université PARIS-SACLAY

TRANSFORMER ARCHITECTURE

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

Introduction

Aggregation problem

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

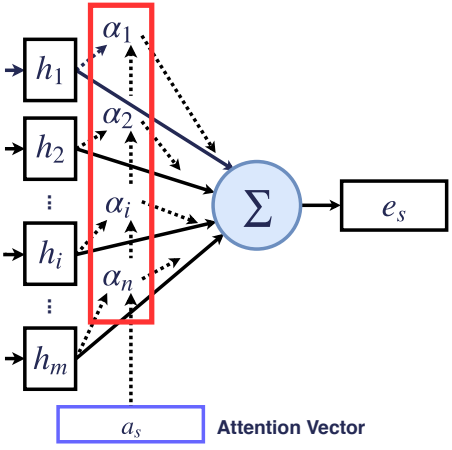
Representation learning : $s_i \rightarrow \mathbf{e}_i \in \mathbb{R}^Z$

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 weigthed sum

Attention & aggregation *vs* Diffusion process



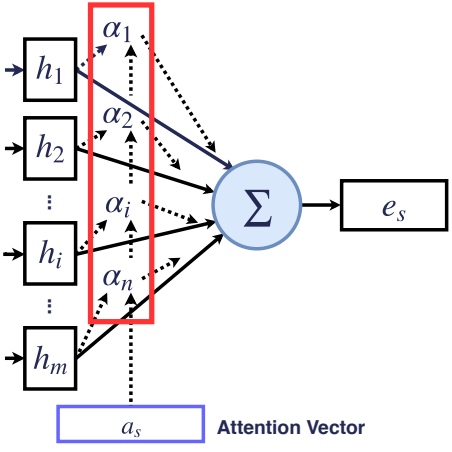
- Learning to weight localized representation
- Many efficient applications [2014-2018]
- Few parameters
- Attention vector = Query vector
- How to code it in `pytorch`?

