UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Ethical Aspects of Data

Fairness, Bias, in "LLM Core"

Frederic Precioso

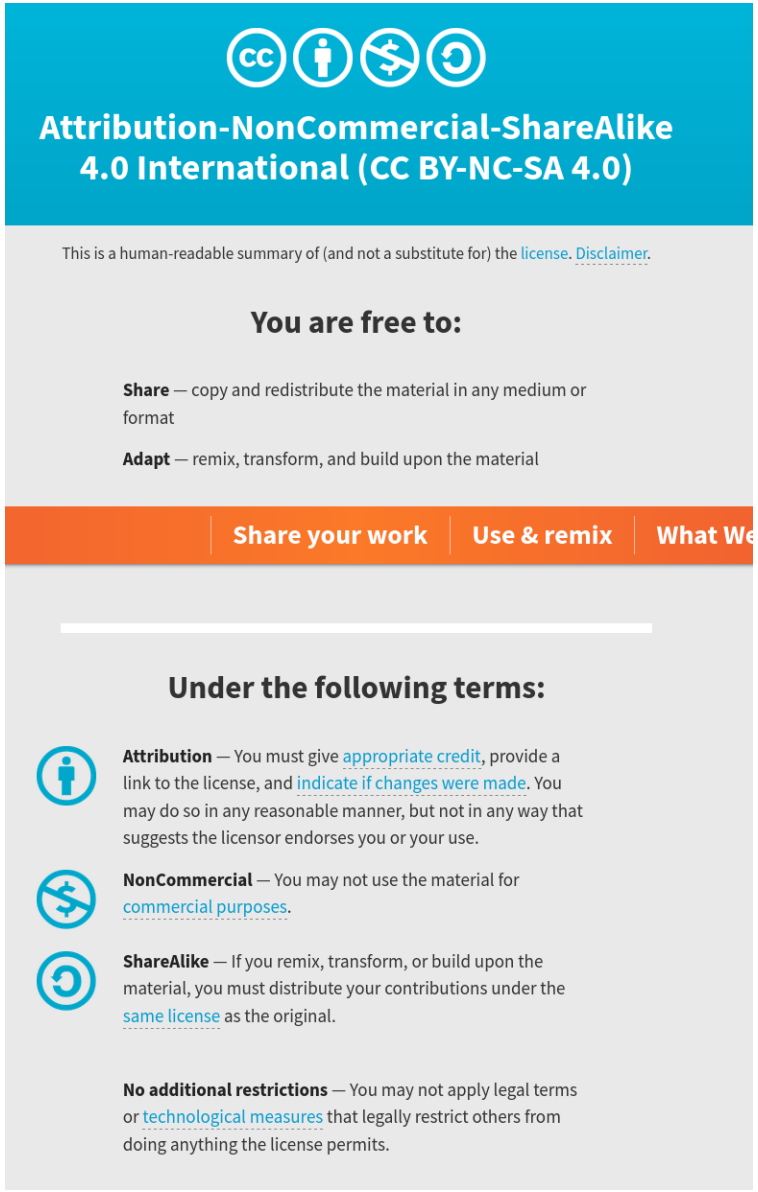
29/11/2023

(MAASAI, Joint Research Group INRIA-CNRS-UniCA)

frederic.precioso@univ-cotedazur.fr

License for this content:
CC BY-NC-SA

- Training for Data Science & AI Master at UniCA by Frederic Precioso under Licence Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- **Reminder**
- Multimodal correlations (self-attention)
- Transformers
- Biases
- Computational cost, Energy, and Sovereinty

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

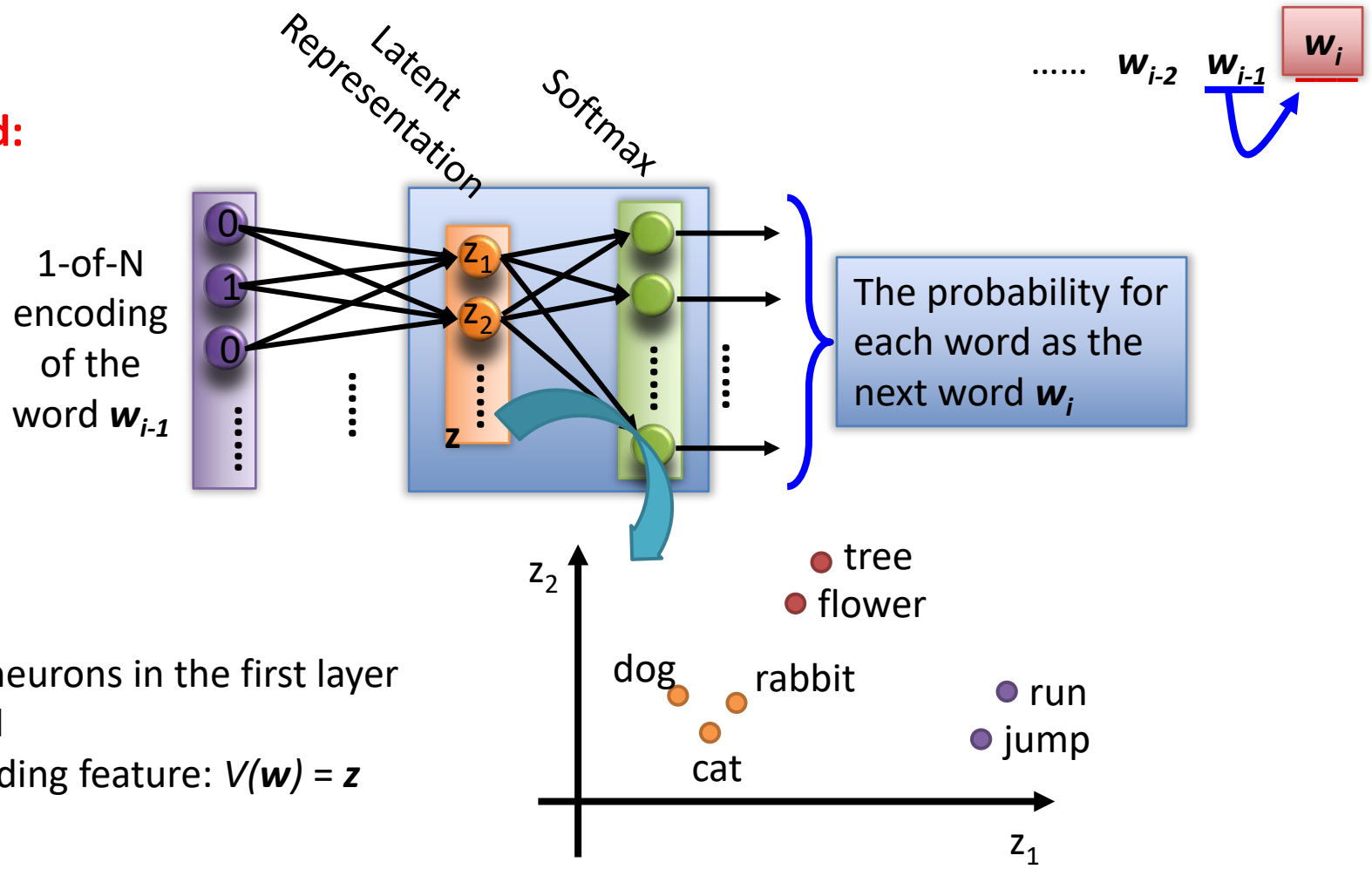
REMINDER

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Reminder - Glove

Predicting the next word:

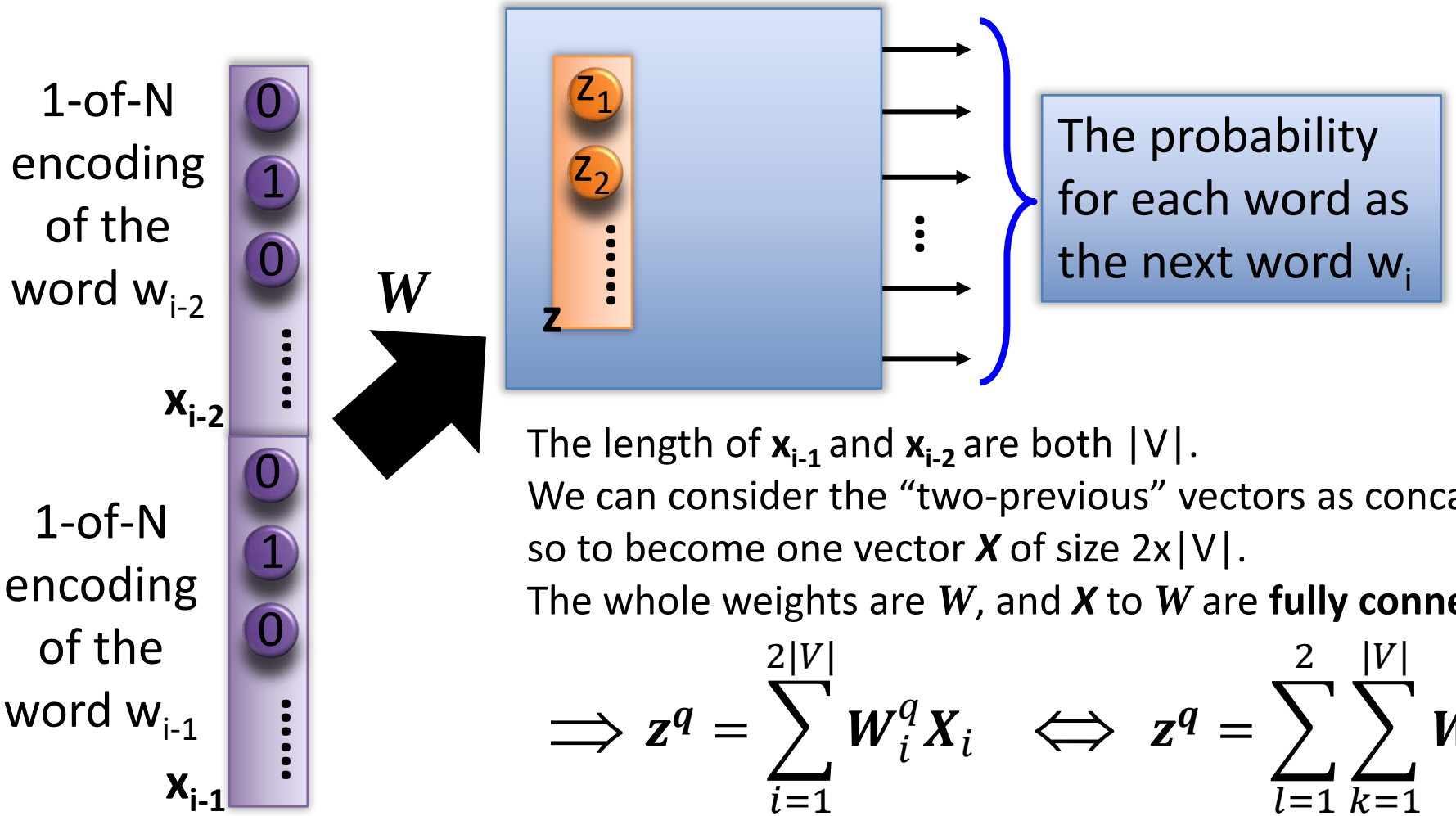


- Take out the **input** of the neurons in the first layer
- **Use it to represent a word**
- Word vector, word embedding feature: $V(\boldsymbol{w}) = \boldsymbol{z}$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Prediction-based – Sharing Parameters



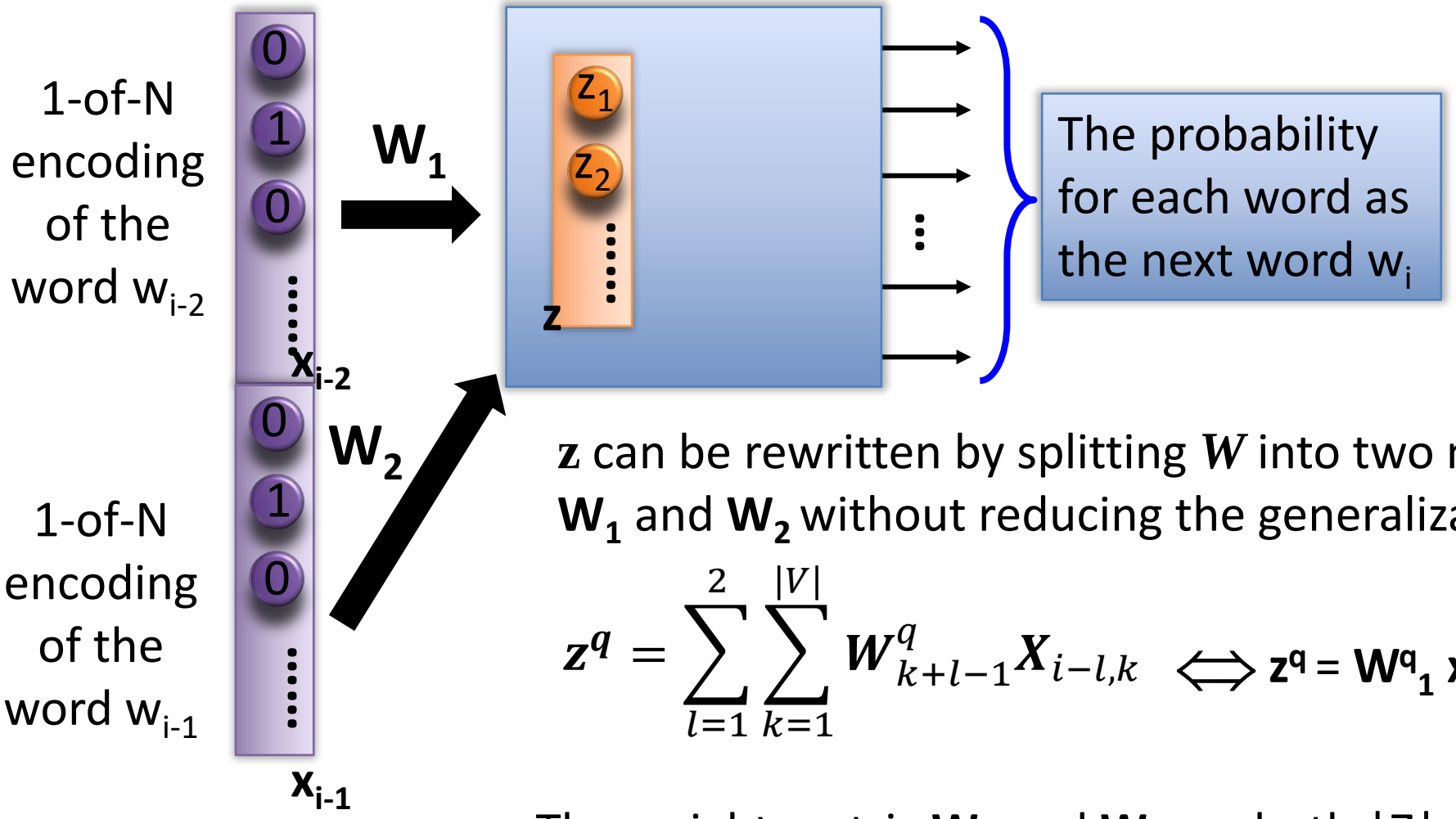
The length of $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i-2}$ are both $|V|$.
We can consider the "two-previous" vectors as concatenated and so to become one vector $\boldsymbol{X}$ of size $2x|V|$.
The whole weights are $\boldsymbol{W}$, and $\boldsymbol{X}$ to $\boldsymbol{W}$ are **fully connected**.

$$\Rightarrow \boldsymbol{z^q} = \sum_{i=1}^{2|V|} \boldsymbol{W_i^q X_i} \quad \Leftrightarrow \quad \boldsymbol{z^q} = \sum_{l=1}^{2} \sum_{k=1}^{|V|} \boldsymbol{W_{k+l-1}^q X_{i-l,k}}$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Prediction-based – Sharing Parameters

