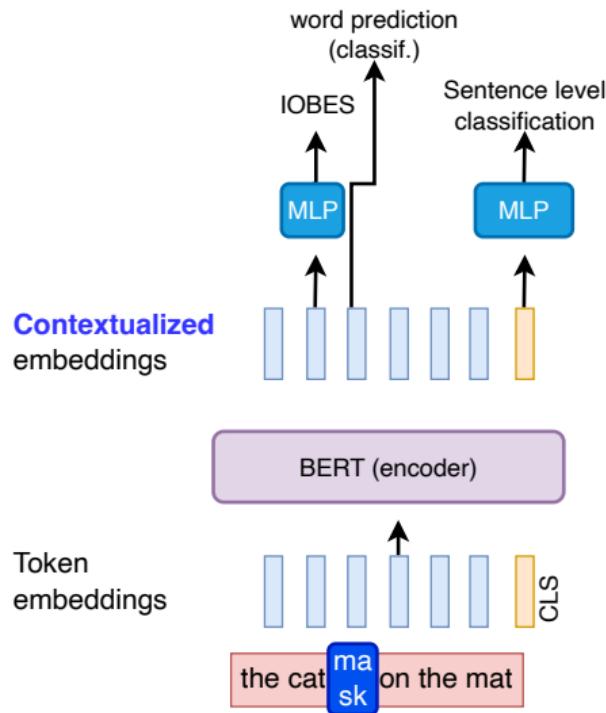
mask

# PRE-TRAINING / TRAINING TASKS



# From Word2Vec to Masked-LM (MLM)

- Masking
- Next token prediction (GPT)



# Next sentence

- Learn to aggregate... cheap manner

On chercher à savoir si une phrase en suit une autre

Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon  
[MASK] milk [SEP]

Label = IsNext

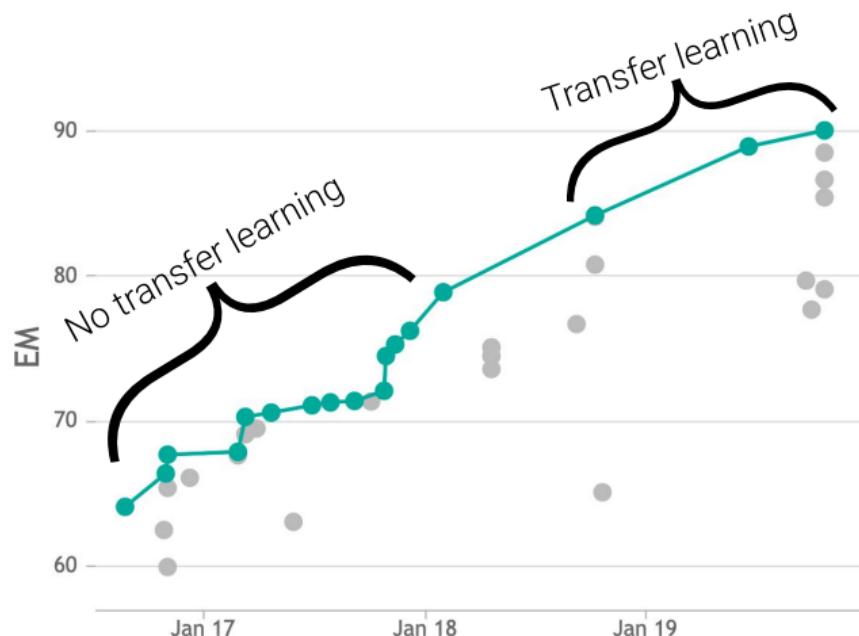
Input = [CLS] the man [MASK] to the store [SEP] penguin [MASK] are  
flight ##less birds [SEP]