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 $\Rightarrow$ 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
```

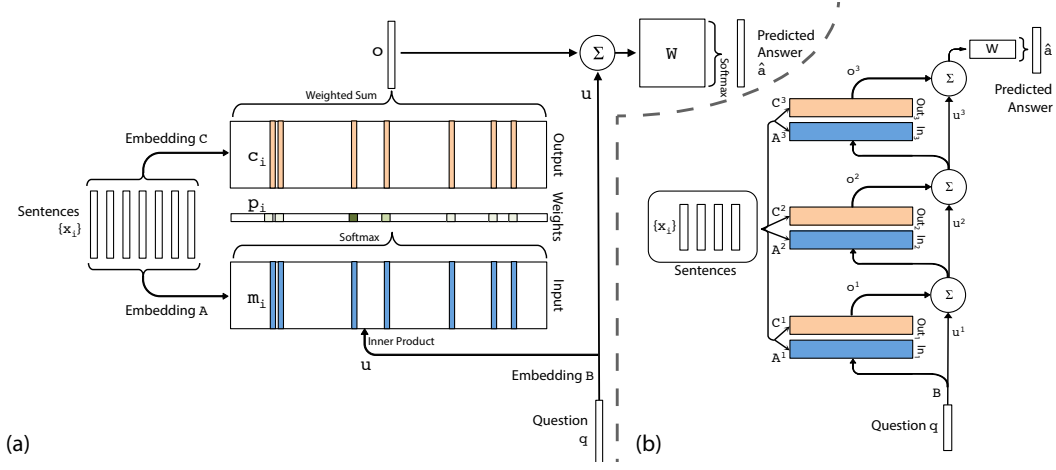
```
Mary journeyed to the den.
Mary went back to the kitchen.
John journeyed to the bedroom.
Mary discarded the milk.
Q: Where was the milk before the den?
A. Hallway
```

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?

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
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
Where is John? Answer: bathroom	**Prediction: bathroom**			

Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
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
Where is the milk? Answer: hallway	**Prediction: hallway**			

Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
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
What color is Greg? Answer: yellow	**Prediction: yellow**			

Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
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
Does the suitcase fit in the chocolate? Answer: no	**Prediction: no**			

Conclusion

Current limitations with RNN:

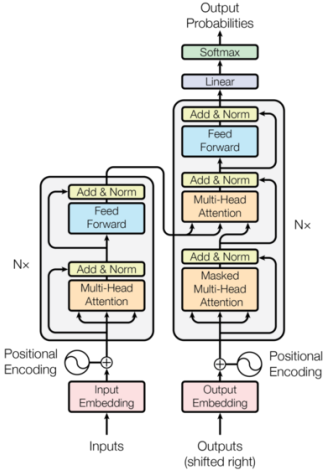
- Long dependencies
- Global message representation
  - Continuous indexing
- Targeted dependencies
  - Co-reference resolution

Architecture

Transformer generality

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

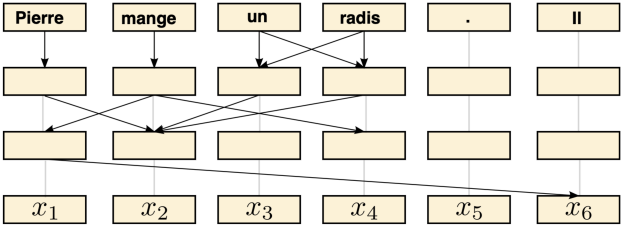
- J. Alammar 2018 `http://jalammar.github.io/illustrated-transformer/`