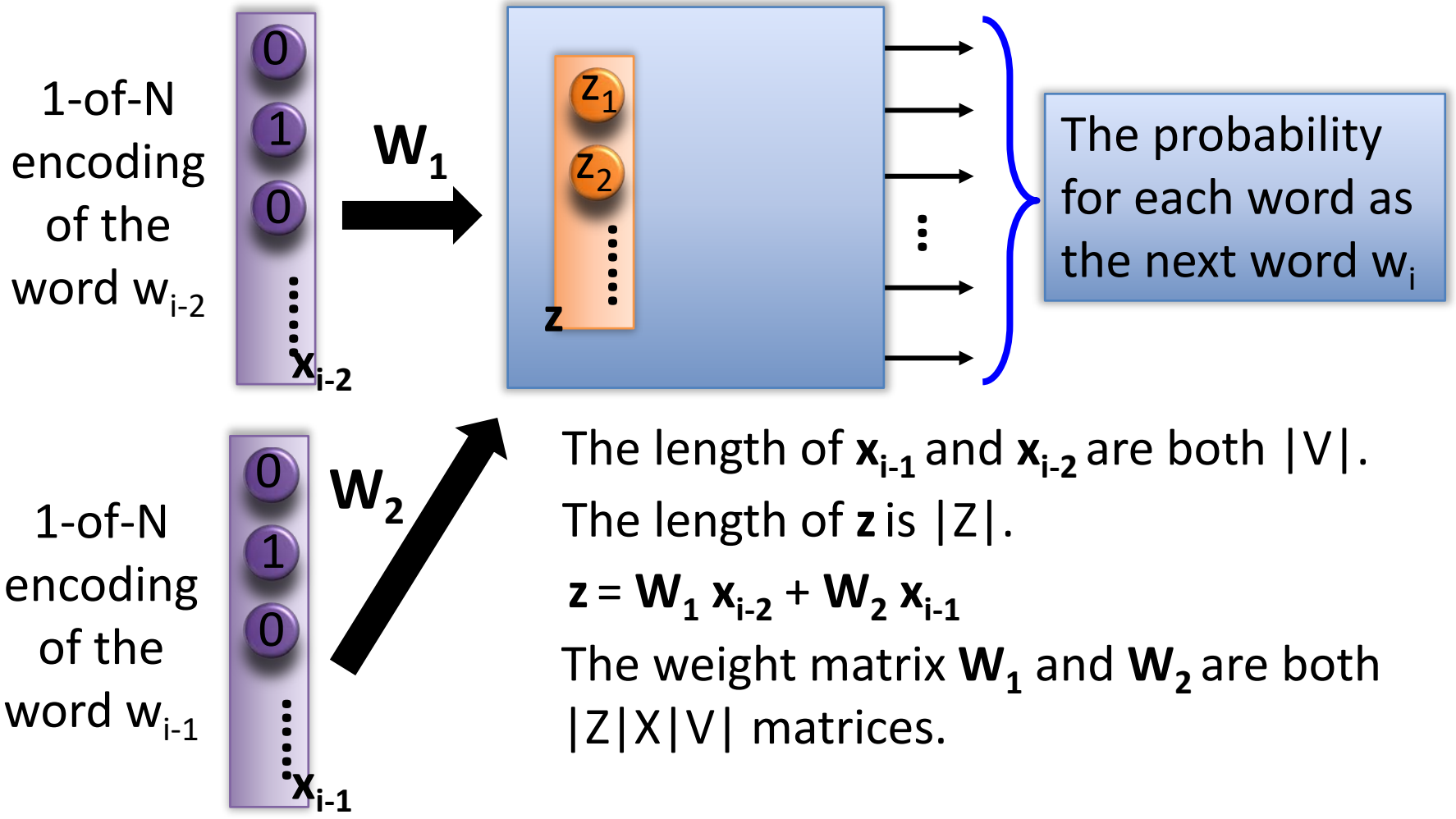


$\mathbf{z}$ can be rewritten by splitting W into two matrices $\mathbf{W_1}$ and $\mathbf{W_2}$ without reducing the generalizability:

$$z^q = \sum_{l=1}^{2} \sum_{k=1}^{|V|} W^q_{k+l-1} X_{i-l,k} \iff \mathbf{z^q = W^q_1\, x_{i-2} + W^q_2\, x_{i-1}}$$

The weight matrix $\mathbf{W_1}$ and $\mathbf{W_2}$ are both |Z|x|V|.

Prediction-based – Sharing Parameters



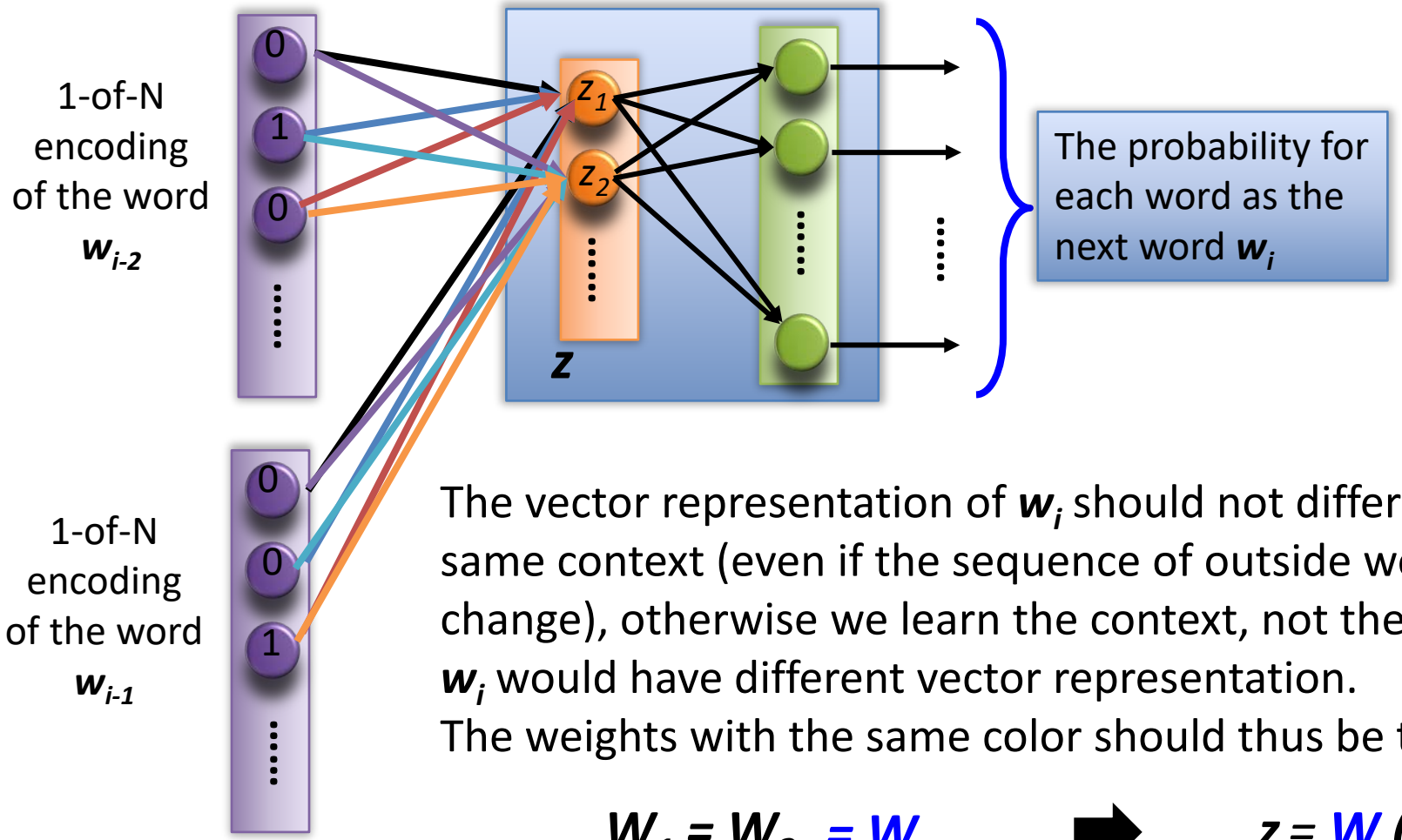
The length of $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i-2}$ are both $|V|$.

The length of $\mathbf{z}$ is $|Z|$.

$\mathbf{z} = \mathbf{W}_1 \mathbf{x}_{i-2} + \mathbf{W}_2 \mathbf{x}_{i-1}$

The weight matrix $\mathbf{W}_1$ and $\mathbf{W}_2$ are both $|Z|X|V|$ matrices.

Prediction-based – Sharing Parameters

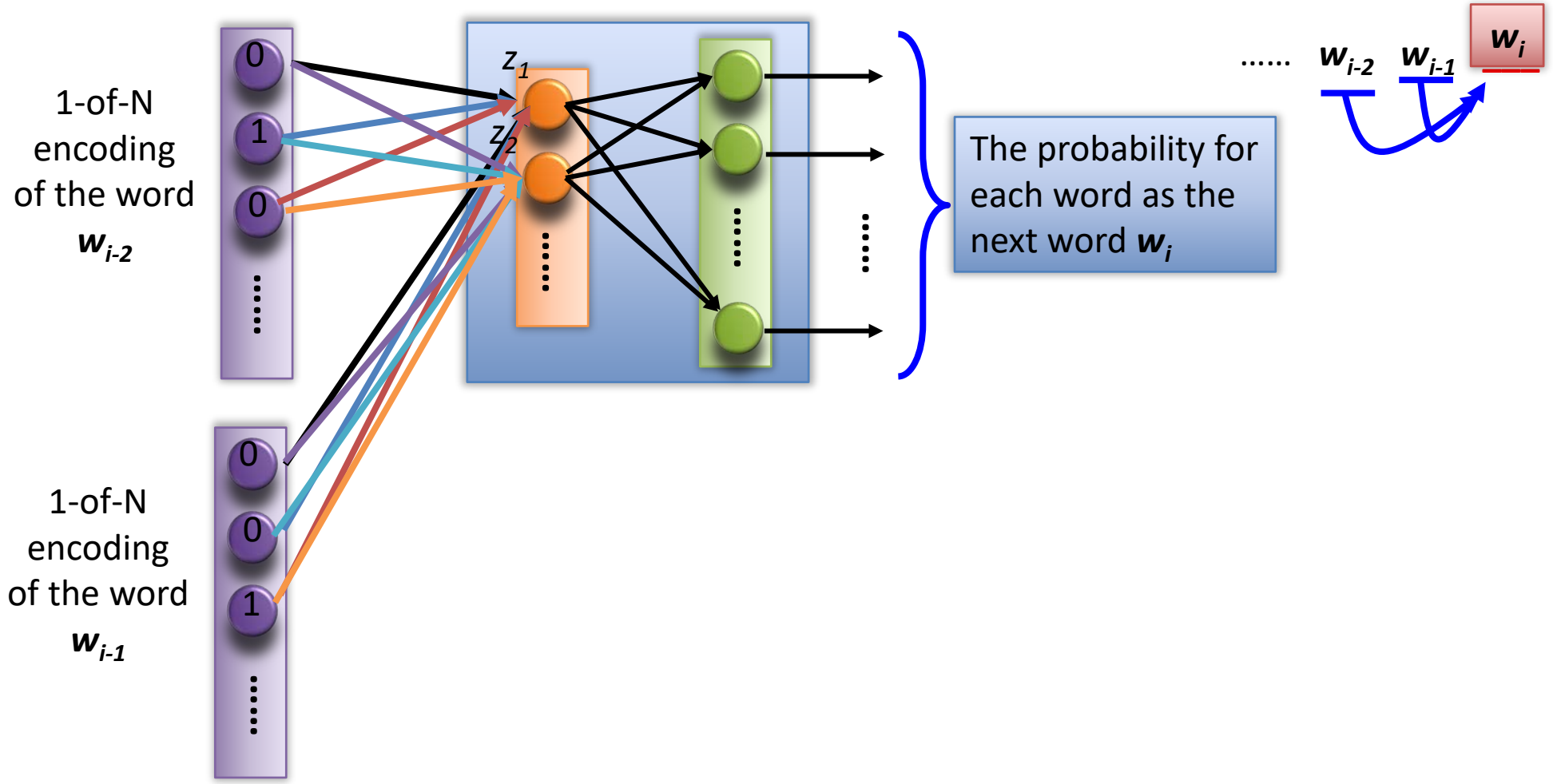


The vector representation of w_i should not differ from the same context (even if the sequence of outside words change), otherwise we learn the context, not the word w_i.
w_i would have different vector representation.
The weights with the same color should thus be the same.

$$W_1 = W_2 = W \quad \Rightarrow \quad z = W (x_{i-2} + x_{i-1})$$

It is the same idea as the shared connections in CNN, edge or corner extractors are actually defined by the weights

Prediction-based – Sharing Parameters



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Word2Vec: objective function

- We want to minimize the objective function, the (average) negative log likelihood :

$$J(\theta) = -\frac{1}{T}\log L(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\le j\le m \\ j\neq 0}} \log P\left(w_{t+j} \mid w_t; \theta\right)$$

Minimizing objective function ⟺ Maximizing predictive accuracy

- Question: How to calculate $P(w_{t+j}|w_t; \theta)$?
- Answer: We will *use two* vectors per word *w*:
  - V_w when *w* is a ***center*** word
  - u_w when *w* is a ***context*** word
- Then for a center word *c* and a context word *o*: $P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w\in V}\exp(u_w^T v_c)}$

Christopher Manning

Word2Vec: prediction function

Exponentiation makes anything positive

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

Dot product compares similarity of *o* and *c*.
$u^T v = u.v = \sum_{i=1}^{n} u_i v_i$
Larger dot product = larger probability

Normalize over entire vocabulary
to give probability distribution

- This is an example of the **softmax function** $\mathbb{R}^n \rightarrow \mathbb{R}^n$

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} = p_i$$

- The softmax function maps arbitrary values to a probability distribution p_i
  - "max" because amplifies probability of largest x_i
  - "soft" because still assigns some probability to smaller x_i

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Reminder
- **Multimodal correlations (self-attention)**
- Transformers
- Biases
- Computational cost, Energy, and Sovereinty

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Multimodal correlations

Self-Attention

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Self-attention

https://arxiv.org/abs/1706.03762

Attention is all you need.

$\boldsymbol{q}$: query (to match others)

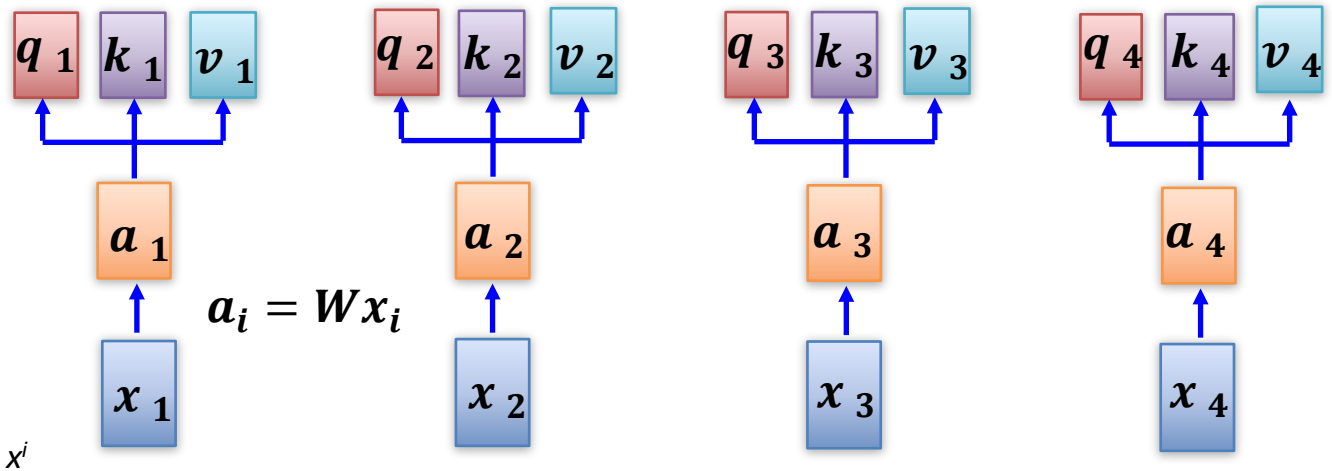
$$\boldsymbol{q_i} = \boldsymbol{W^q a_i}$$

$\boldsymbol{k}$: key (to be matched)

$$\boldsymbol{k_i} = \boldsymbol{W^k a_i}$$

$\boldsymbol{v}$: information to be extracted

$$\boldsymbol{v_i} = \boldsymbol{W^v a_i}$$



$\boldsymbol{a_i} = \boldsymbol{W x_i}$

a^i is the embedding of word x^i

Self-Attention

- Attention operates on **queries**, **keys**, and **values.**
  - We have some **queries** $q_1, q_2, \ldots, q_T$. Each query is $q_i \in \mathbb{R}^d$
  - We have some **keys** $k_1, k_2, \ldots, k_T$. Each key is $k_i \in \mathbb{R}^d$
  - We have some **values** $v_1, v_2, \ldots, v_T$. Each value is $v_i \in \mathbb{R}^d$

The number of queries can differ from the number of keys and values in practice.