Label = NotNext



# T5 architecture: generalizing gen. AI

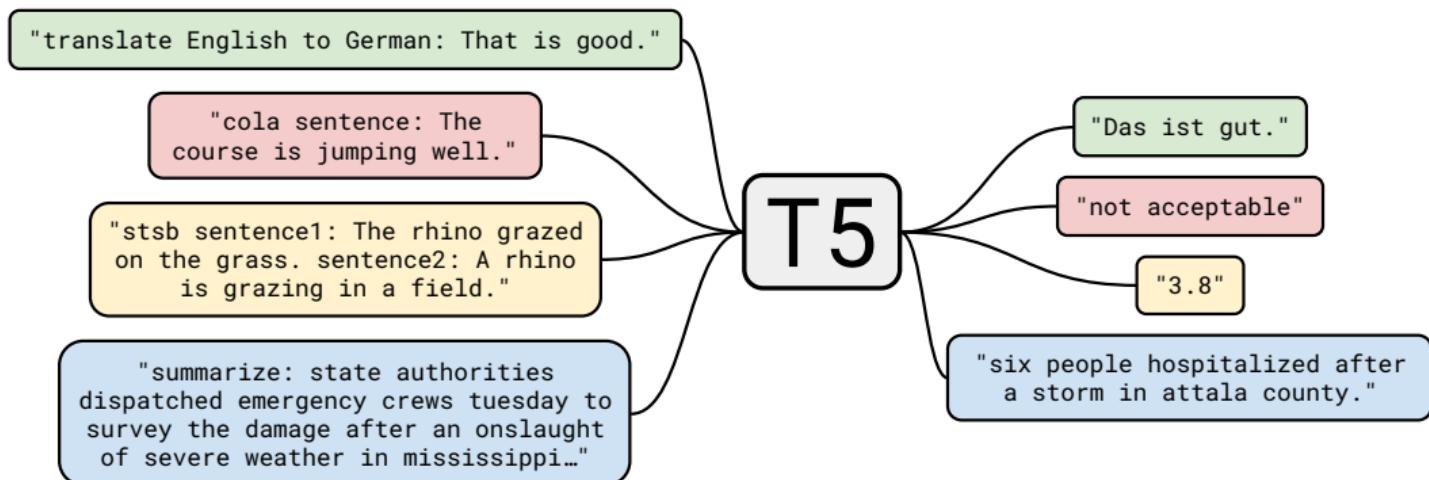
- Refining a generative model to answer different tasks ... T5 :)
- <https://colinraffel.com/talks/mila2020transfer.pdf>



Source: <https://paperswithcode.com/sota/question-answering-on-squad11-dev>

# T5 architecture: generalizing gen. AI

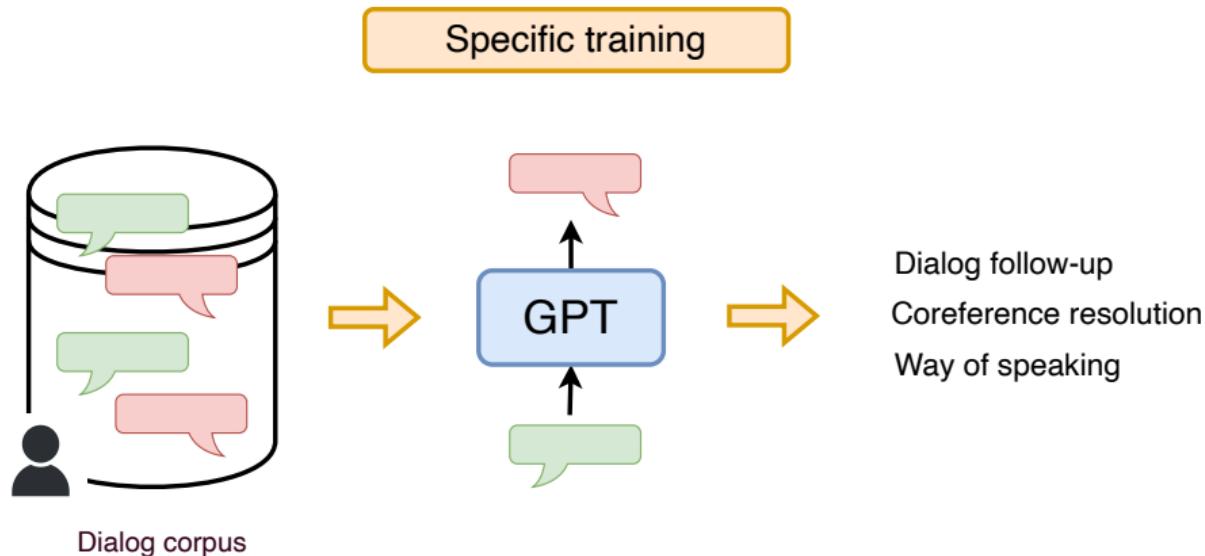
- Refining a generative model to answer different tasks ... T5 :)
- <https://colinraffel.com/talks/mila2020transfer.pdf>





# The Ingredients of chatGPT

## 2. Dialogue Tracking



■ **Very clean** data

Data generated/validated/ranked by humans



# The Ingredients of chatGPT

## 3. Fine-tuning on different ( $\pm$ ) complex reasoning tasks

### Instruction finetuning

Please answer the following question.