- Alexander Rush 2018 `http://nlp.seas.harvard.edu/2018/04/03/attention.html`
- P. Bloem 2019 `http://www.peterbloem.nl/blog/transformers`



Overcoming classical pictures
$\Rightarrow$ 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

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

les chats sont des félidés

est transformé en

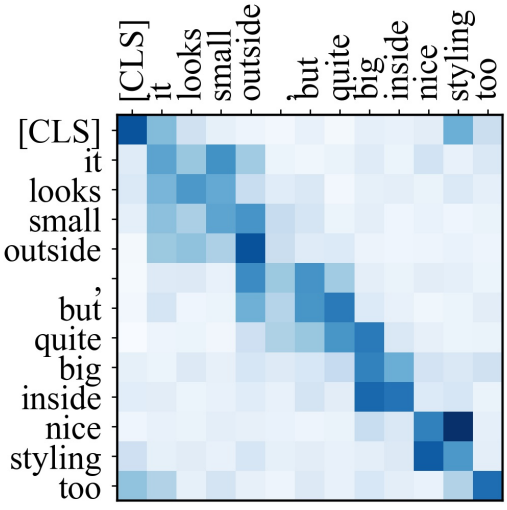
`[CLS] _les _chats _sont _des _f éli dés [SEP]`

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)

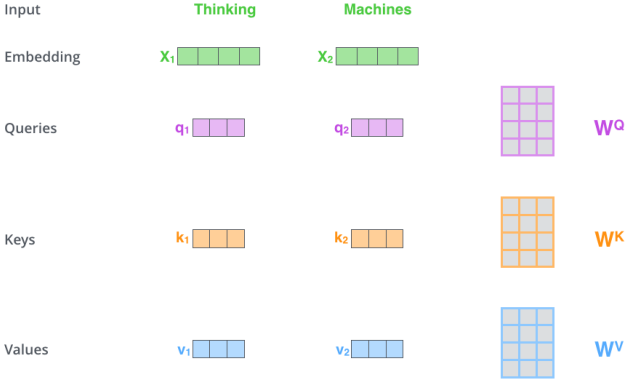


Attention block

Interactions through vectors derived from the embedding

1. Compute `Keys`, `Queries`, `Values` by linear mapping

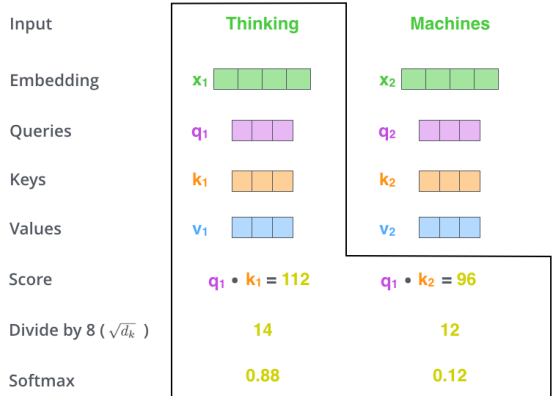
Note: here are the main parameters of the transformer architecture



Attention block

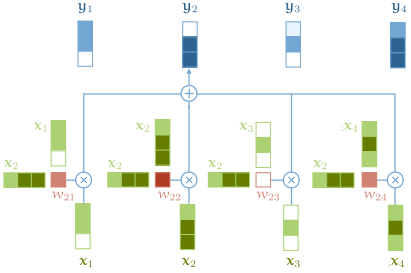
Computing the attention

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



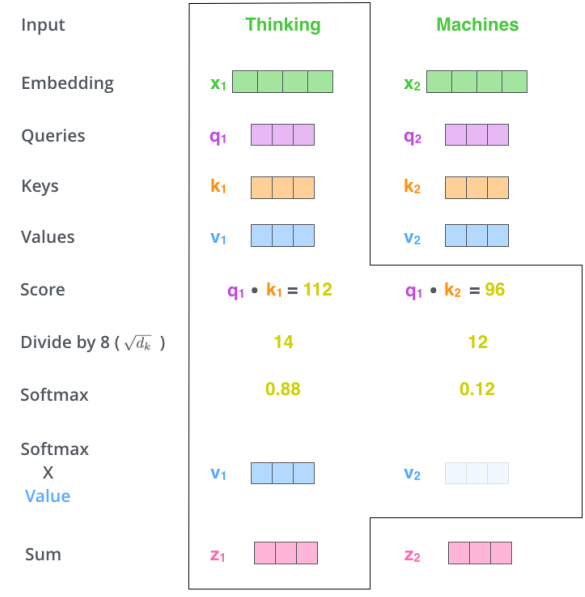
Attention block

New values:



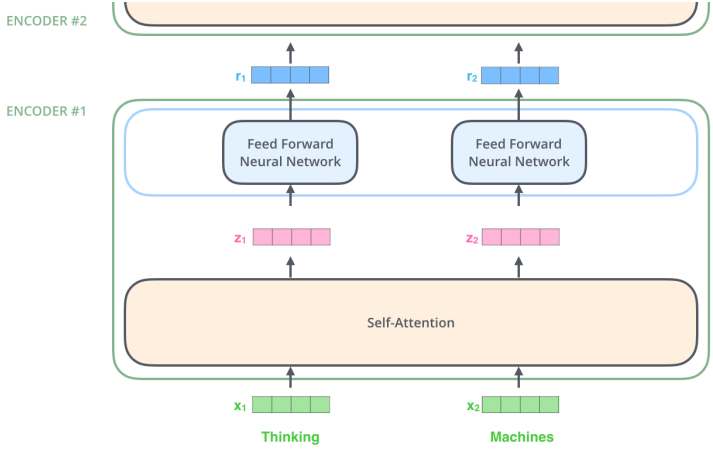
$$v_i^{\ell+1} = \sum \alpha_k v_k^{\ell}$$

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



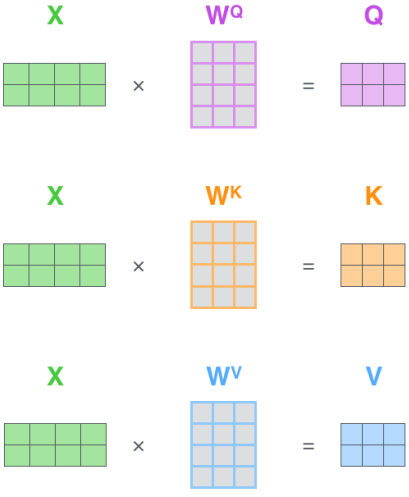
Attention block

Use a fully connected to map to embedding space



Parallelism ⇒ Batch of data

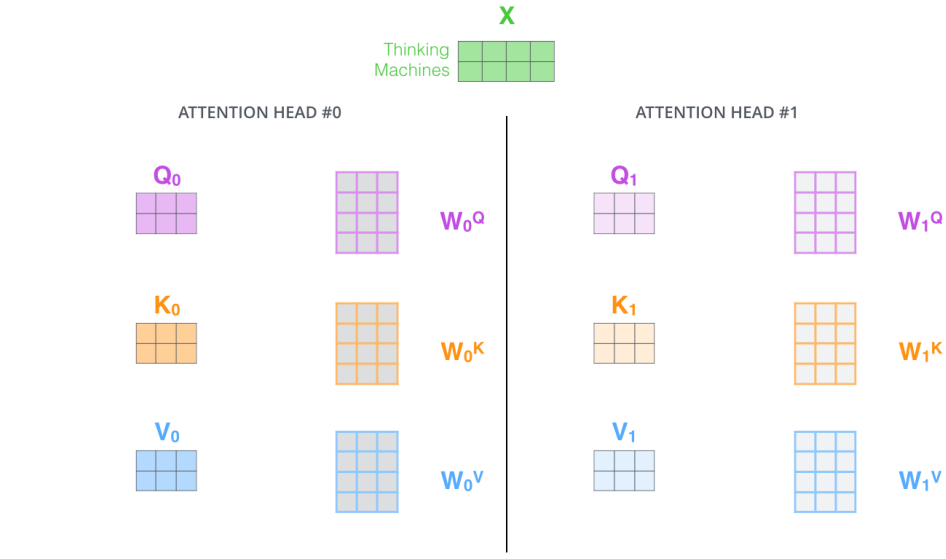
All tokens at once !



$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

Multiple heads : dimension compatibility challenge

Why only one interaction?



Multiple heads : dimension compatibility challenge

Why only one interaction?