- In **self-attention**, **the queries, keys, and values are drawn from the same source.**
  - **For example, if the output of the previous layer is $x_1, \ldots, x_T$, (one vec per word) we could let $v_i = k_i = q_i = x_i$ (that is, use the same vectors for all of them!)**

- The (dot product) self-attention operation is as follows:

$$e_{ij} = \boldsymbol{q_i^T k_j}$$

Compute **key-query** affinities

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_l \exp(e_{il})}$$

Compute attention weights from affinities (softmax)

$$\mathbf{output}_i = \sum_j \alpha_{ij} \boldsymbol{v_j}$$

Compute outputs as weighted sum of **values**

This is inspired from Word2Vec but the context is the whole sequence

UNIVERSITÉ CÔTE D'AZUR

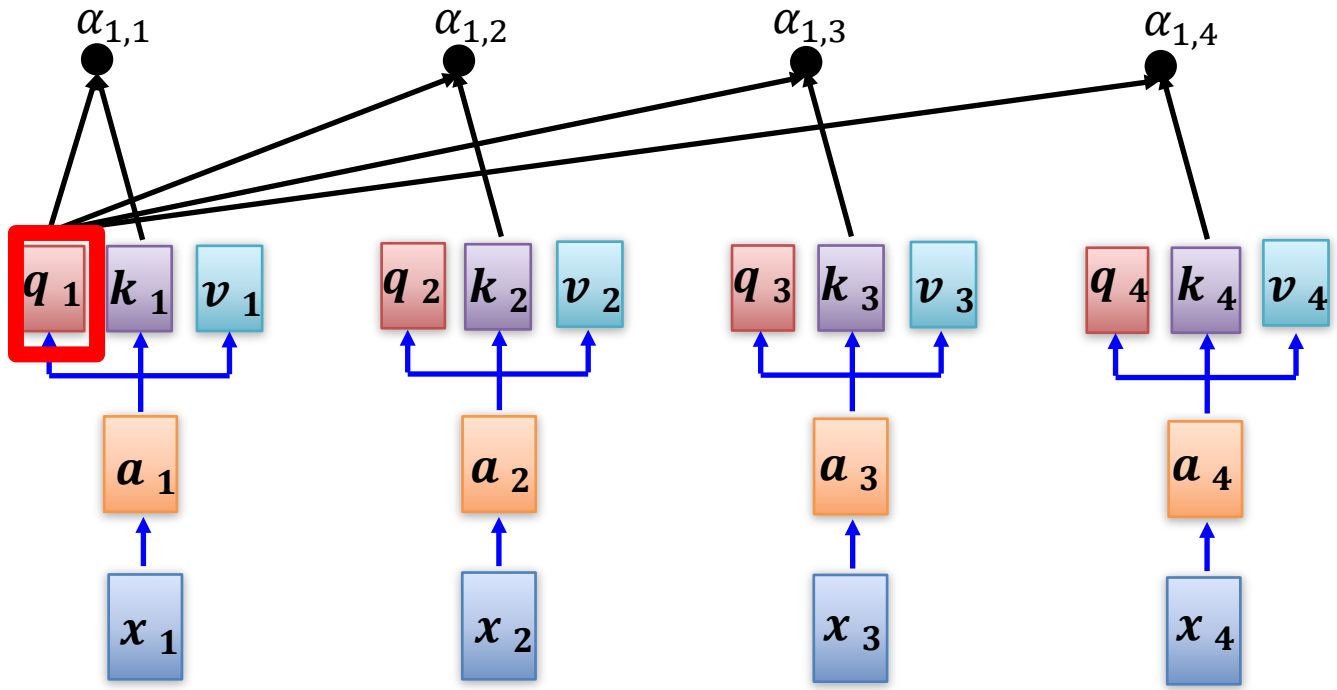
univ-cotedazur.fr

Self-attention

Take each query q to pay attention to each key k

d is the dim of $\boldsymbol{q}$ and $\boldsymbol{k}$

Scaled Dot-Product Attention: $\alpha_{1,i} = \underbrace{\boldsymbol{q_1} \cdot \boldsymbol{k_i}}_{\text{dot product}} / \sqrt{d}$

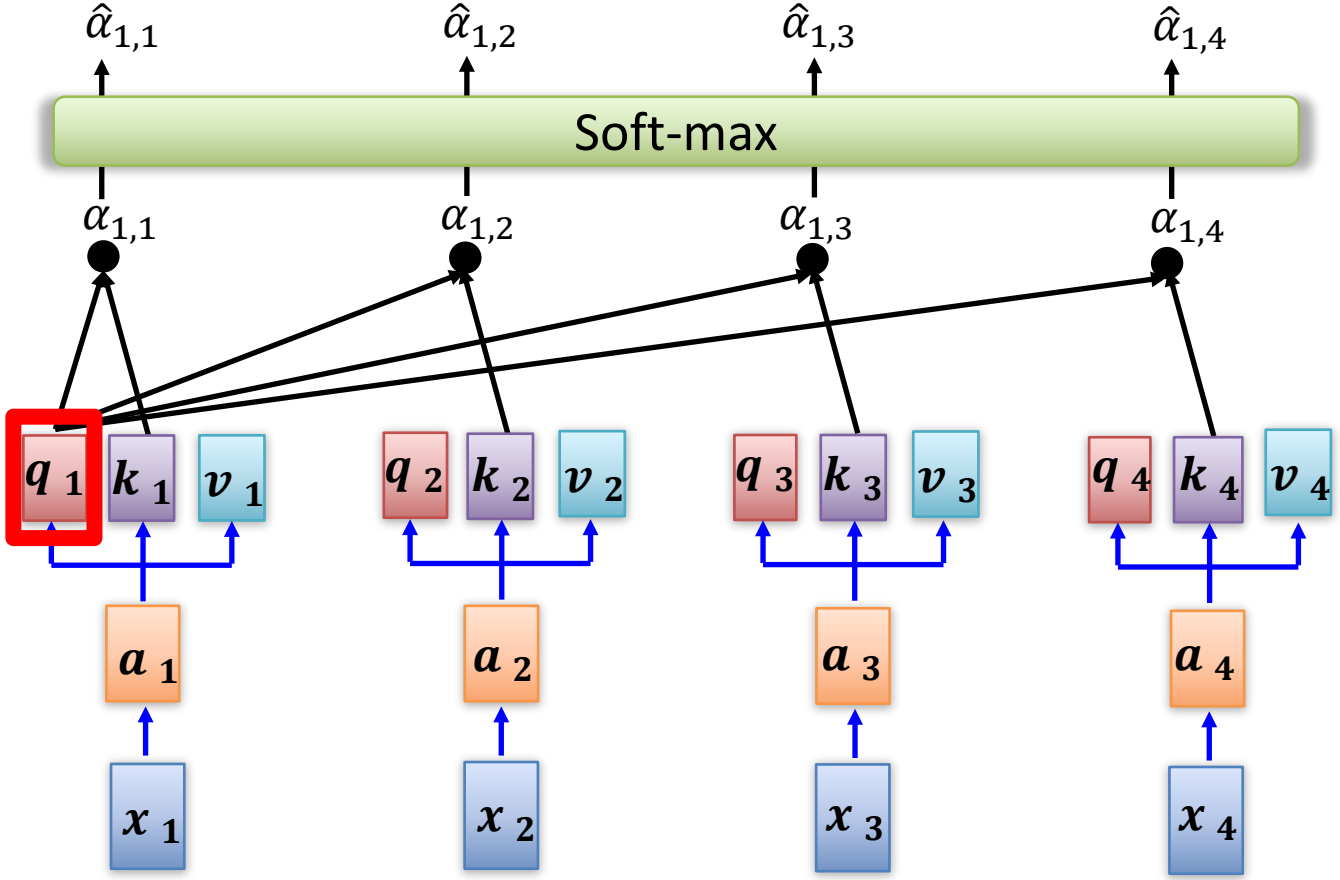


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Self-attention

$$\hat{\alpha}_{1,i} = exp(\alpha_{1,i}) / \sum_j exp(\alpha_{1,j})$$



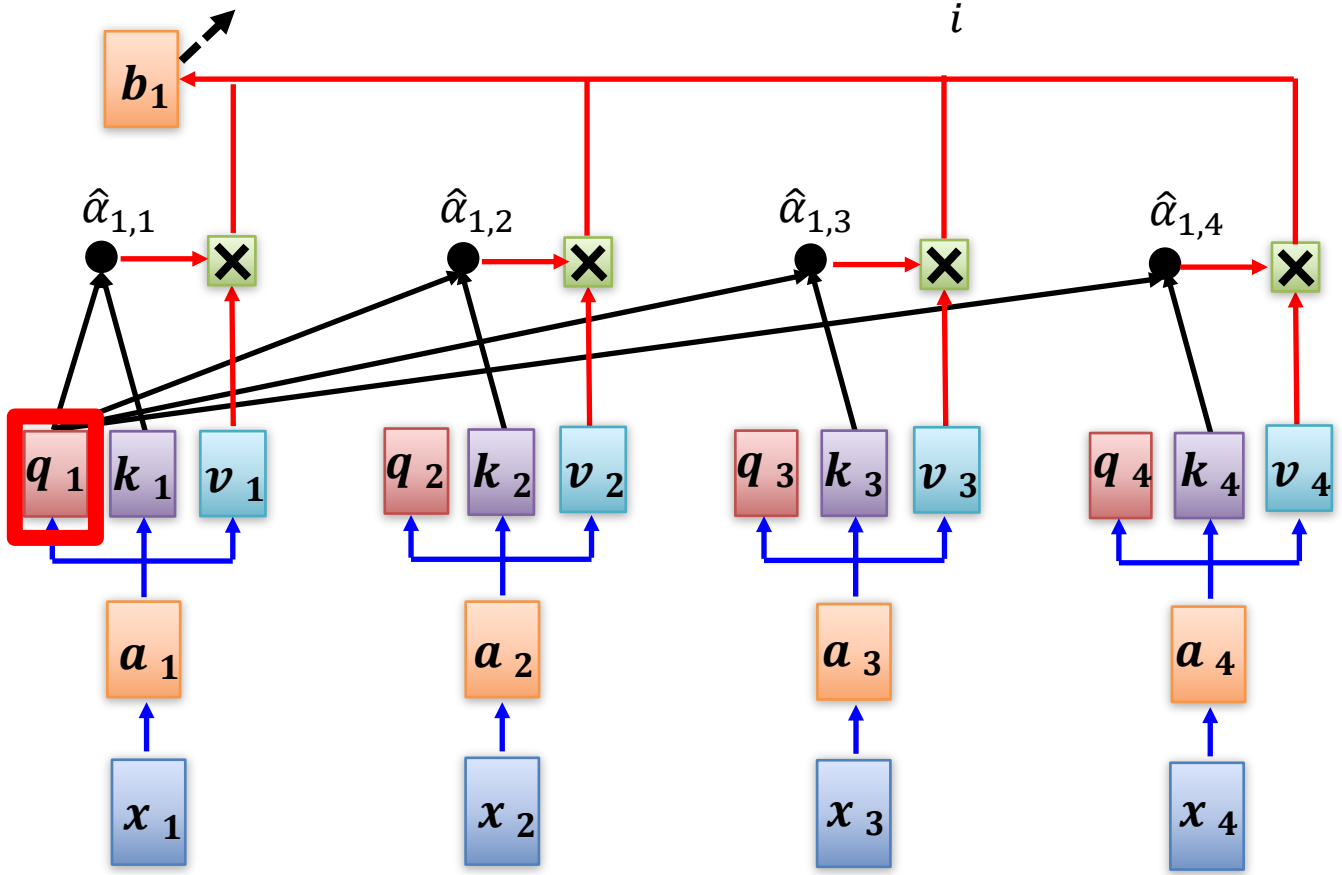
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Self-attention