What is the boiling point of Nitrogen?

### Chain-of-thought finetuning

Answer the following question by reasoning step-by-step.

The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

-320.4F

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ .

Language model

### Multi-task instruction finetuning (1.8K tasks)

### Inference: generalization to unseen tasks

Q: Can Geoffrey Hinton have a conversation with George Washington?

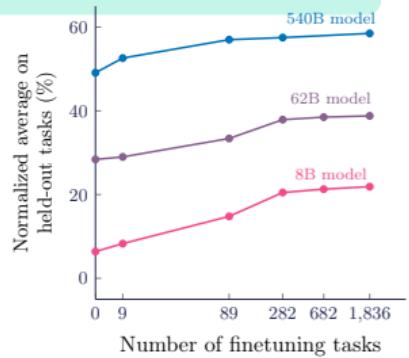
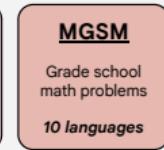
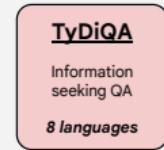
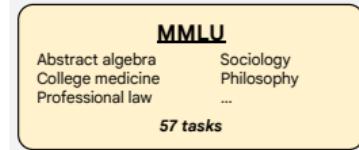
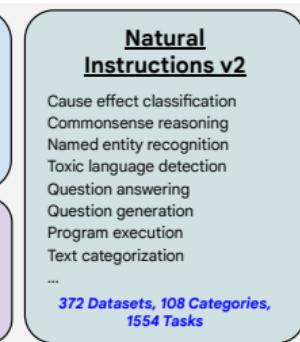
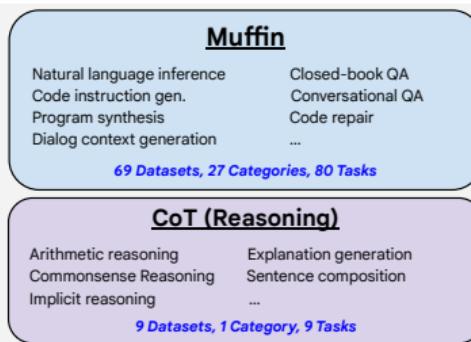
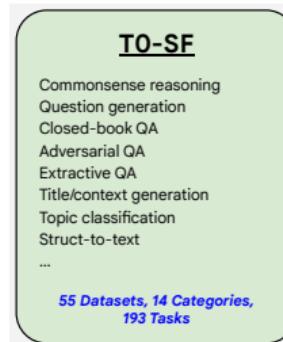
Give the rationale before answering.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is "no".

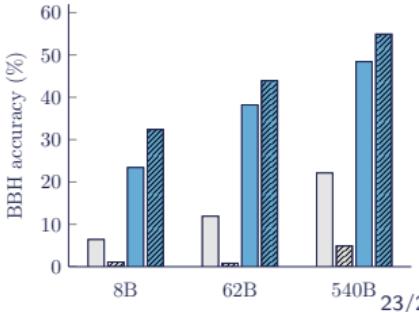


# The Ingredients of chatGPT

## 3. Fine-tuning on different ( $\pm$ ) complex reasoning tasks

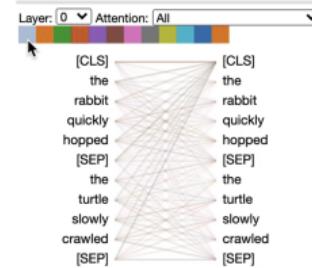


- PaLM: Zero-shot
- PaLM: Zero-shot + CoT
- Flan-PaLM: Zero-shot
- Flan-PaLM: Zero-shot + CoT