1) This is our input sentence*

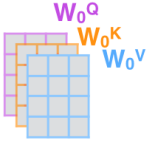
2) We embed each word*

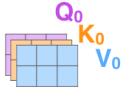
3) Split into 8 heads. We multiply X or R with weight matrices

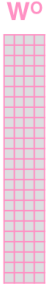
4) Calculate attention using the resulting Q/K/V matrices

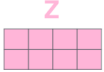
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

Thinking Machines









* 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 concateneted then mapped to rebuild embeddings

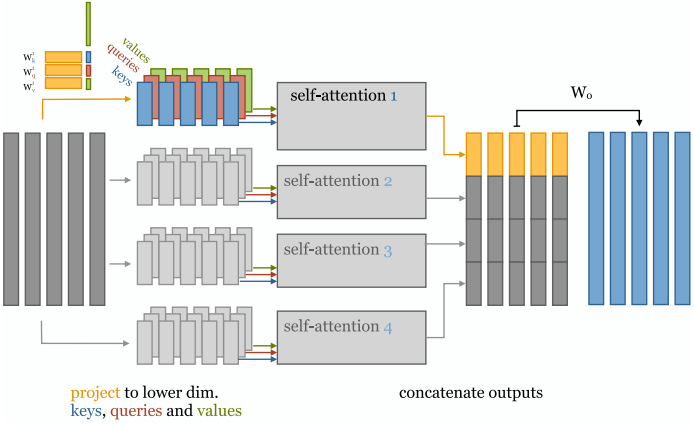
BERT

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

$\Rightarrow 12 \times 64 = 768$

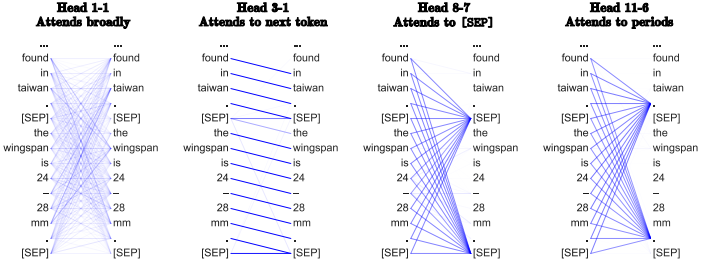
ChatGPT:

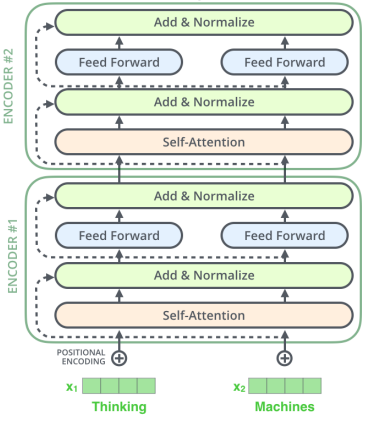
- Embedding : 12288
- key/query/values : 128
- 96 heads



Multiple layers

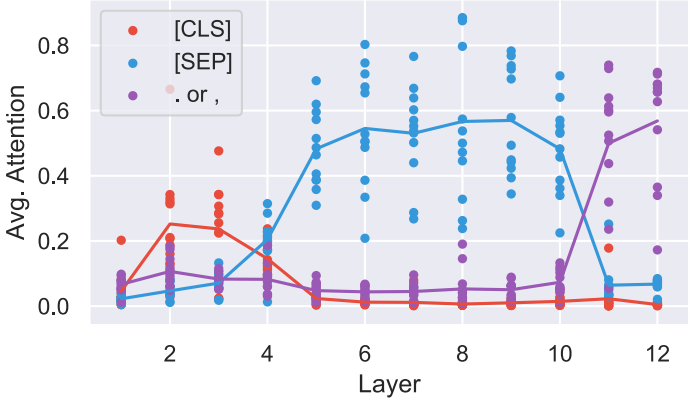
- Have a look on the skip connexion
- Diffusion is made step-by-step
- Bertology shows that different relations are extracted at different scales

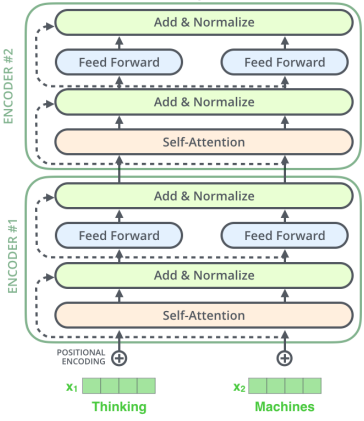




Multiple layers

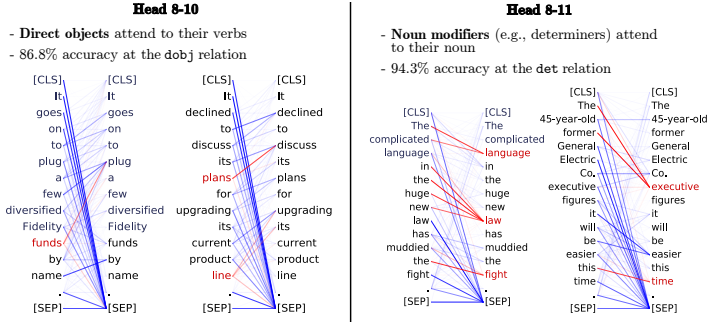
- Have a look on the skip connexion
- Diffusion is made step-by-step
- Bertology shows that different relations are extracted at different scales

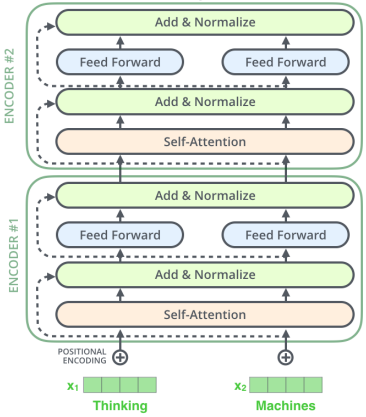




Multiple layers

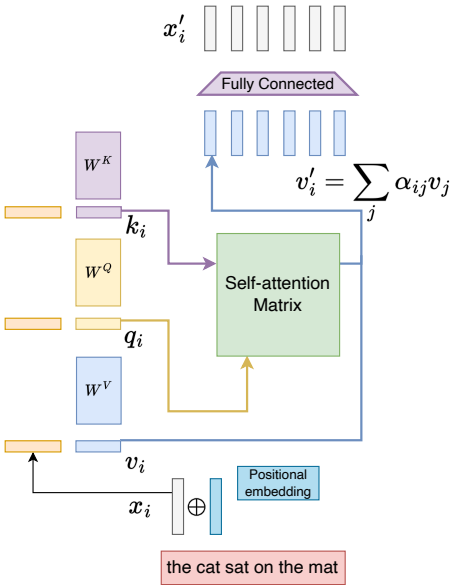
- Have a look on the skip connexion
- Diffusion is made step-by-step
- Bertology shows that different relations are extracted at different scales





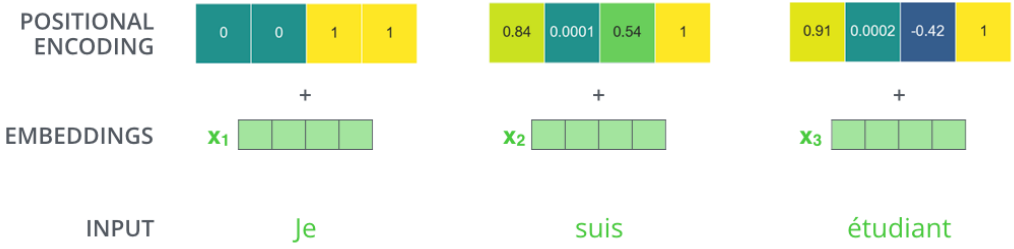