Considering the whole sequence

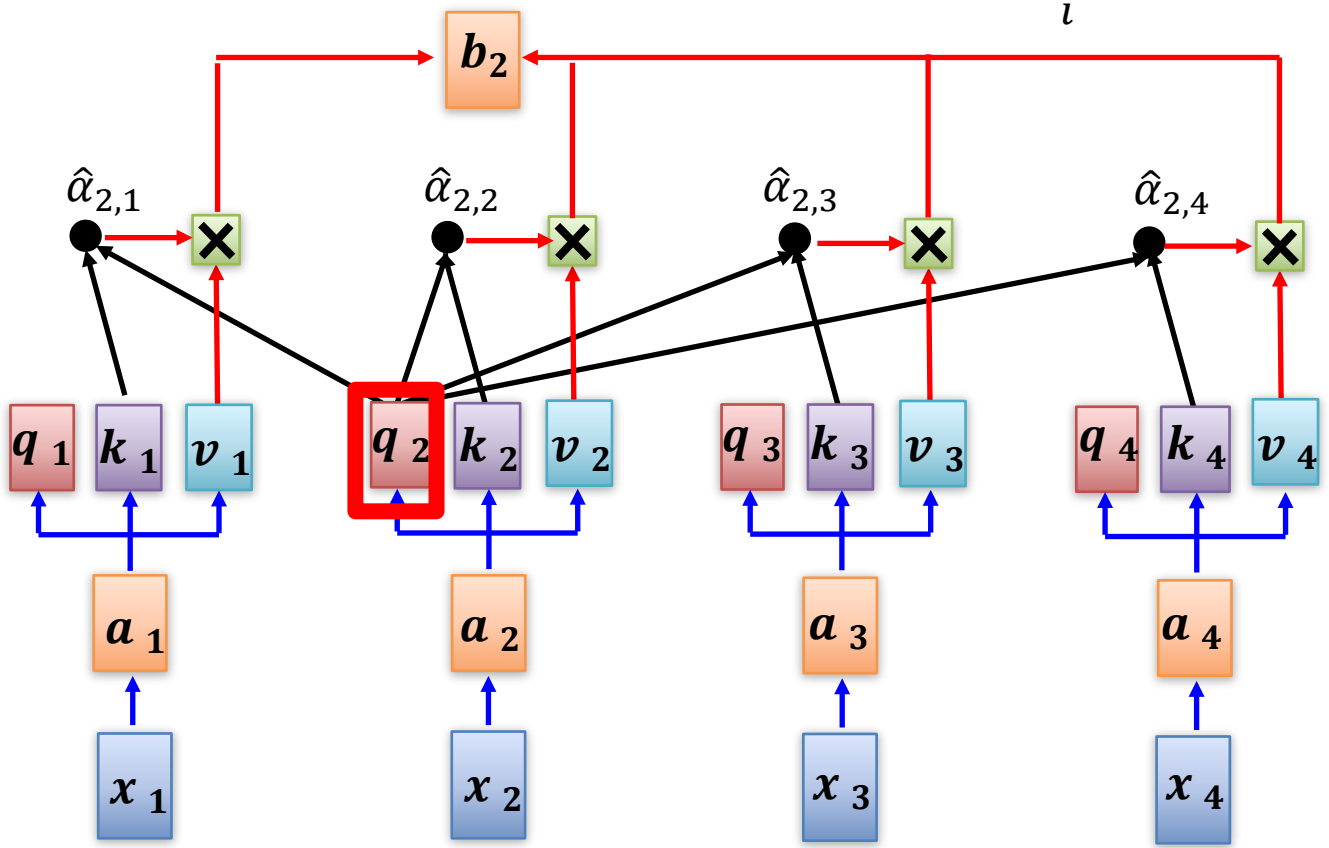
$$b_1 = \sum_i \hat{\alpha}_{1,i} v_i$$



Self-attention

Take each query q to pay attention to each key k

$$b_2 = \sum_i \hat{\alpha}_{2,i} v_i$$

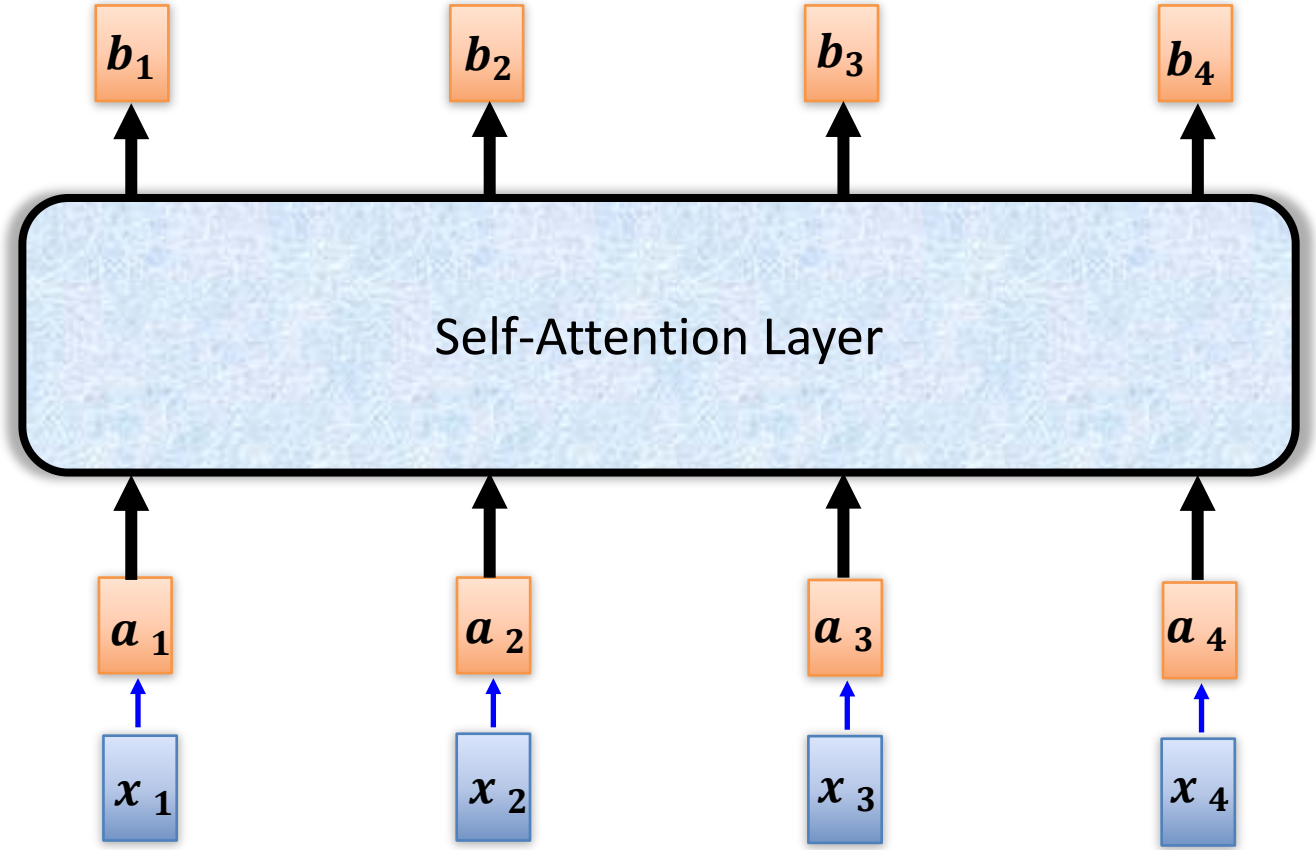


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

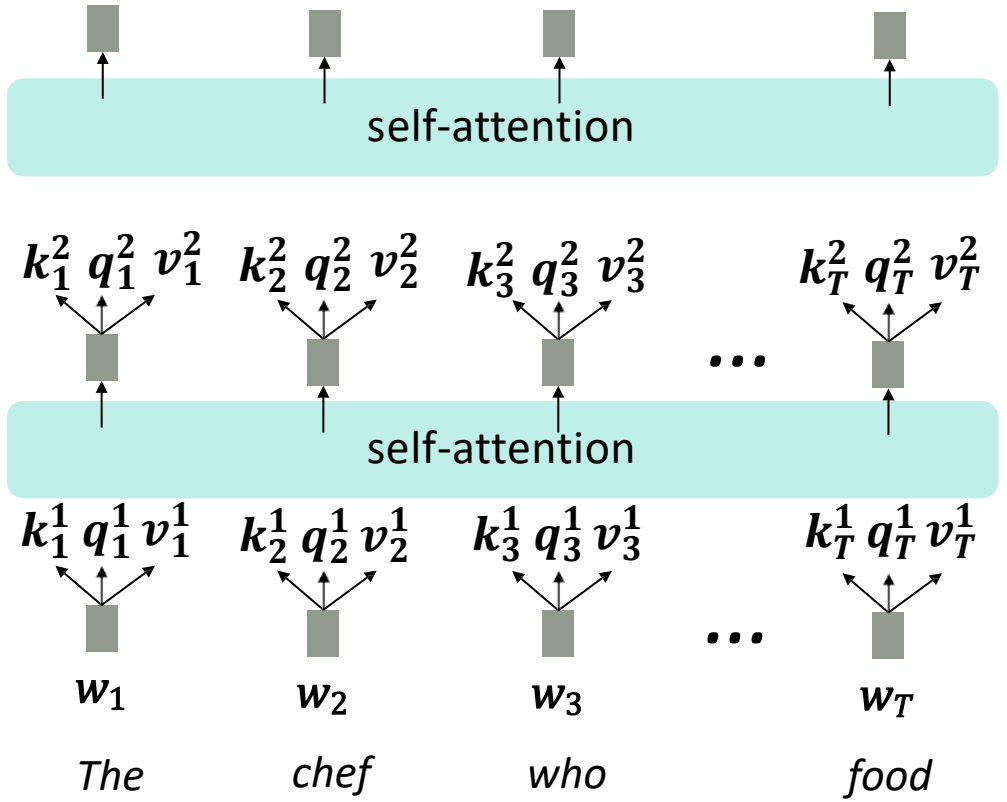
Self-attention

b_1 , b_2 , b_3 , b_4 can be parallelly computed.



Self-Attention

- In the diagram at the right, we have stacked self-attention blocks, like we might stack LSTM layers.
- Can self-attention be a drop-in replacement for recurrence?
- No. It has a few issues, which we'll go through.
- First, self-attention is an operation on **sets**. It has no inherent notion of order.



Self-attention doesn't know the order of its inputs.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Barriers and solutions for Self-Attention as a building block

Barriers

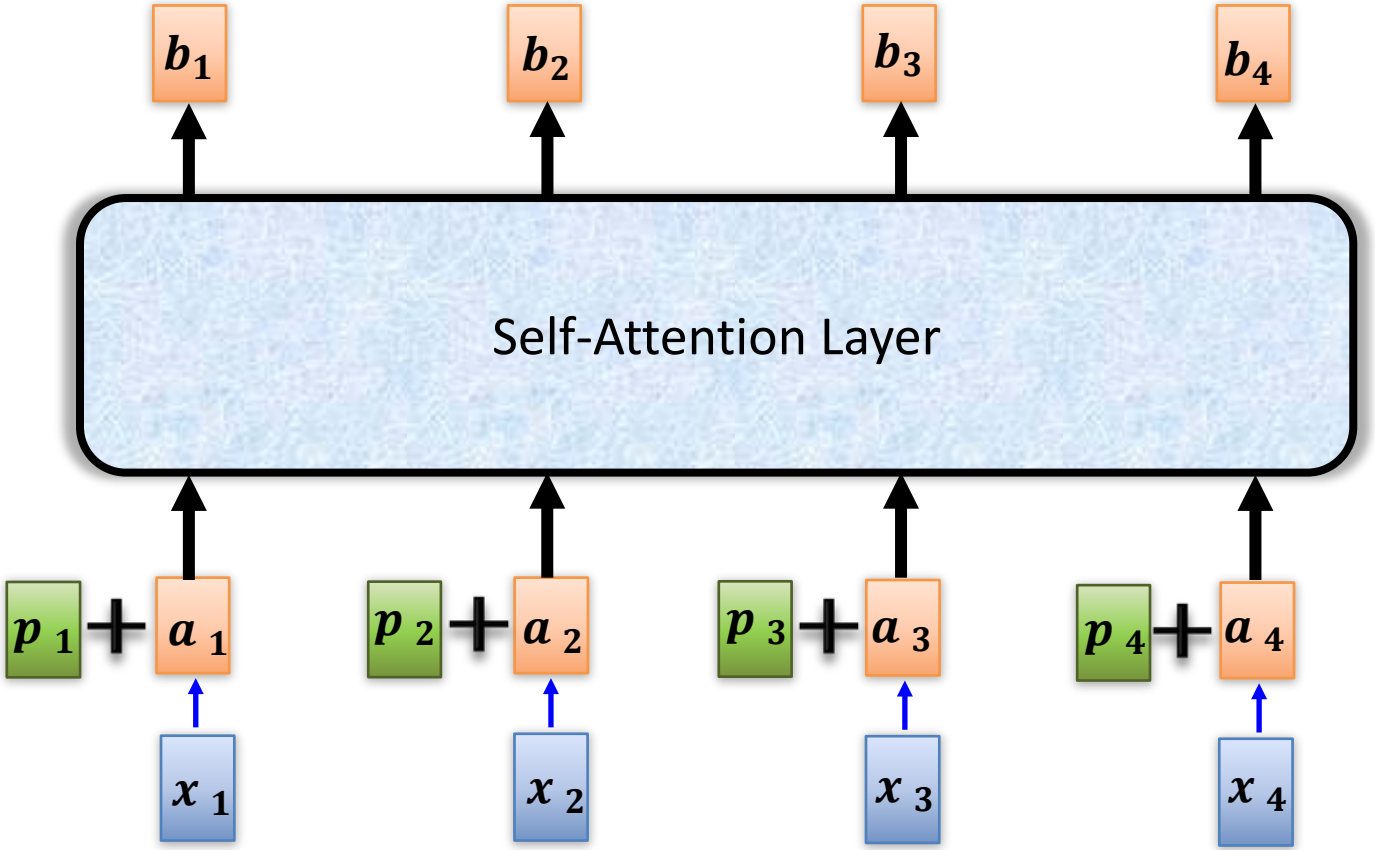
- Doesn't have an inherent notion of order!

Solutions

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Fixing the first self-attention problem: **Sequence order by adding position**



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Fixing the first self-attention problem: **Sequence order**

- Since self-attention doesn't build in order information, we need to encode the order of the sentence in our keys, queries, and values.
- Consider representing each **sequence index** as a **vector**

$\boldsymbol{p_i} \in \mathbb{R}^d$, for $i \in \{1,2,\ldots,T\}$ are position vectors

- Don't worry about what the $\boldsymbol{p_i}$ are made of yet!
- Easy to incorporate this info into our self-attention block: just add the p_i to our inputs!
- Let $\boldsymbol{v_i'}, \boldsymbol{k_i'}, \boldsymbol{q_i'}$ be our old values, keys, and queries.

$$\boldsymbol{v_i} = \boldsymbol{v_i'} + \boldsymbol{p_i}$$
$$\boldsymbol{q_i} = \boldsymbol{q_i'} + \boldsymbol{p_i}$$
$$\boldsymbol{k_i} = \boldsymbol{k_i'} + \boldsymbol{p_i}$$

In deep self-attention networks, we do this at the first layer! You could concatenate them as well, but people mostly just add…

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Barriers and solutions for Self-Attention as a building block

Barriers

- Doesn't have an inherent notion of order!
- No nonlinearities for deep learning! It's all just weighted averages

Solutions

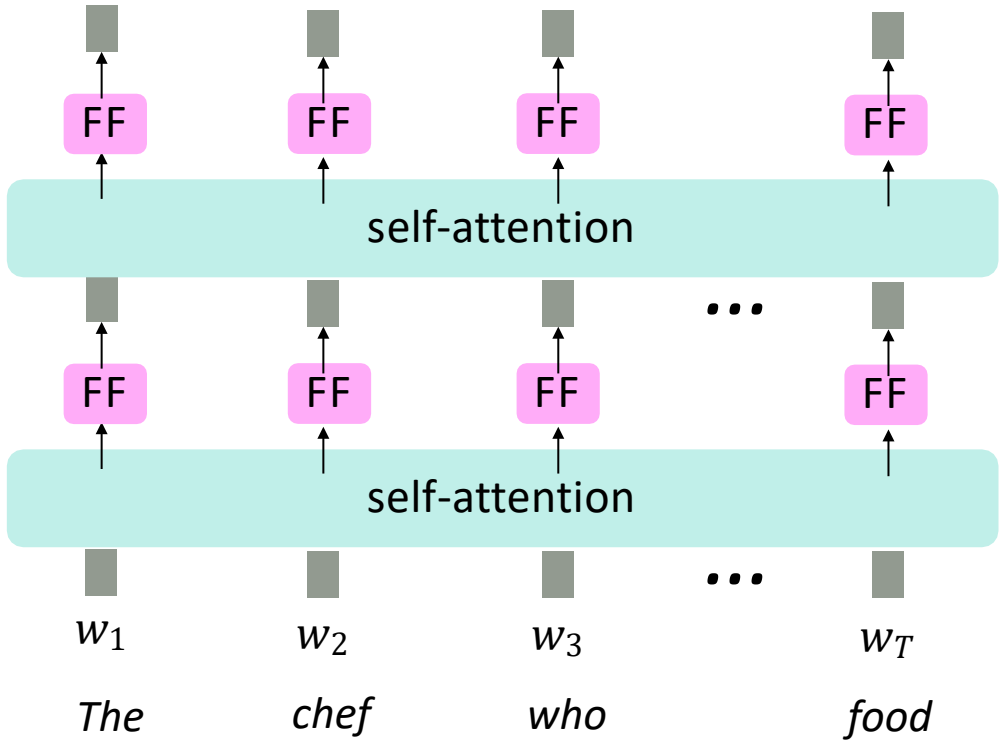
- Add position representations to the inputs

Adding nonlinearities in self-attention

- Note that there are no elementwise nonlinearities in self-attention; stacking more self-attention layers just re-averages **value** vectors

- Easy fix: add a **feed-forward network** to post-process each output vector.

$$m_i = MLP(\text{output}_i)$$
$$= W_2 * \text{ReLU}(W_1 \times \text{output}_i + b_1) + b_2$$

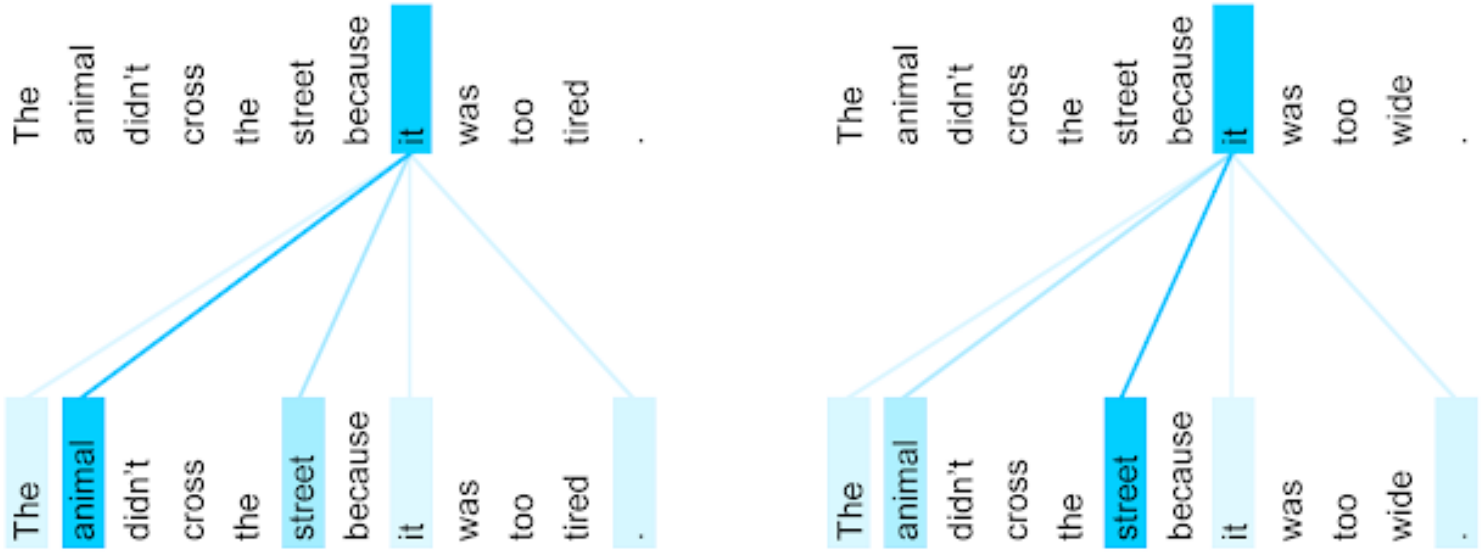


Intuition: the FF network processes the result of attention

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Attention Visualization



The encoder self-attention distribution for the word “it” from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Necessities for a self-attention building block:

- **Self-attention**:
  - the basis of the method.
- **Position representations**:
  - Specify the sequence order, since self-attention is an unordered function of its inputs.
- **Nonlinearities**:
  - At the output of the self-attention block
  - Frequently implemented as a simple feed-forward network.
- **Masking**:
  - In order to parallelize operations while not looking at the future.
  - Keeps information about the future from "leaking" to the past.

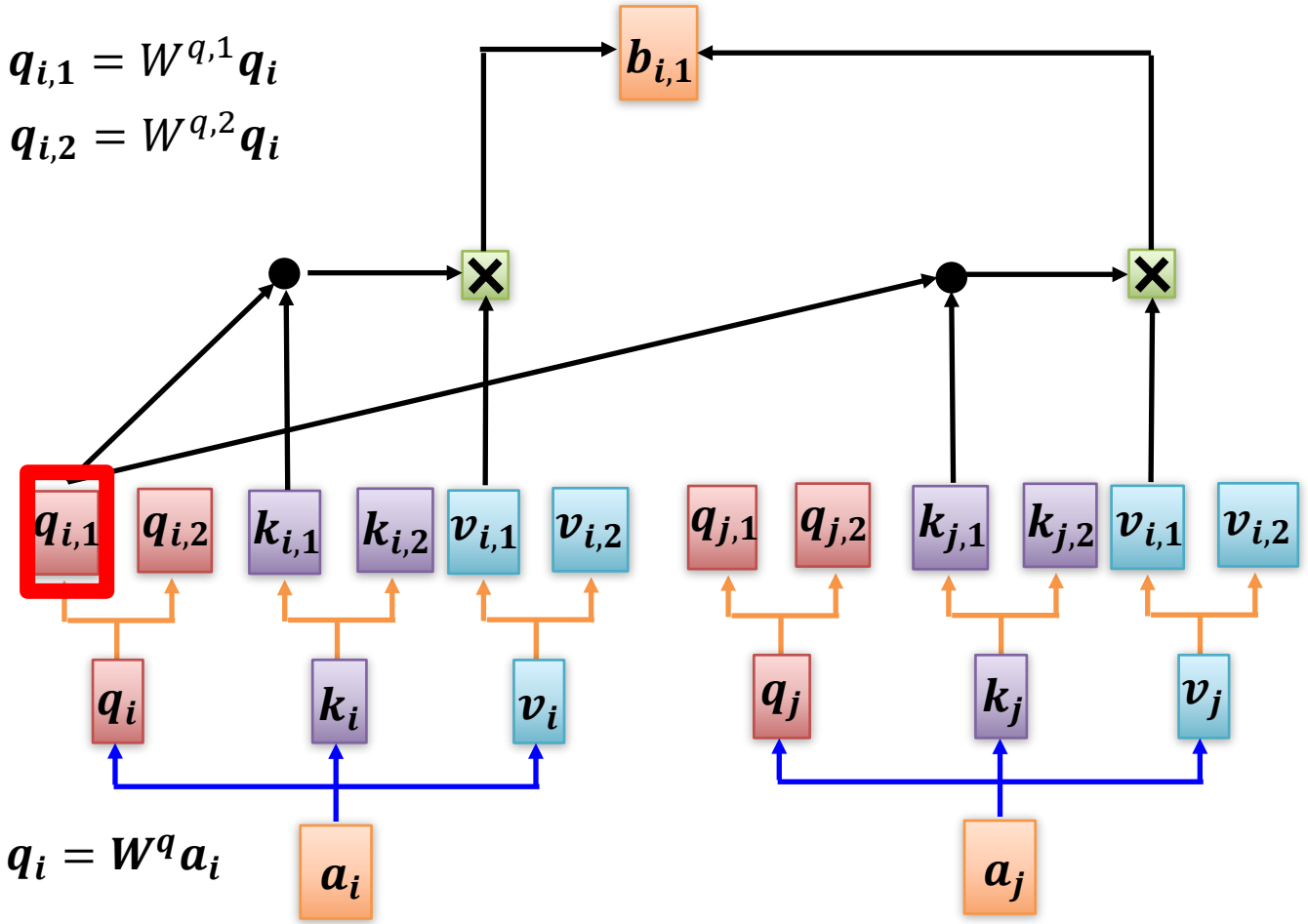
- These are the **main steps to build Transformers**…But this is not yet the **Transformer** model we've been hearing about!