# EXPLOITATION

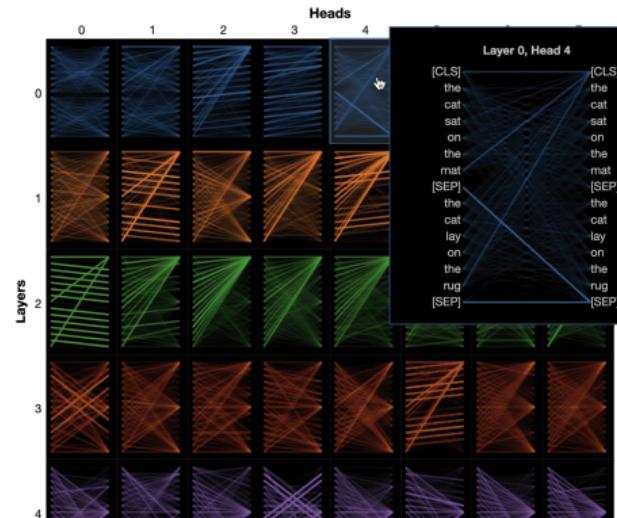
# Layer visualization



Model View

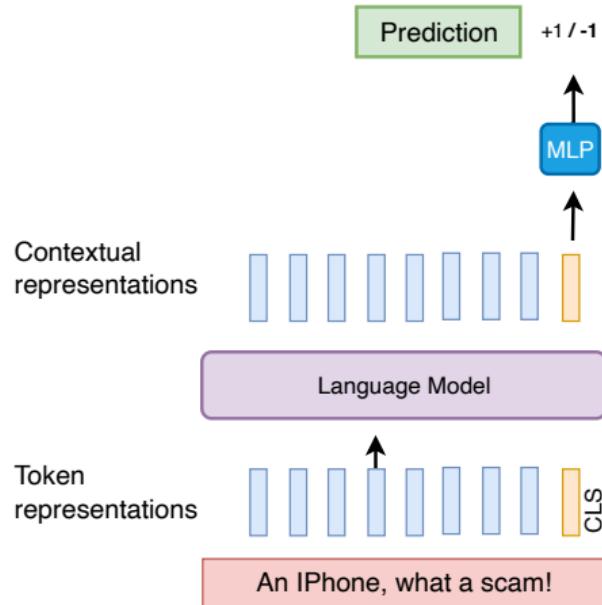
The *model view* shows a bird's-eye view of attention across all layers and heads.

Try out the model view in the [Interactive Colab Tutorial](#) (all visualizations pre-loaded).