Position coding

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



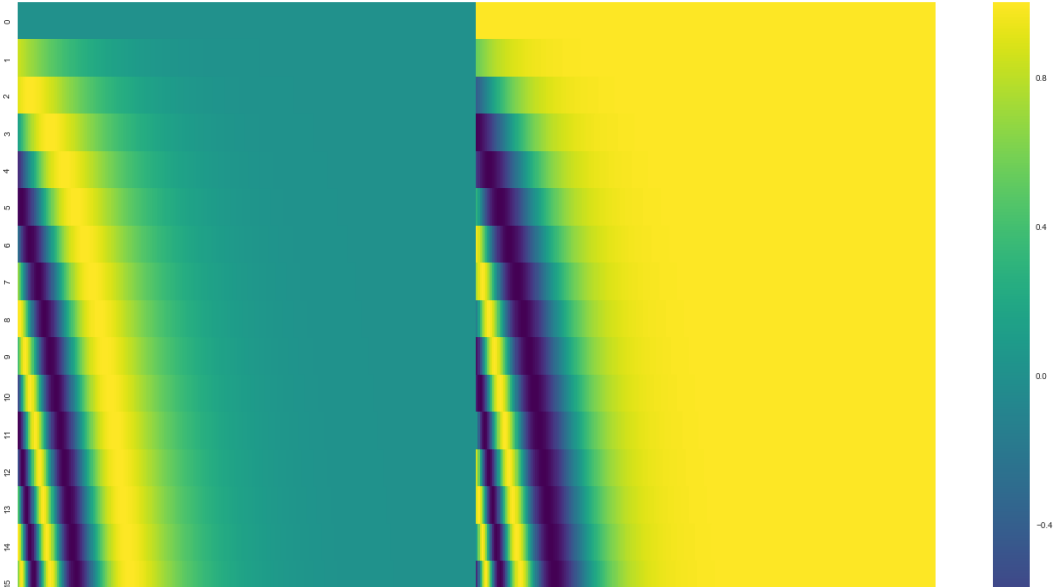
Position coding

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



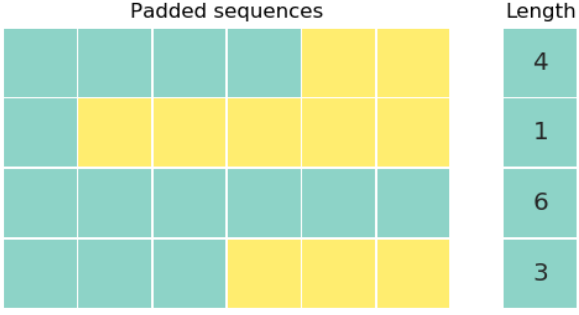
Position coding

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



Masking attention computation

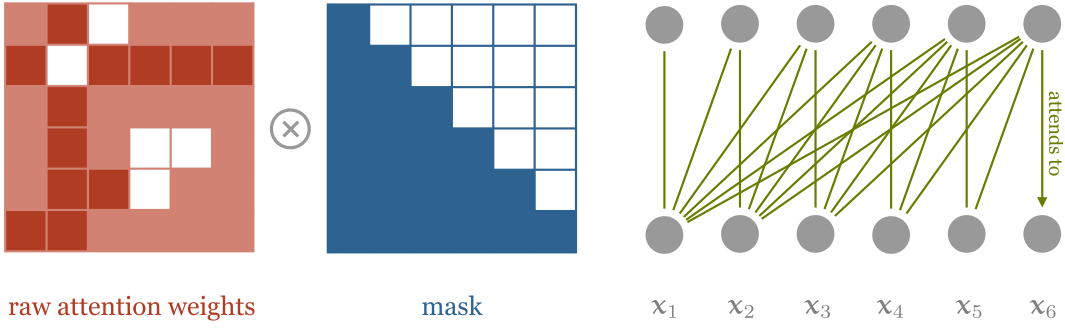
Think about **batches** :



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

Masking attention computation

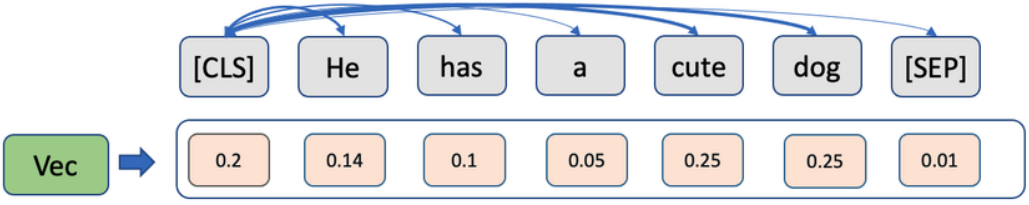
Constraint on the attention to improve prediction



Specific tokens

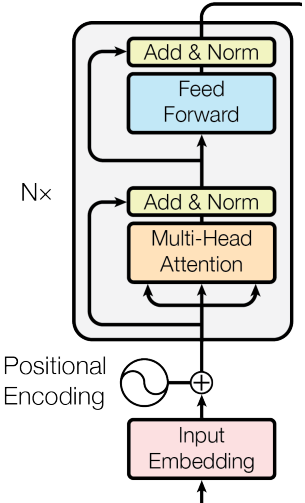
How to split texts or aggregate all informations? $\Rightarrow$ Introducing specific tokens:

- CLS
- SEP



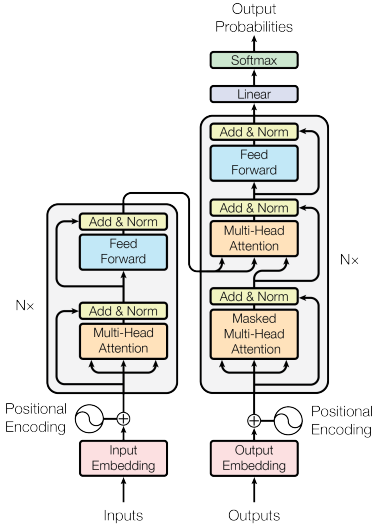
Different architectures (and different use-cases)

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



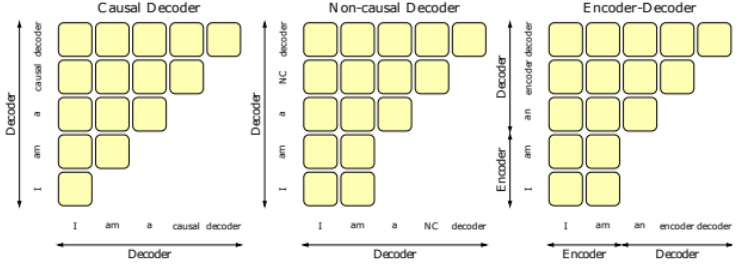
Different architectures (and different use-cases)

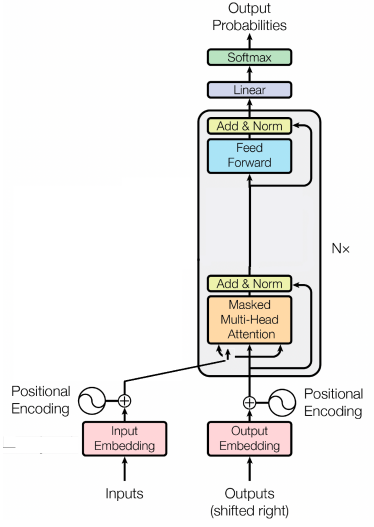
1. Encoder-Decoder (original article, T5, ...)
2. Encoder Only (BERT)
3. Decoder Only (GPT)



Different architectures (and different use-cases)

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

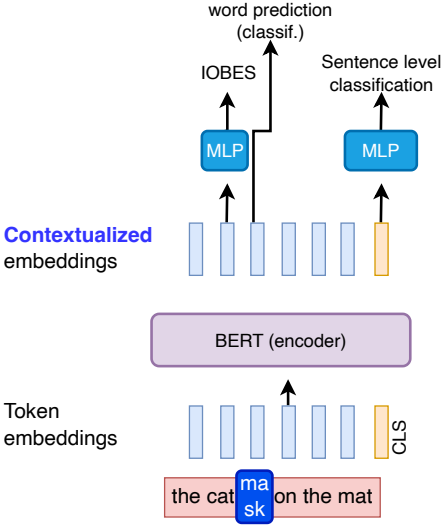




Pre-training / training tasks

From W2V to MLM

- Masking
- Next token prediction (GPT)



Next sentence

- Learn to aggragate... cheap manner

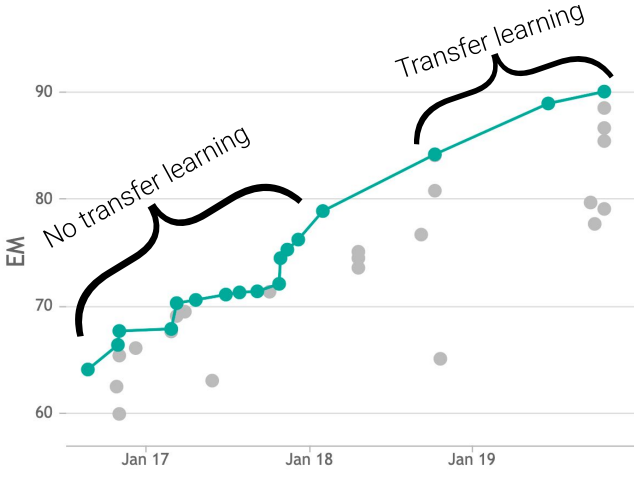
On chercher à savoir si une phrase en suit une autre