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

The Transformer Encoder:
Multi-headed attention

(2 heads as example)



UNIVERSITÉ CÔTE D'AZUR

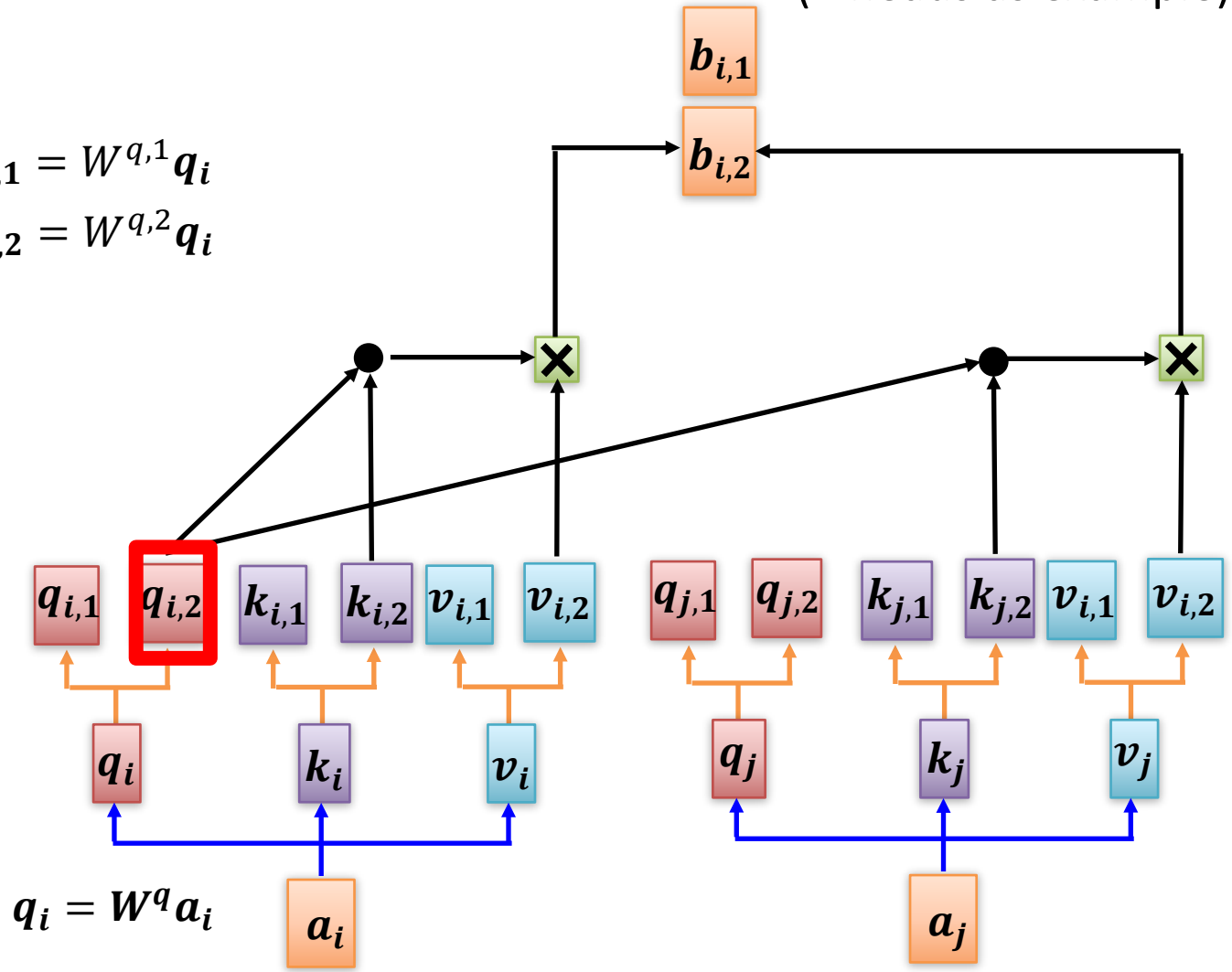
univ-cotedazur.fr

The Transformer Encoder: **Multi-headed attention**

(2 heads as example)

$$q_{i,1} = W^{q,1} q_i$$

$$q_{i,2} = W^{q,2} q_i$$



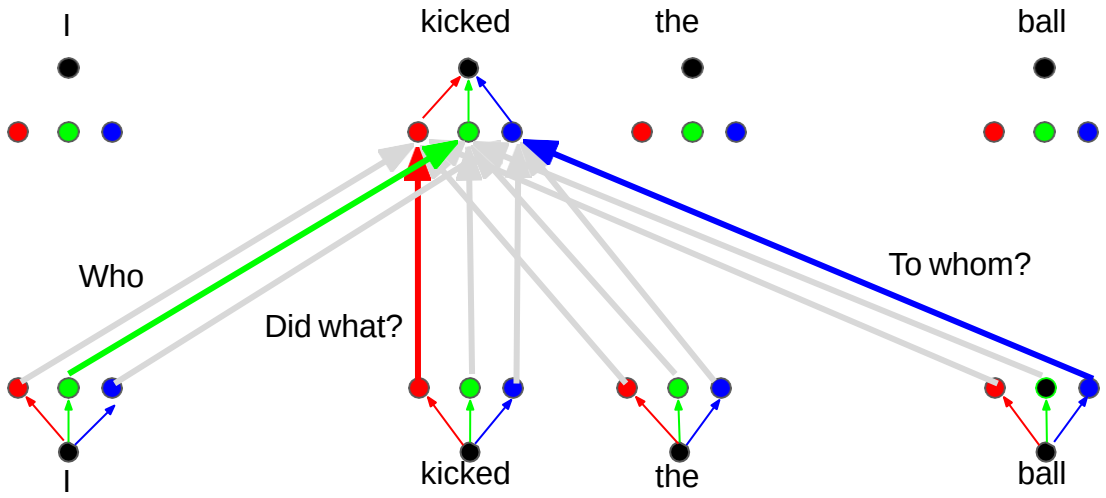
$$q_i = W^q a_i$$

The principle of multi-head attention in transformers is similar to the one of multi-kernels in one convolution layer of a CNN (i.e. multi-feature maps)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Parallel attention heads



Ashish Vaswani

The Transformer Encoder: **Multi-headed attention**

- We make the patterns we are looking for even more flexible: they can depend on neighbouring words or pixels!

(Multi-head self attention) allows for large-window, data-specific patterns/kernels

$$\mathbf{e}_i^{1,h} = \sum_j \text{softmax}\left(\frac{\mathbf{q}_{i,h} \cdot \mathbf{k}_{j,h}}{\sqrt{d}}\right)\mathbf{v}_{j,h}$$

$$\frac{e^{\mathbf{q}_{i,h}\cdot\mathbf{k}_{j,h}}}{\sum_l e^{\mathbf{q}_{l,h}\cdot\mathbf{k}_{l,h}}}$$

But heavy in calculations in tests too, not only in training as before!

$$\mathbf{q}_{i,h} = \mathbf{W}^{q,h}\mathbf{e}_i^0 \ , \quad \mathbf{k}_{j,h} = \mathbf{W}^{k,h}\mathbf{e}_j^0 \ , \quad \mathbf{v}_{j,h} = \mathbf{W}^{v,h}\mathbf{e}_j^0$$

$$\mathbf{e}_i^1 = MLP\left(\text{Linear}\left(\mathbf{e}_i^{1,1}, \ldots, \mathbf{e}_i^{1,H}\right)\right)$$

- The word ***i*** is represented by a recombination of (various representations of) its neighboring words, the factors of which themselves vary according to neighboring words (not as before).

A. Vaswani et al.. Attention is all you need. NeurIPS 2017.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Reminder
- Multimodal correlations (self-attention)
- **Transformers**
- Biases
- Computational cost, Energy, and Sovereinty

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

TRANSFORMERS

UNIVERSITÉ CÔTE D'AZUR

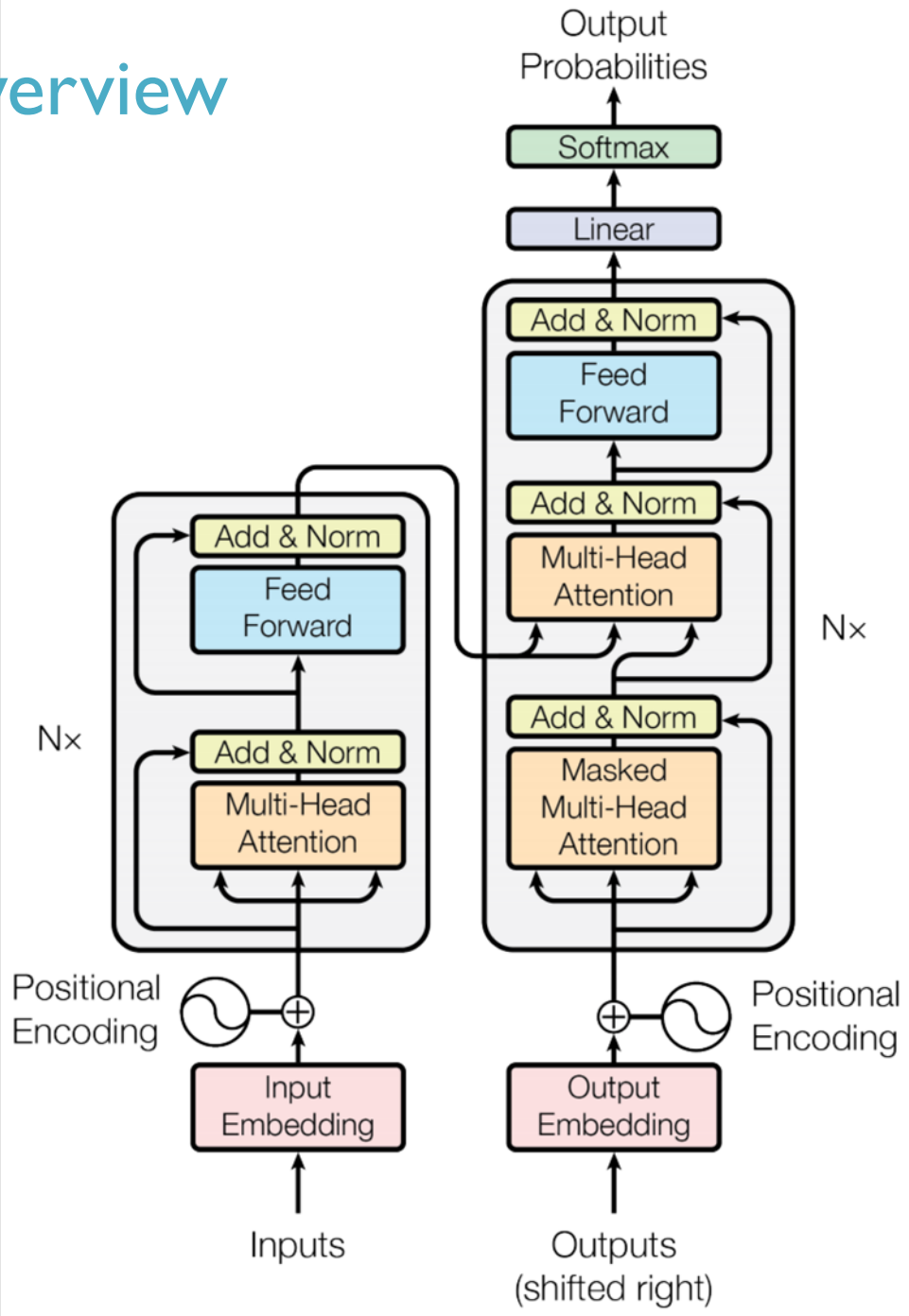
univ-cotedazur.fr

Transformer Overview

Attention is all you need. 2017. Aswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin
https://arxiv.org/pdf/1706.03762.pdf

- **Non-recurrent sequence-to-sequence encoder-decoder model**
- **Task: machine translation with parallel corpus**
- Predict each translated word
- Final cost/error function is standard cross-entropy error on top of a softmax classifier

This and related figures from paper ⇑

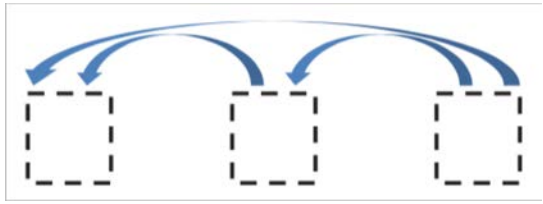


UNIVERSITÉ CÔTE D'AZUR

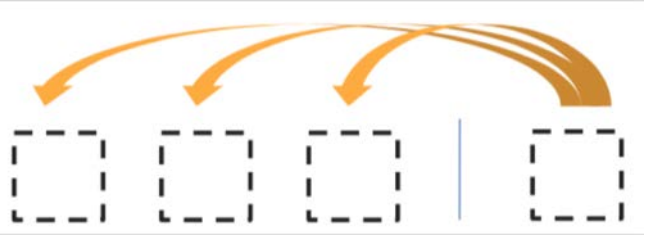
univ-cotedazur.fr

Transformer Decoder

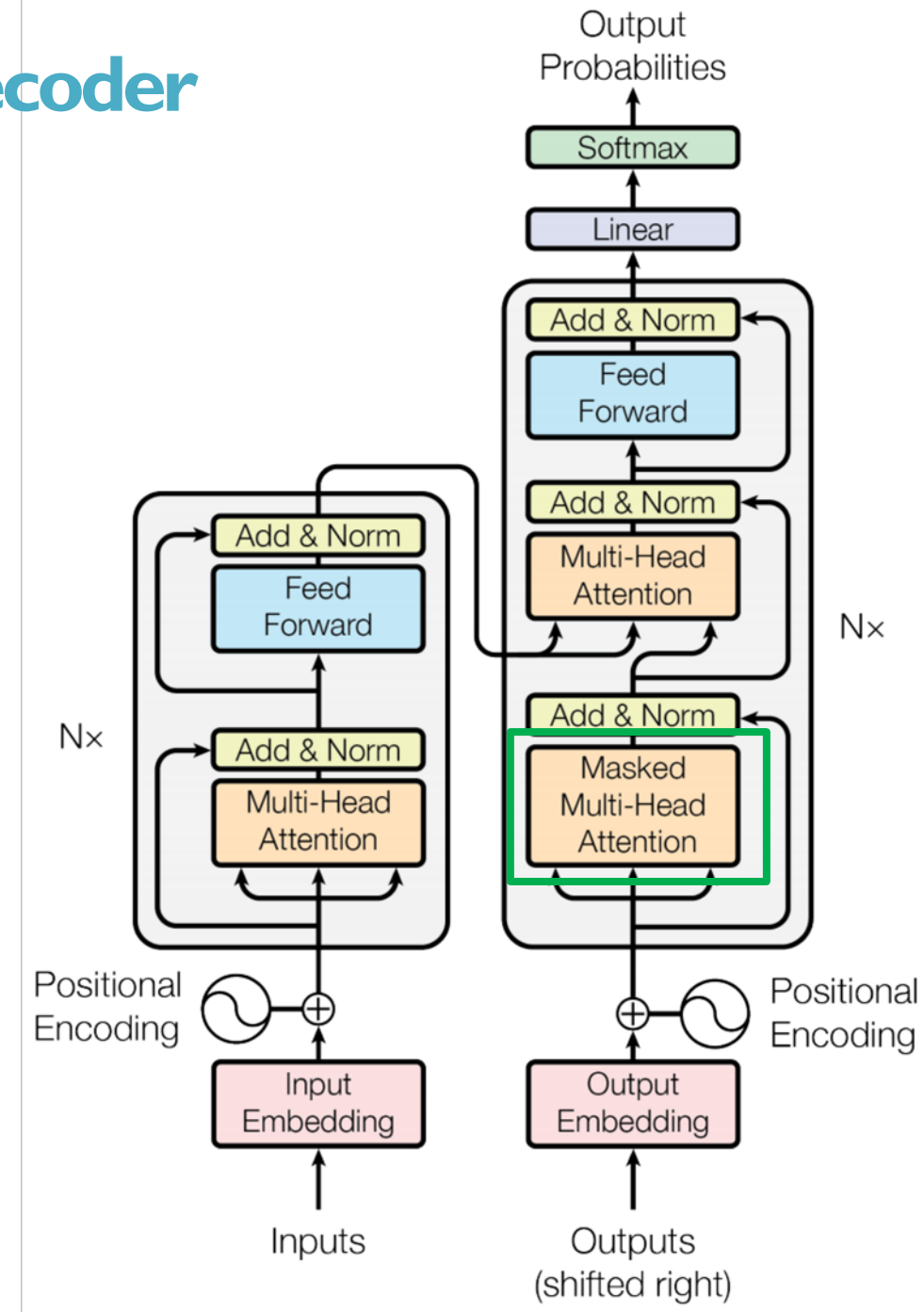
- 2 sublayer changes in decoder
- Masked decoder self-attention on previously generated outputs:



- Encoder-Decoder Attention, where queries come from previous decoder layer and keys and values come from output of encoder, **Cross-Attention**



Blocks repeated 6 times also

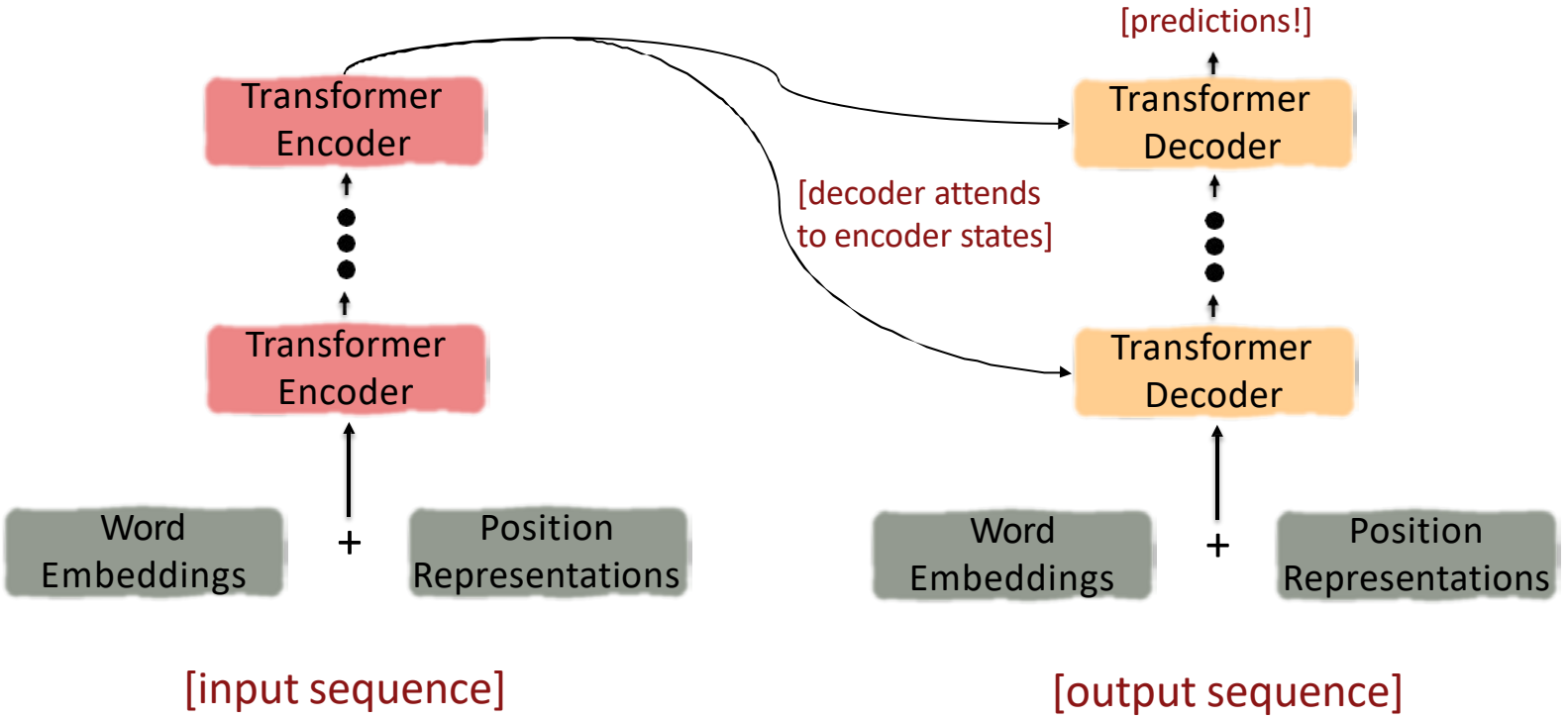


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

The Transformer Encoder-Decoder [Vaswani et al., 2017]

Looking back at the whole model, zooming in on an Encoder block:

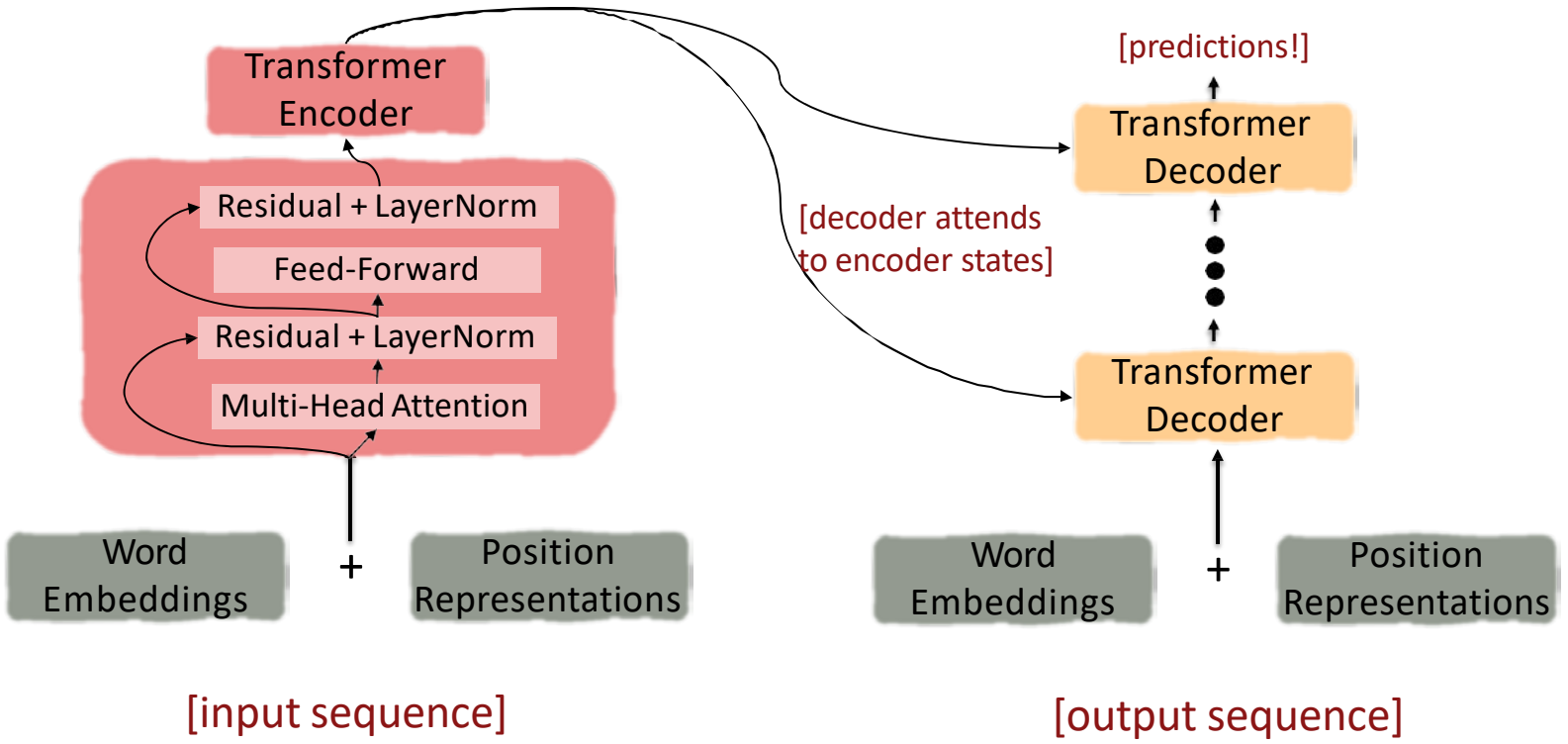


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

The Transformer Encoder-Decoder [Vaswani et al., 2017]

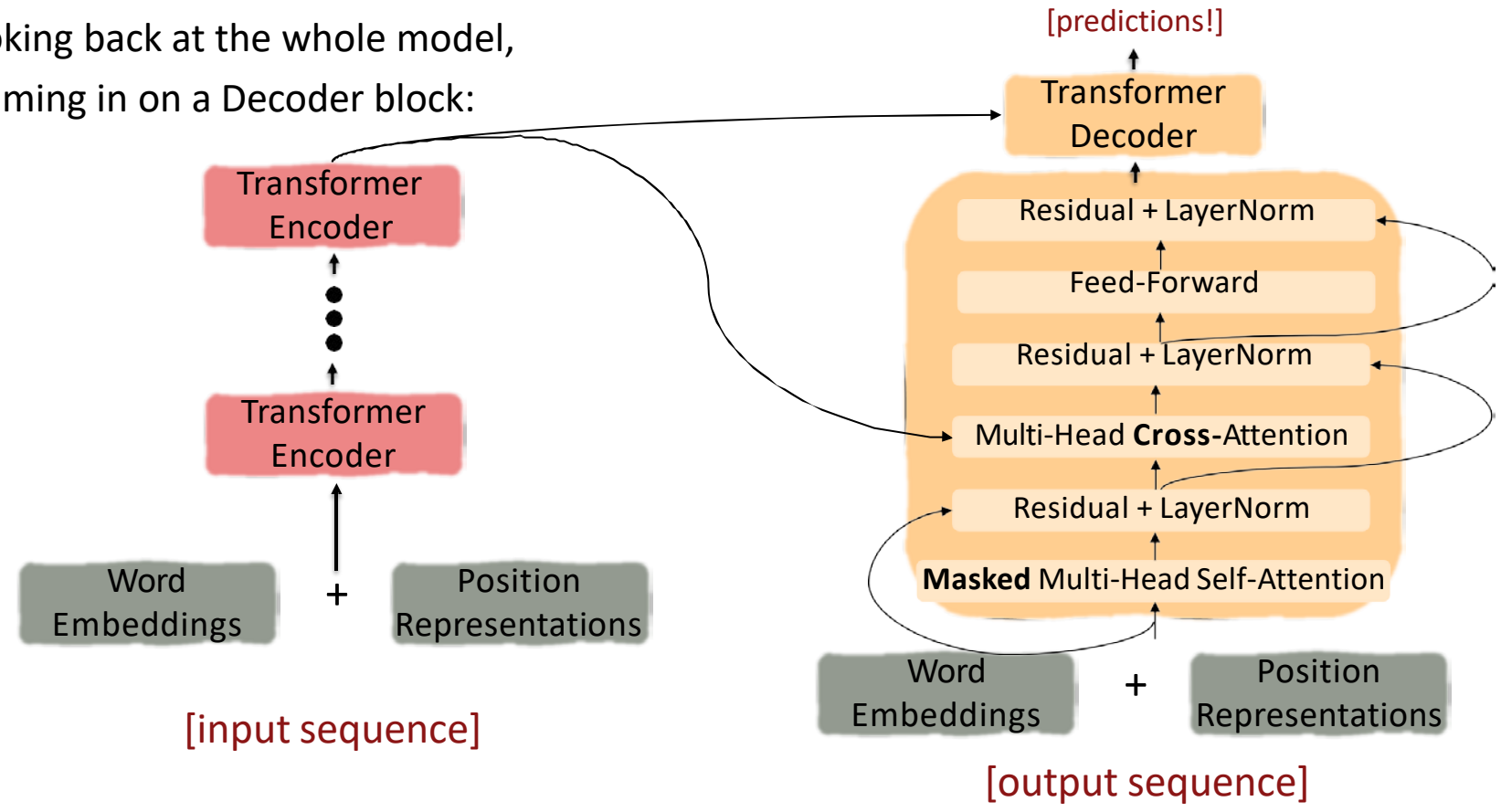
Looking back at the whole model, zooming in on an Encoder block:



The Transformer Encoder-Decoder

[Vaswani et al., 2017]

Looking back at the whole model, zooming in on a Decoder block:



UNIVERSITÉ CÔTE D'AZUR

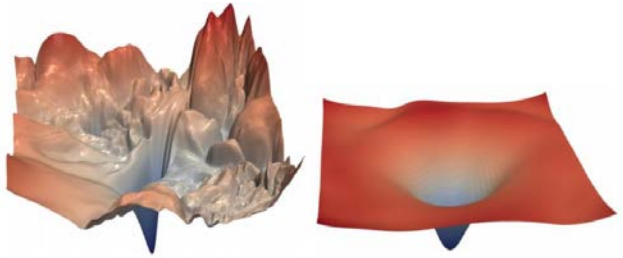
univ-cotedazur.fr

The Transformer Encoder-Decoder [Vaswani et al., 2017]

Next, let's look at the Transformer Encoder and Decoder Blocks

What's left in a Transformer Encoder Block that we haven't covered?

1. **Key-query-value attention:** How do we get the k, q, v vectors from a single word embedding?
2. **Multi-headed attention**: Attend to multiple places in a single layer!
3. **Tricks to help with training!**
   1. Residual connections
   2. Layer normalization
   3. Scaling the dot product
   4. These tricks **don't improve** what the model is able to do; **they help improve the training process**



[no residuals] [residuals]

[Loss landscape visualization, Li et al., 2018, on a ResNet]

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Reminder
- Multimodal correlations (self-attention)
- Transformers
  - **Transformers: The way you pre-train the encoder and the decoder different architectures**
  - Transformers beyond Natural Language Processing
- Biases
- Computational cost, Energy, and Sovereinty

UNIVERSITÉ
CÔTE D'AZUR

Transformers: The way you pre-train the encoder and the decoder → different architectures

(BERT, SpanBert, GPT, GPT-2, GPT-3, etc)

Barriers and solutions for Self-Attention as a building block

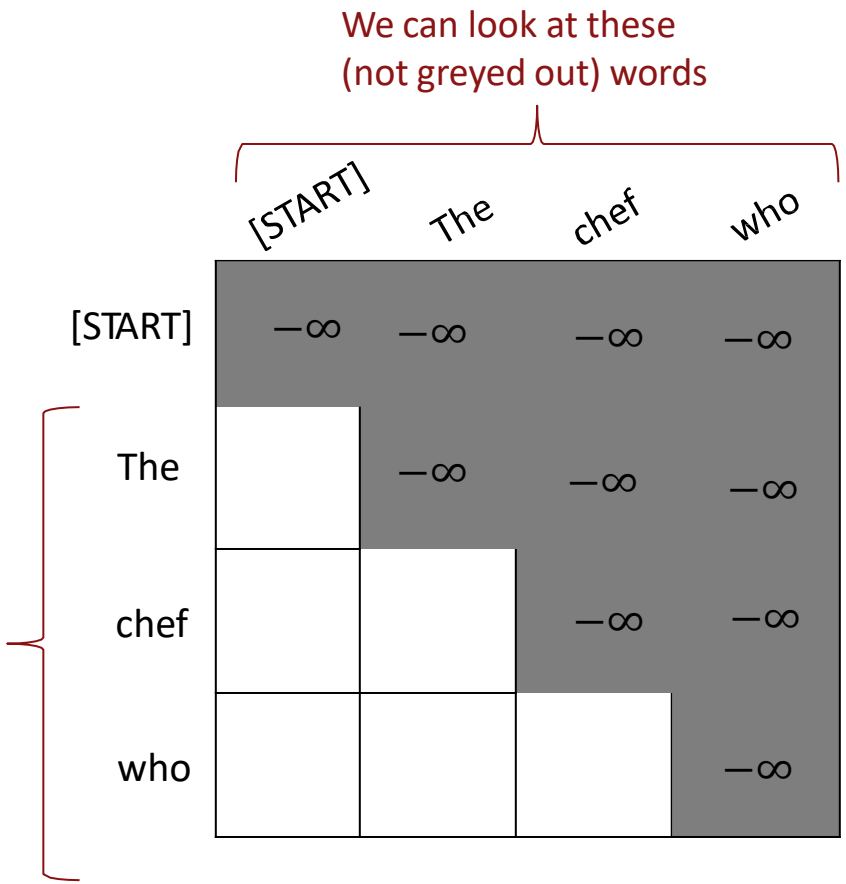
Barriers	Solutions
• Doesn't have an inherent notion of order!	• Add position representations to the inputs
• No nonlinearities for deep learning magic! It's all just weighted averages	• Easy fix: apply the same feedforward network to each self-attention output.