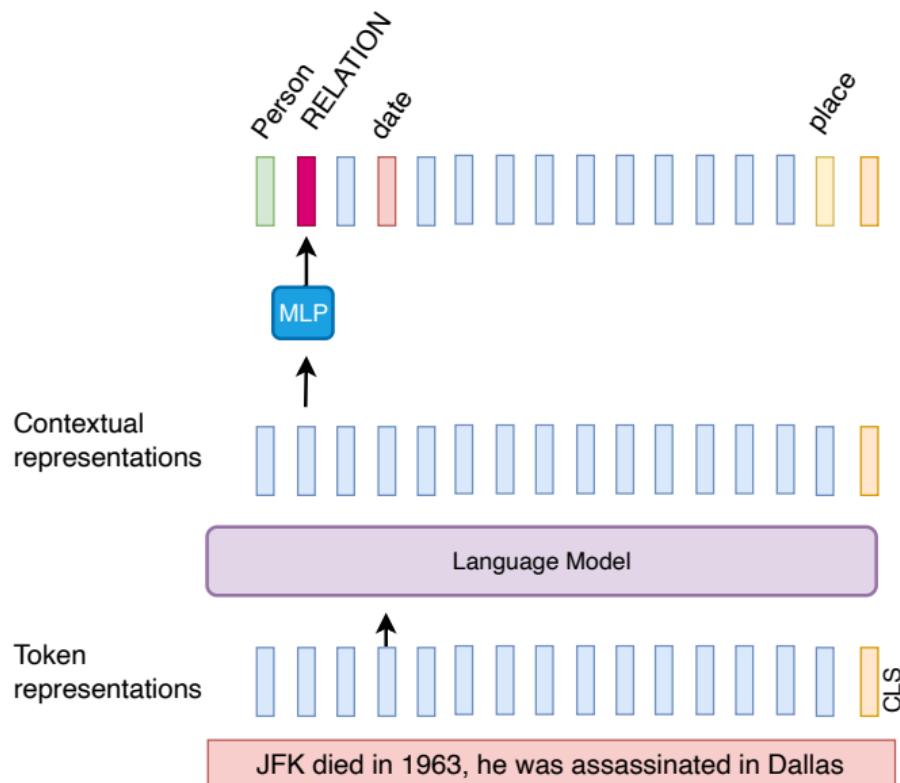


- BERT viz: <https://github.com/jessevig/bertviz>

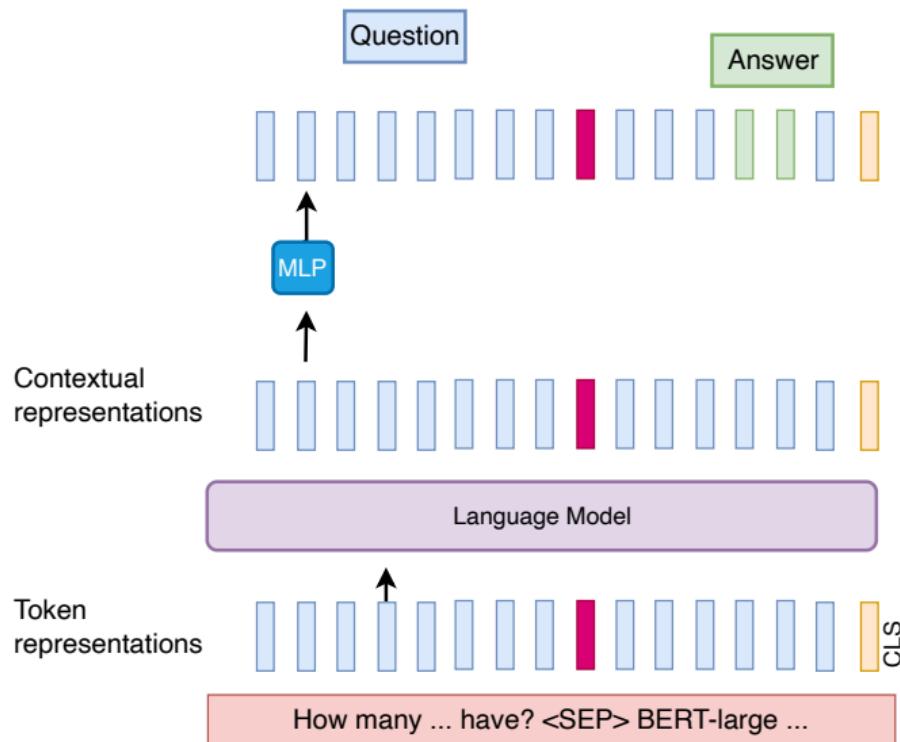
# Example BERT



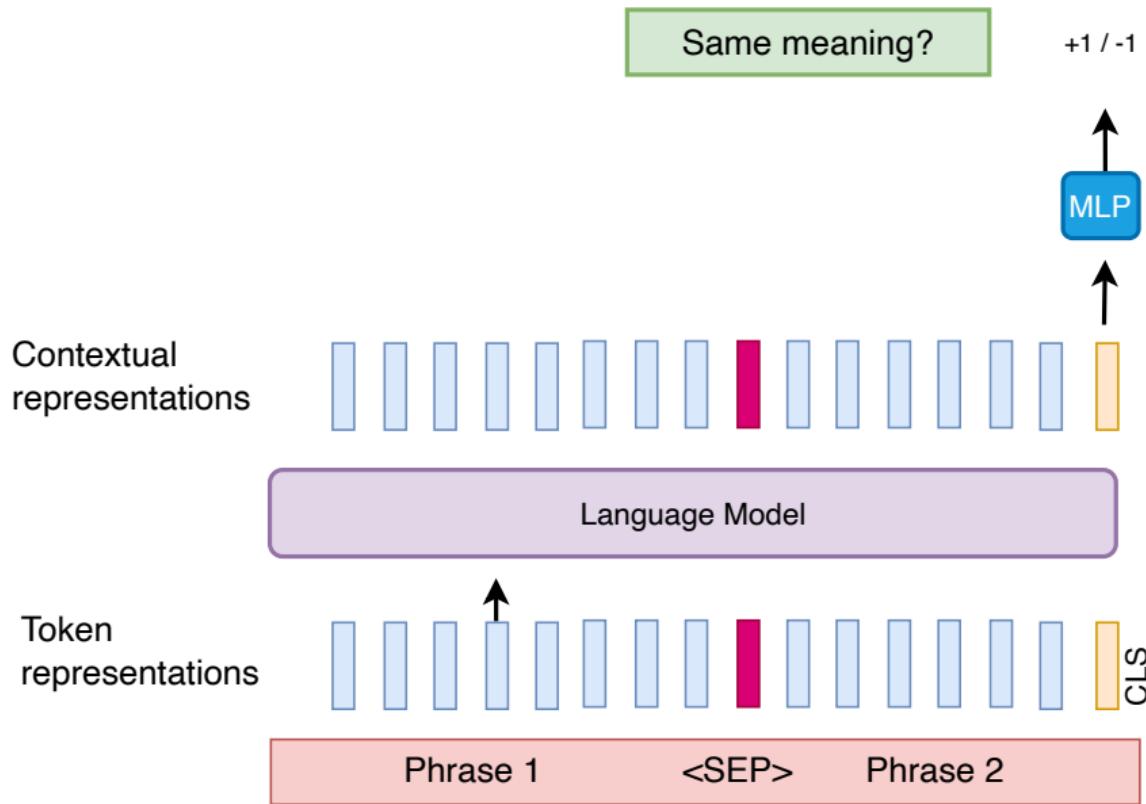
# Example BERT



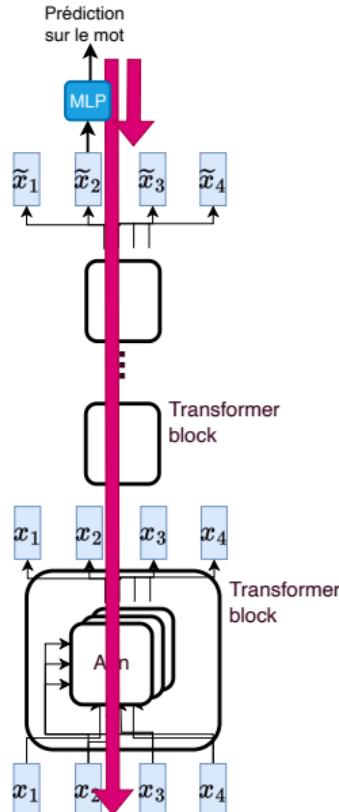
# Example BERT



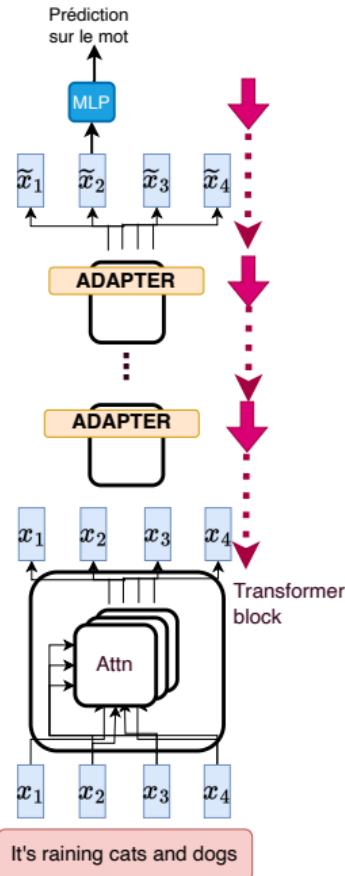
# Example BERT



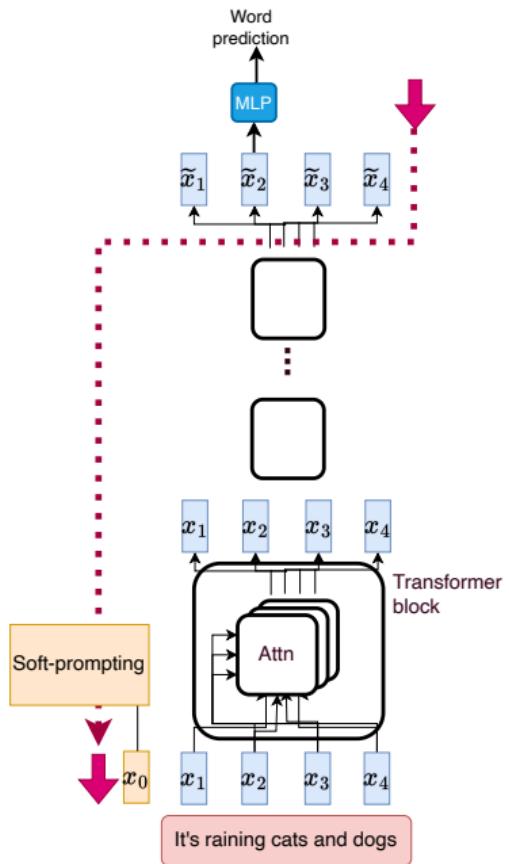