```
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

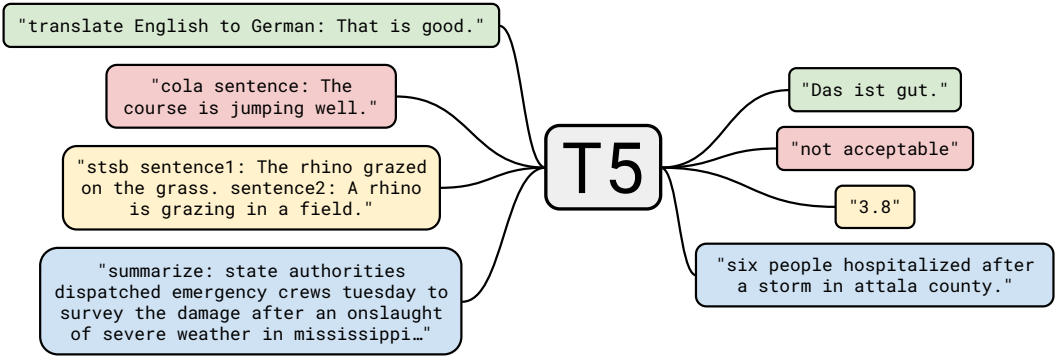
- Refining a generative model to answer different tasks ... T5 :)
- `https://colinraffel.com/talks/mila2020transfer.pdf`