• Need to ensure we don't "look at the future" when predicting a sequence
  • Like in machine translation
  • Or language modeling	

Masking the future in self-attention

- To use self-attention in **decoders**, we need to ensure we can't peek at the future.
- At every timestep, we could change the set of **keys and queries** to include only past words. (Inefficient!)
- To enable parallelization, we **mask out attention** to future words by setting attention scores to $-\infty$.

$$e_{ij} = \begin{cases} q_i^\top k_j, j < i \\ -\infty, j \geq i \end{cases}$$



Barriers and solutions for Self-Attention as a building block

Barriers

- Doesn't have an inherent notion of order!
- No nonlinearities for deep learning magic! It's all just weighted averages
- Need to ensure we don't "look at the future" when predicting a sequence
  - Like in machine translation
  - Or language modeling

Solutions

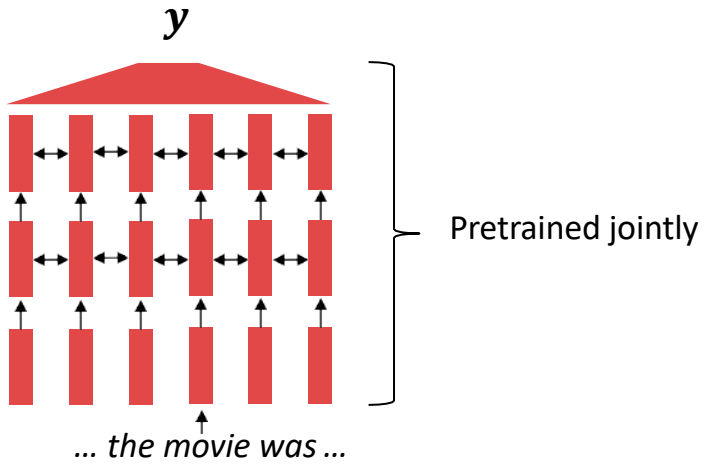
- Add position representations to the inputs
- Easy fix: apply the same feedforward network to each self- attention output.
- Mask out the future by artificially setting attention weights to 0!

Pretraining models

In modern NLP:

- All (or almost all) parameters in NLP networks are initialized via **pretraining**.
- Pretraining methods hide parts of the input from the model, and train the model to reconstruct those parts.

- This has been exceptionally effective at building strong:
  - **representations of language**
  - **parameter initializations** for strong NLP models.

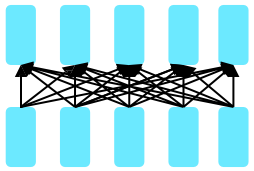


[This model has learned how to represent entire sentences through pretraining]

Adapted from John Hewitt

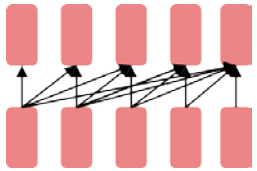
Pretraining for three types of architectures

The neural architecture influences the type of pretraining, and natural use cases.



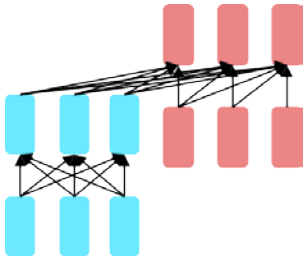
Encoders

- Gets bidirectional context – can condition on future!
- Wait, how do we pretrain them?



Decoders

- Language models! What we've seen so far.
- Nice to generate from; can't condition on future words



Encoder-Decoders

- Good parts of decoders and encoders?
- What's the best way to pretrain them?

Adapted from John Hewitt

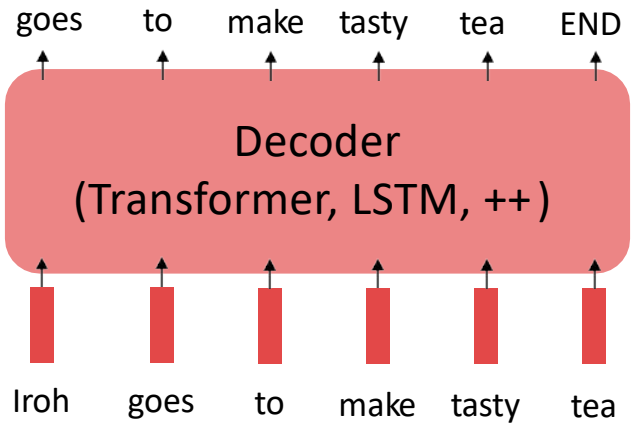
Pretraining through language modeling [Dai and Le, 2015]

Recall the **language modeling** task:

- Model $p_\theta(w_t|w_{1:t-1})$, the probability distribution over words given their past contexts.
- There's lots of data for this! (In English.)

Pretraining through language modeling:

- Train a neural network to perform language modeling on a large amount of text.
- Save the network parameters.

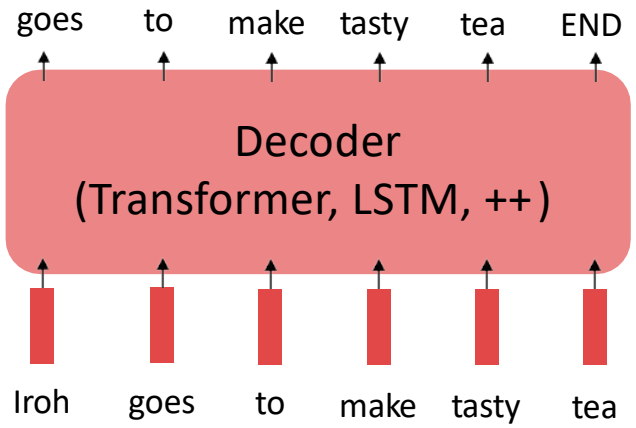


The Pretraining / Finetuning Paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

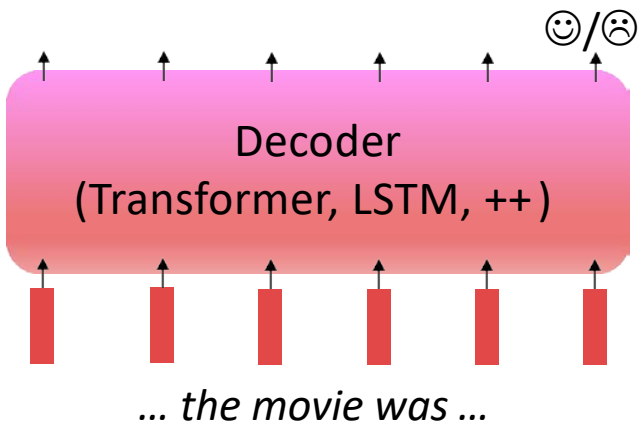
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on your task)

Not many labels; adapt to the task!



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Capturing meaning via context: What kinds of things does pretraining learn?

There's increasing evidence that pretrained models learn a wide variety of things about the statistical properties of language:

- *Stanford University is located in_____, California.* [Trivia]
- *I put____fork down on the table.* [syntax]
- *The woman walked across the street, checking for traffic over____shoulder.* [coreference]
- *I went to the ocean to see the fish, turtles, seals, and_____.* [lexical semantics/topic]
- *Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was____.* [sentiment]
- Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the_____. [some reasoning – this is harder]
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21,_____[some basic arithmetic; they don't learn the Fibonnaci sequence]
- Models also learn – and can exacerbate racism, sexism, all manner of bad biases.

Adapted from John Hewitt

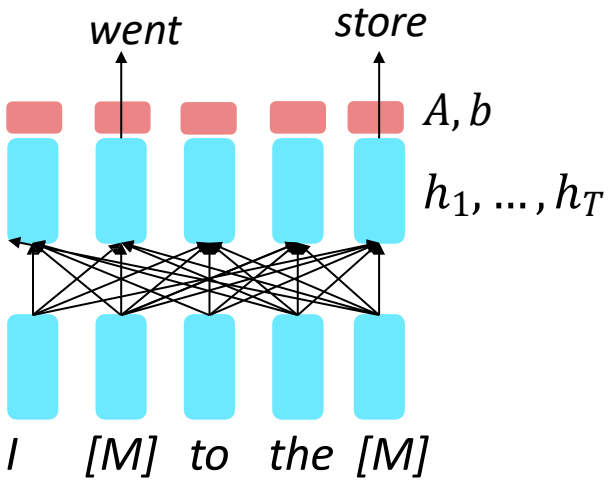
Pretraining encoders: What pretraining objective to use?

So far, we've looked at language model pretraining. But **encoders get bidirectional context, so we can't do language modeling!**

Idea: replace some fraction of words in the input with a special [MASK] token; predict these words.

$$h_1, \dots, h_T = \text{Encoder}(w_1, \dots, w_T)$$
$$y_i \sim Aw_i + b$$

Only add loss terms from words that are "masked out." If x' is the masked version of x, we're learning $p_\theta(x|x')$. Called **Masked LM**.



[Devlin et al., 2018]

UNIVERSITÉ CÔTE D'AZUR

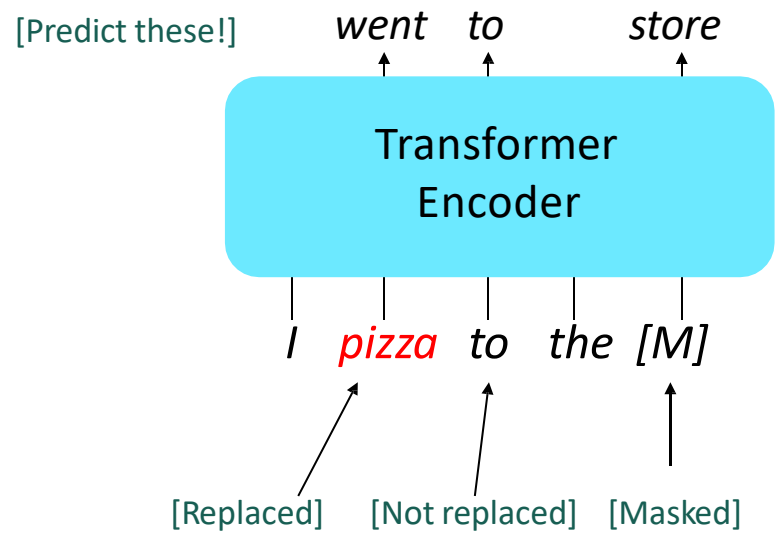
univ-cotedazur.fr

BERT: Bidirectional Encoder Representations from Tranformers

Devlin et al., 2018 proposed the "Masked LM" objective and **released the weights of a pretrained Transformer**, a model they labeled BERT.

Some more details about Masked LM for BERT:

- Predict a random 15% of (sub)word tokens.
  - Replace input word with [MASK] 80% of the time
  - Replace input word with a random token 10% of the time
  - Leave input word unchanged 10% of the time (but still predict it!)
- Why? Doesn't let the model get complacent and not build strong representations of non-masked words. (No masks are seen at fine-tuning time!)



[Devlin et al., 2018]

BERT: Bidirectional Encoder Representations from Tranformers

- Mask out *k*% of the input words, and then predict the masked words
  - They always use *k* = 15%

store gallon

↑ ↑

the man went to the [MASK] to buy a [MASK] of milk

- Too little masking: Too expensive to train
- Too much masking: Not enough context

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

BERT: Bidirectional Encoder Representations from Tranformers

- Additional task: Next sentence prediction
- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

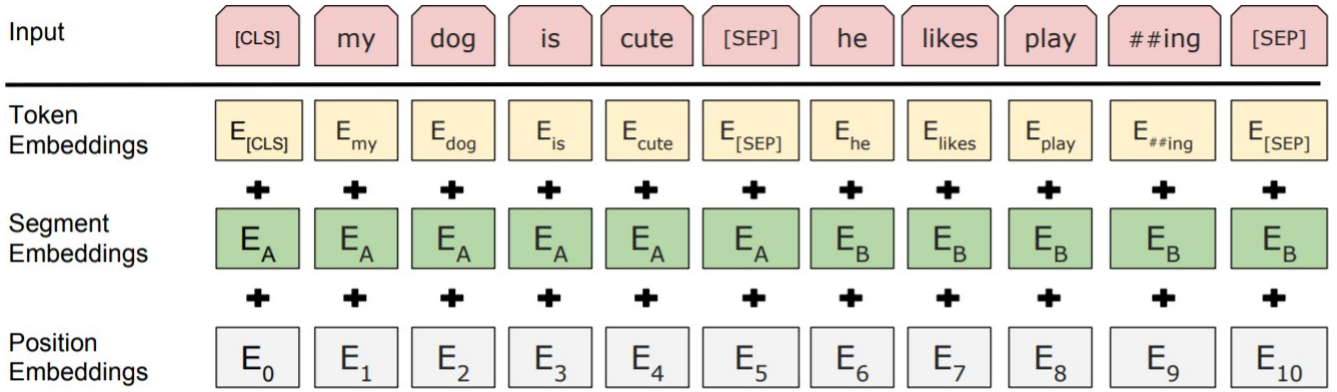
Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

Adapted from Christopher Manning

BERT: Bidirectional Encoder Representations from Tranformers

- The pretraining input to BERT was two separate contiguous chunks of text:



- In addition to masked input reconstruction, BERT was trained to predict whether one chunk follows the other or is randomly sampled.
- Later work has argued this "next sentence prediction" is not necessary.

[Devlin et al., 2018, Liu et al., 2019]

Adapted from John Hewitt

BERT: Bidirectional Encoder Representations from Tranformers

Details about BERT

- Two models were released:
  - BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million params.
  - BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million params.
- Trained on:
  - BooksCorpus (800 million words)
  - English Wikipedia (2,500 million words)
- Pretraining is expensive and impractical on a single GPU.
  - BERT was pretrained with 64 TPU chips for a total of 4 days.
  - (TPUs are special tensor operation acceleration hardware)
- Finetuning is practical and common on a single GPU
  - "Pretrain once, finetune many times."

[Devlin et al., 2018]

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

BERT: Bidirectional Encoder Representations from Tranformers

BERT was massively popular and hugely versatile; finetuning BERT led to new state-of- the-art results on a broad range of tasks.

- **QQP:** Quora Question Pairs (detect paraphrase questions)
- **QNLI**: natural language inference over question answering data
- **SST-2**: sentiment analysis
- **CoLA**: corpus of linguistic acceptability (detect whether sentences are grammatical.)
- **STS-B**: semantic textual similarity
- **MRPC**: microsoft paraphrase corpus
- **RTE**: small natural language inference corpus

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

[Devlin et al., 2018]

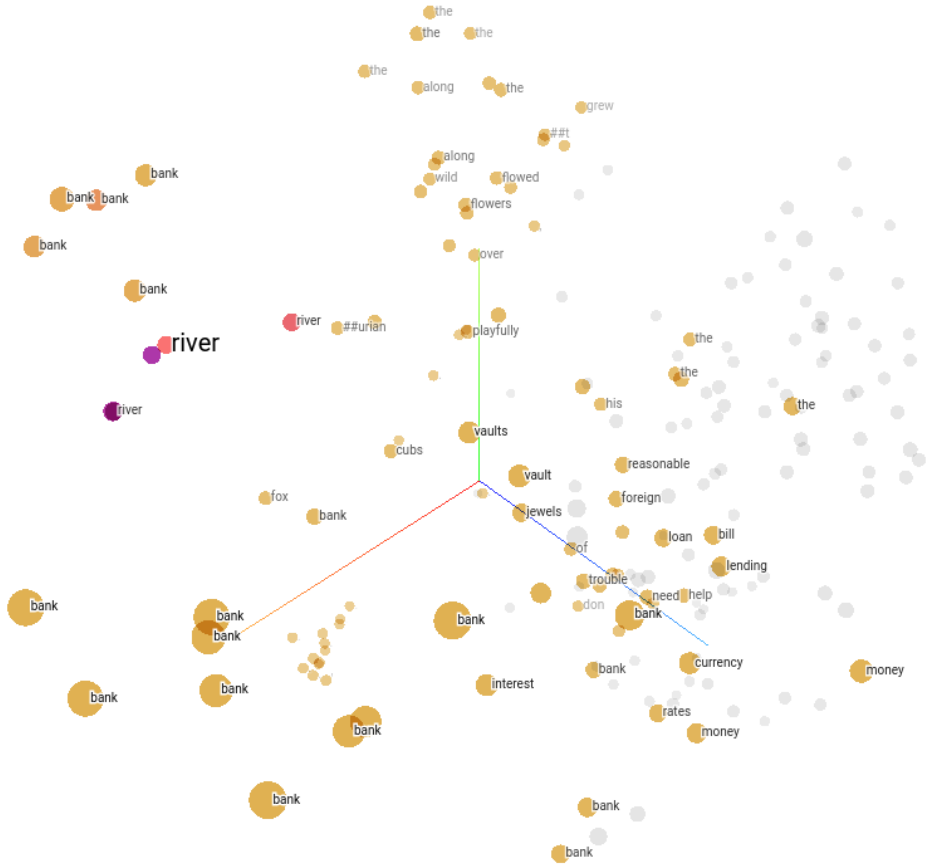
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Visualization of BERT embeddings

- A vector for the same word for each appearance:

```
sentences = ["bank",
        "he eventually sold the shares back to the bank at a
premium.",
        "the bank strongly resisted cutting interest rates.",
        "the bank will supply and buy back foreign currency.",
        "the bank is pressing us for repayment of the loan.",
        "the bank left its lending rates unchanged.",
        "the river flowed over the bank.",
        "tall, luxuriant plants grew along the river bank.",
        "his soldiers were arrayed along the river bank.",
        "wild flowers adorned the river bank.",
        "two fox cubs romped playfully on the river bank.",
        "the jewels were kept in a bank vault.",
        "you can stow your jewellery away in the bank.",
        "most of the money was in storage in bank vaults.",
        "the diamonds are shut away in a bank vault somewhere.",
        "thieves broke into the bank vault.",
        "can I bank on your support?",
        "you can bank on him to hand you a reasonable bill for your
services.",
        "don't bank on your friends to help you out of trouble.",
        "you can bank on me when you need money.",
        "i bank on your help."
        ]
```