# Different fine-tuning strategies



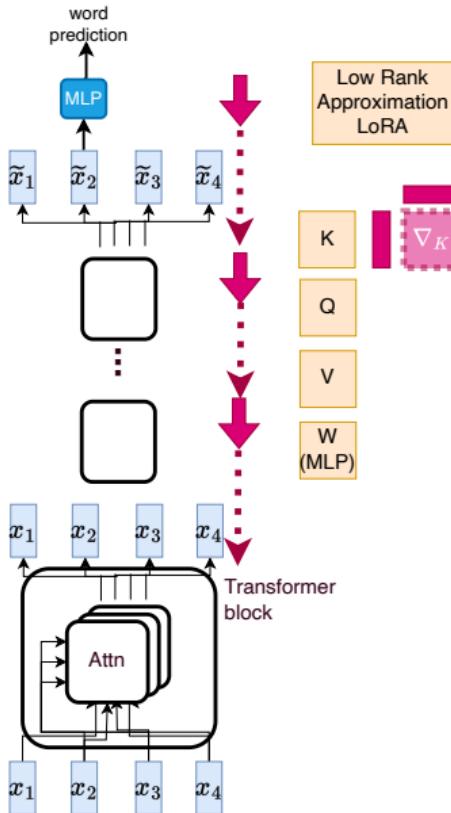
# Different fine-tuning strategies



# Different fine-tuning strategies



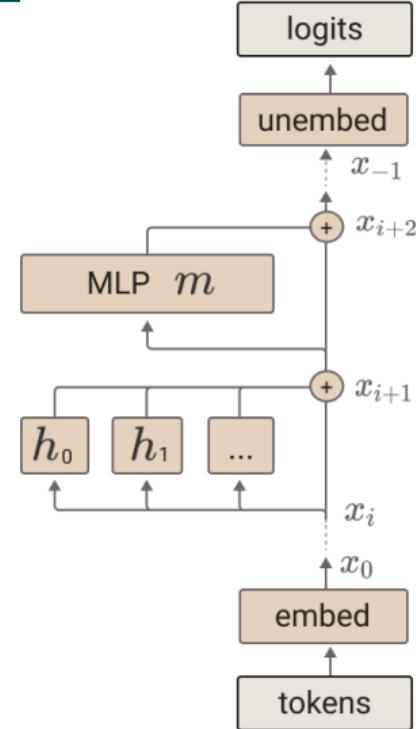
# Different fine-tuning strategies



It's raining cats and dogs

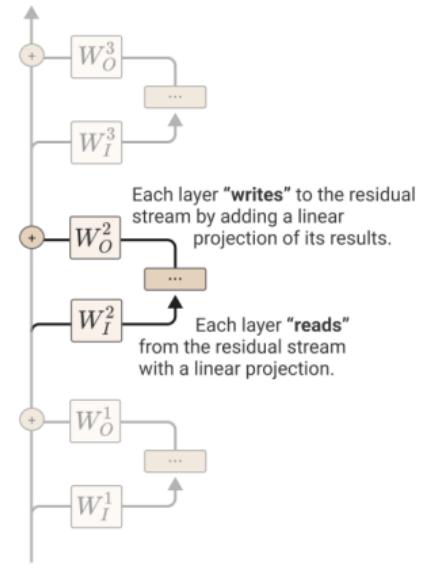
# Care of novelty: improvements are not guaranteed

- New ways to see the transformer architecture  
(Anthropic)



# Care of novelty: improvements are not guaranteed

- New ways to see the transformer architecture  
(Anthropic)



<https://transformer-circuits.pub/2021/framework/index.html>

# Care of novelty: improvements are not guaranteed

- New ways to see the transformer architecture  
(Anthropic)
- Time series

Are Transformers Effective for Time Series Forecasting? Zeng et al. 2022