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

T5 architecture: generalizing gen. AI

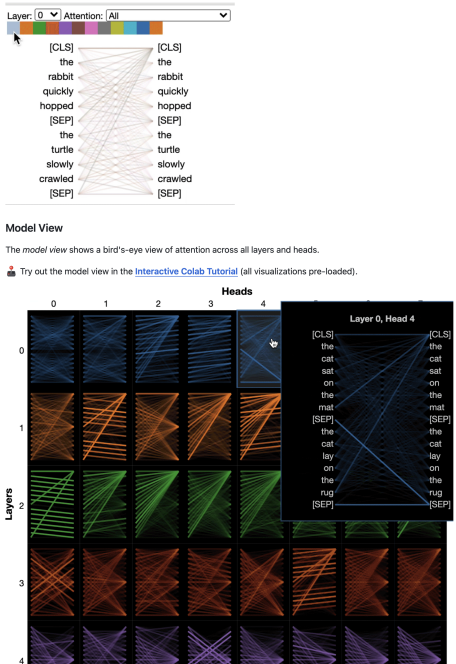
- Refining a generative model to answer different tasks ... T5 :)
- https://colinraffel.com/talks/mila2020transfer.pdf



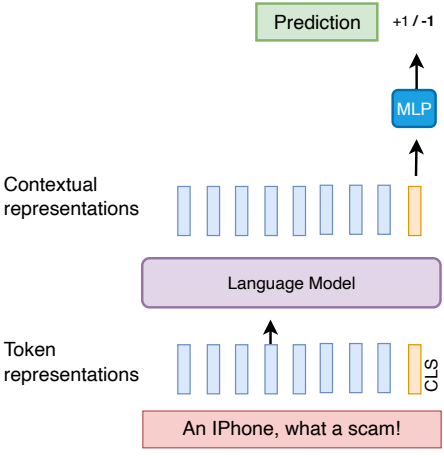
Exploitation

Layer visualization

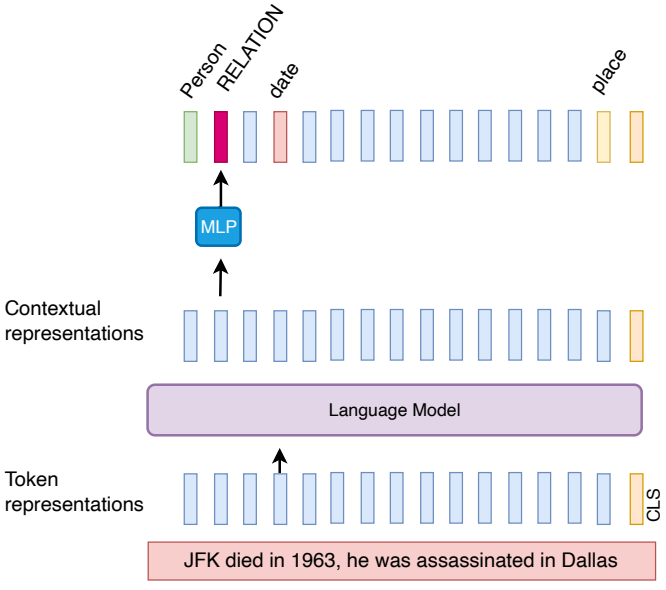
- BERT viz: `https://github.com/jessevig/bertviz`



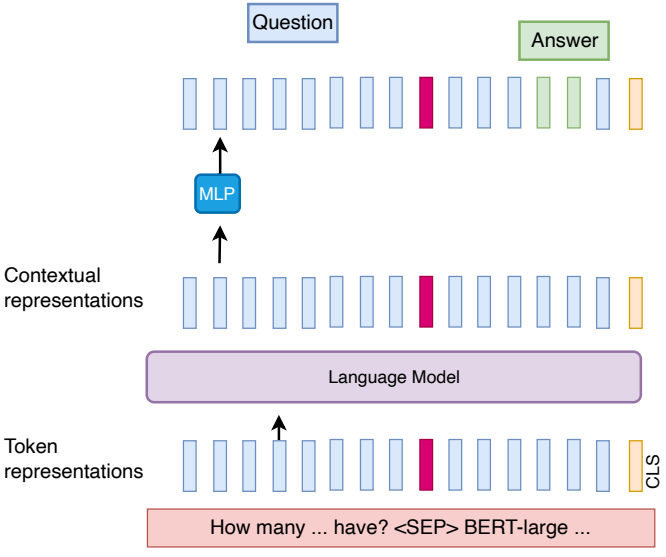
Example BERT



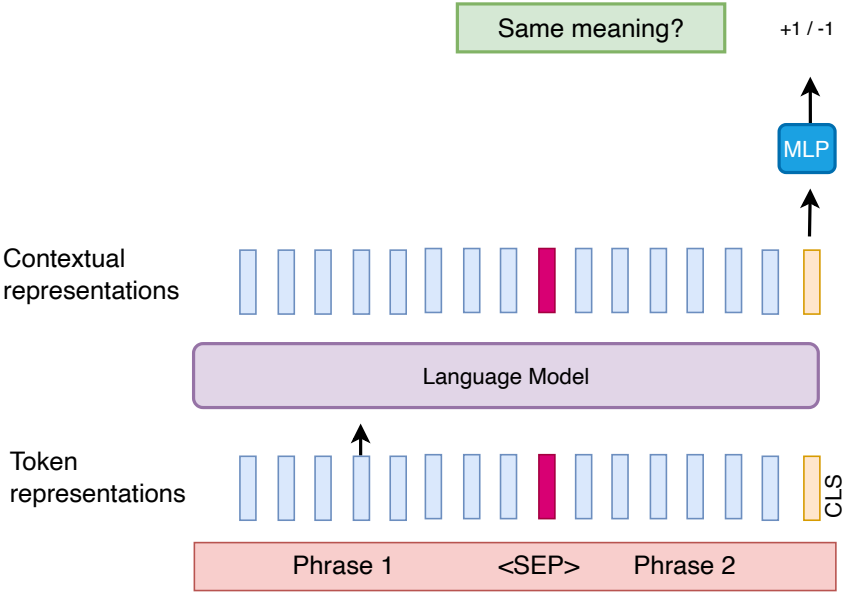
Example BERT



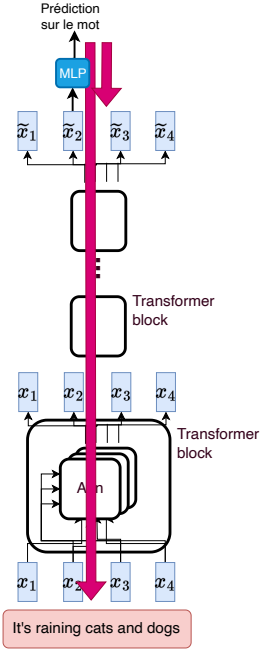
Example BERT



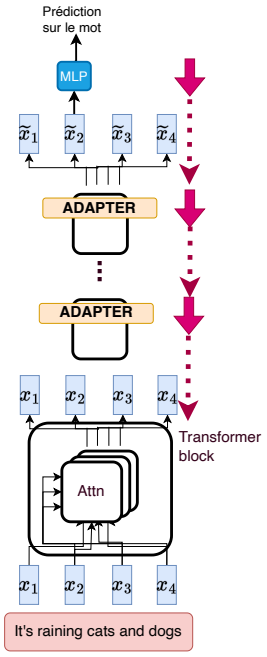
Example BERT



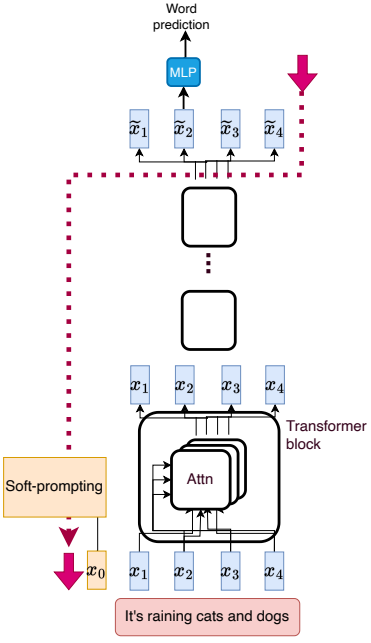
Different fine-tuning strategies



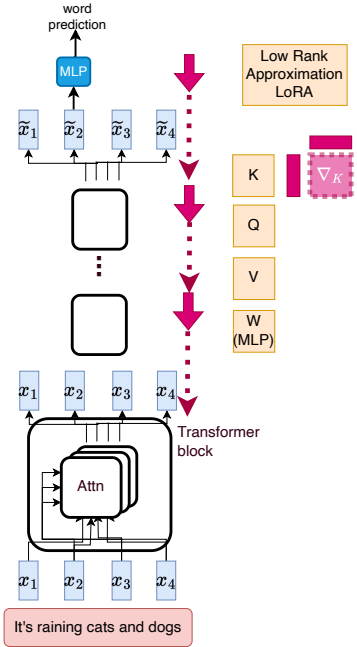
Different fine-tuning strategies



Different fine-tuning strategies

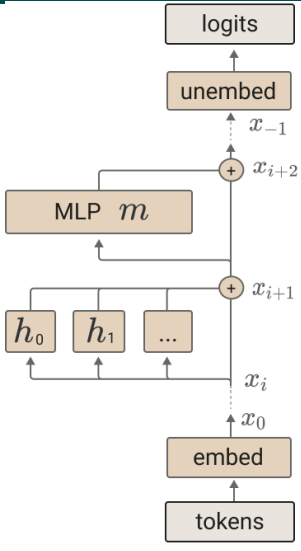


Different fine-tuning strategies



Care of novelty: improvments are not guaranted

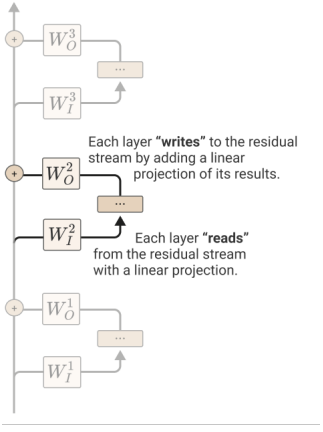
- New ways to see the transformer architecture (Anthropic)



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

Care of novelty: improvments are not guaranted

- New ways to see the transformer architecture (Anthropic)



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

Care of novelty: improvments are not guaranted

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

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