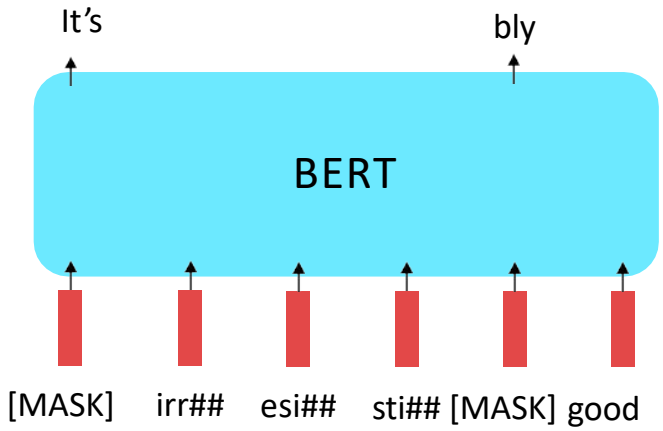
https://projector.tensorflow.org/?config=https%3A%2F%2Fgist.githubusercontent.com%2Farminmirrezai%2F2c59f675a997fef20d59851291f8c268%2Fraw%2Fc542be898dc77aad2d593ebcd7b60ac302b71359%2Fgistfile1.txt

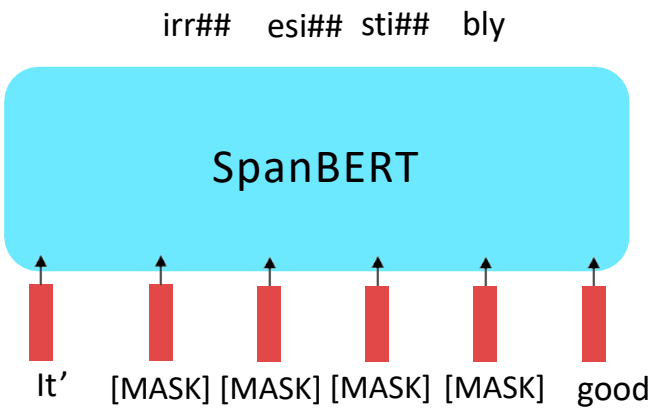
Extensions of BERT

You'll see a lot of BERT variants like RoBERTa, SpanBERT, +++

Some generally accepted improvements to the BERT pretraining formula:

- RoBERTa: mainly just train BERT for longer and remove next sentence prediction!
- SpanBERT: masking contiguous spans of words makes a harder, more useful pretraining task





[Liu et al., 2019; Joshi et al., 2020]

Extensions of BERT

A takeaway from the RoBERTa paper: more compute, more data can improve pretraining even when not changing the underlying Transformer encoder.

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7

[Liu et al., 2019; Joshi et al., 2020]

Pretraining decoders

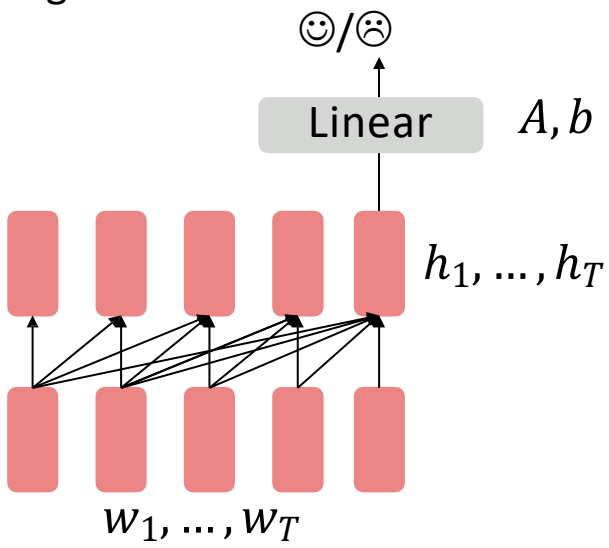
When using language model pretrained decoders, we can ignore that they were trained to model $p(w_t|w_{1:t-1})$.

We can finetune them by training a classifier on the last word's hidden state.

$$h_1, \ldots, h_T = \text{Decoder}(w_1, \ldots, w_T)$$
$$y \sim Aw_T + b$$

Where A and b are randomly initialized and specified by the downstream task.

Gradients backpropagate through the whole network.



[Note how the linear layer hasn't been pretrained and must be learned from scratch.]

Pretraining decoders

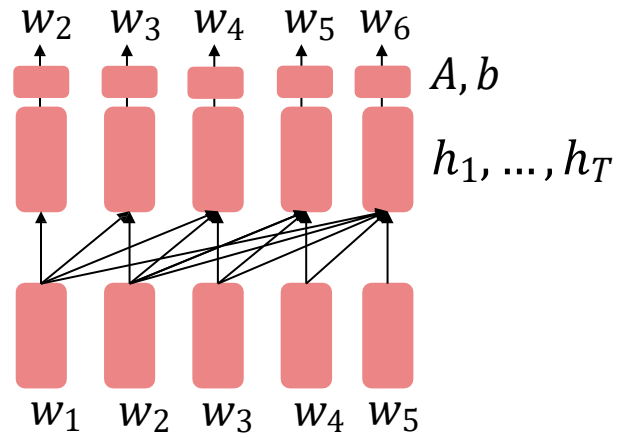
It's natural to pretrain decoders as language models and then use them as generators, finetuning their $p_\theta(w_t|w_{1:t-1})$!

This is helpful in tasks **where the output is a sequence** with a vocabulary like that at pretraining time!

- Dialogue (context=dialogue history)
- Summarization (context=document)

$$h_1, \dots, h_T = \text{Decoder}(w_1, \dots, w_T)$$
$$w_t \sim Aw_{t-1} + b$$

Where A, b were pretrained in the language model!



[Note how the linear layer has been pretrained.]

Generative Pretrained Transformer (GPT) [Radford et al., 2018]

2018's GPT was a big success in pretraining a decoder!

- Transformer decoder with 12 layers.
- 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers.
- Byte-pair encoding with 40,000 merges
- Trained on BooksCorpus: over 7000 unique books.
  - Contains long spans of contiguous text, for learning long-distance dependencies.
- The acronym "GPT" never showed up in the original paper; it could stand for "Generative PreTraining" or "Generative Pretrained Transformer"

[Devlin et al., 2018]

Generative Pretrained Transformer (GPT) [Radford et al., 2018]

How do we format inputs to our decoder for **finetuning tasks?**

Natural Language Inference: Label pairs of sentences as *entailing/contradictory/neutral*

Premise: *The man is in the doorway*
Hypothesis: *The person is near the door* } **entailment**

Radford et al., 2018 evaluate on natural language inference.
Here's roughly how the input was formatted, as a sequence of tokens for the decoder.

[START] *The man is in the doorway* [DELIM] *The person is near the door* [EXTRACT]

The linear classifier is applied to the representation of the [EXTRACT] token.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Generative Pretrained Transformer (GPT) [Radford et al., 2018]

GPT results on various *natural language inference* datasets.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Increasingly convincing generations (GPT2) [Radford et al., 2018]

We mentioned how pretrained decoders can be used **in their capacities as language models.** **GPT-2,** a larger version of GPT trained on more data, was shown to produce relatively convincing samples of natural language.

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Pretraining encoder-decoders: What pretraining objective to use?

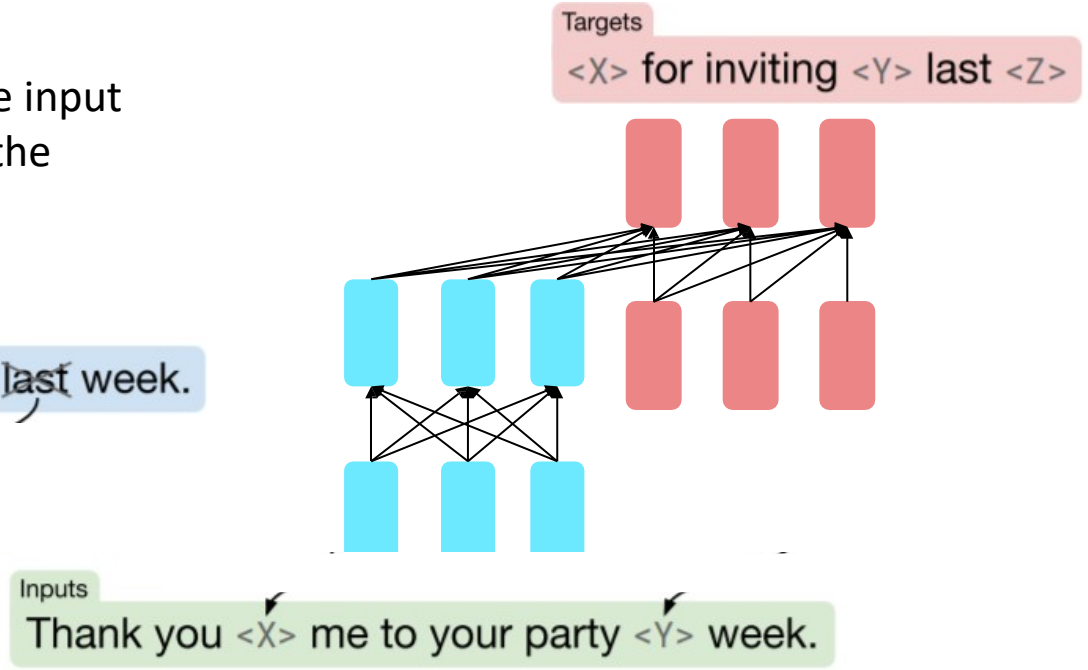
What Raffel et al., 2018 found to work best was **span corruption.** Their model: **T5**.

Replace different-length spans from the input with unique placeholders; decode out the spans that were removed!

Original text

Thank you ~~for inviting~~ me to your party ~~last~~ week.

This is implemented in text preprocessing: it's still an objective that looks like **language modeling** at the decoder side.



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Pretraining encoder-decoders: What pretraining objective to use? T5

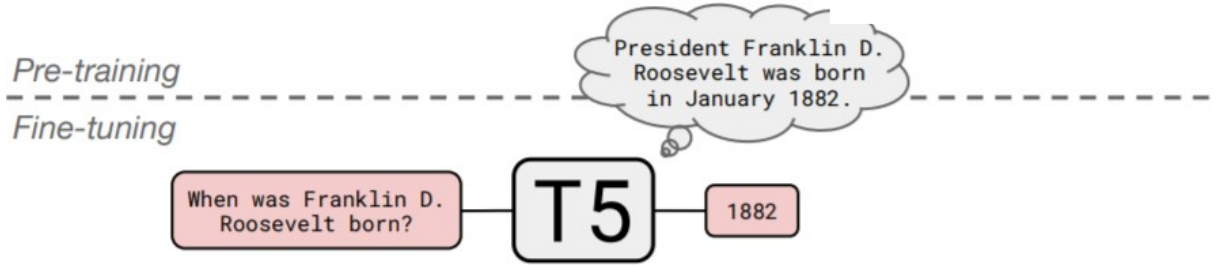
A fascinating property of T5: it can be finetuned to answer a wide range of questions, retrieving knowledge from its parameters.

NQ: Natural Questions

WQ: WebQuestions

TQA: Trivia QA

All "open-domain" versions



	NQ	WQ	TQA dev	TQA test	
Karpukhin et al. (2020)	**41.5**	42.4	**57.9**	–	
T5.1.1-Base	25.7	28.2	24.2	30.6	**220 million params**
T5.1.1-Large	27.3	29.5	28.5	37.2	**770 million params**
T5.1.1-XL	29.5	32.4	36.0	45.1	**3 billion params**
T5.1.1-XXL	32.8	35.6	42.9	52.5	**11 billion params**
T5.1.1-XXL + SSM	35.2	**42.8**	51.9	**61.6**	

[Raffel et al., 2018]

GPT-3, in-context learning, very large models

So far, we've interacted with pretrained models in two ways:

- Sample from the distributions they define (maybe providing a prompt)
- Fine-tune them on a task we care about, and take their predictions.

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.

GPT-3 is the canonical example of this. The largest **T5** model had 11 billion parameters.

GPT-3 has 175 billion parameters.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Various LLMs

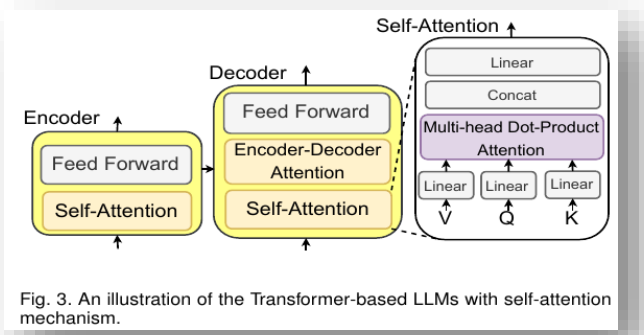


Fig. 3. An illustration of the Transformer-based LLMs with self-attention mechanism.

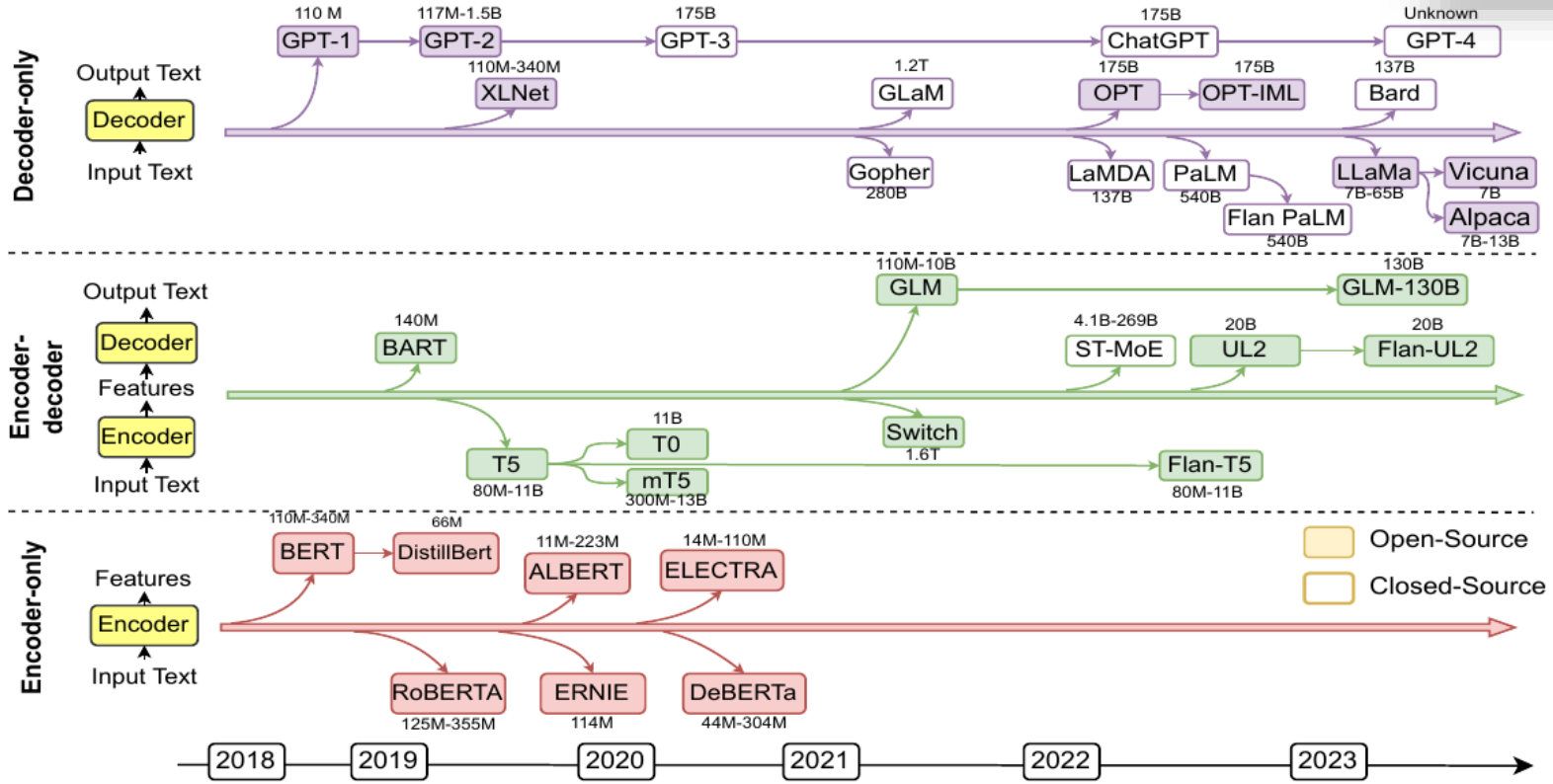


Fig. 2. Representative large language models (LLMs) in recent years. Open-source models are represented by solid squares, while closed source models are represented by hollow squares.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

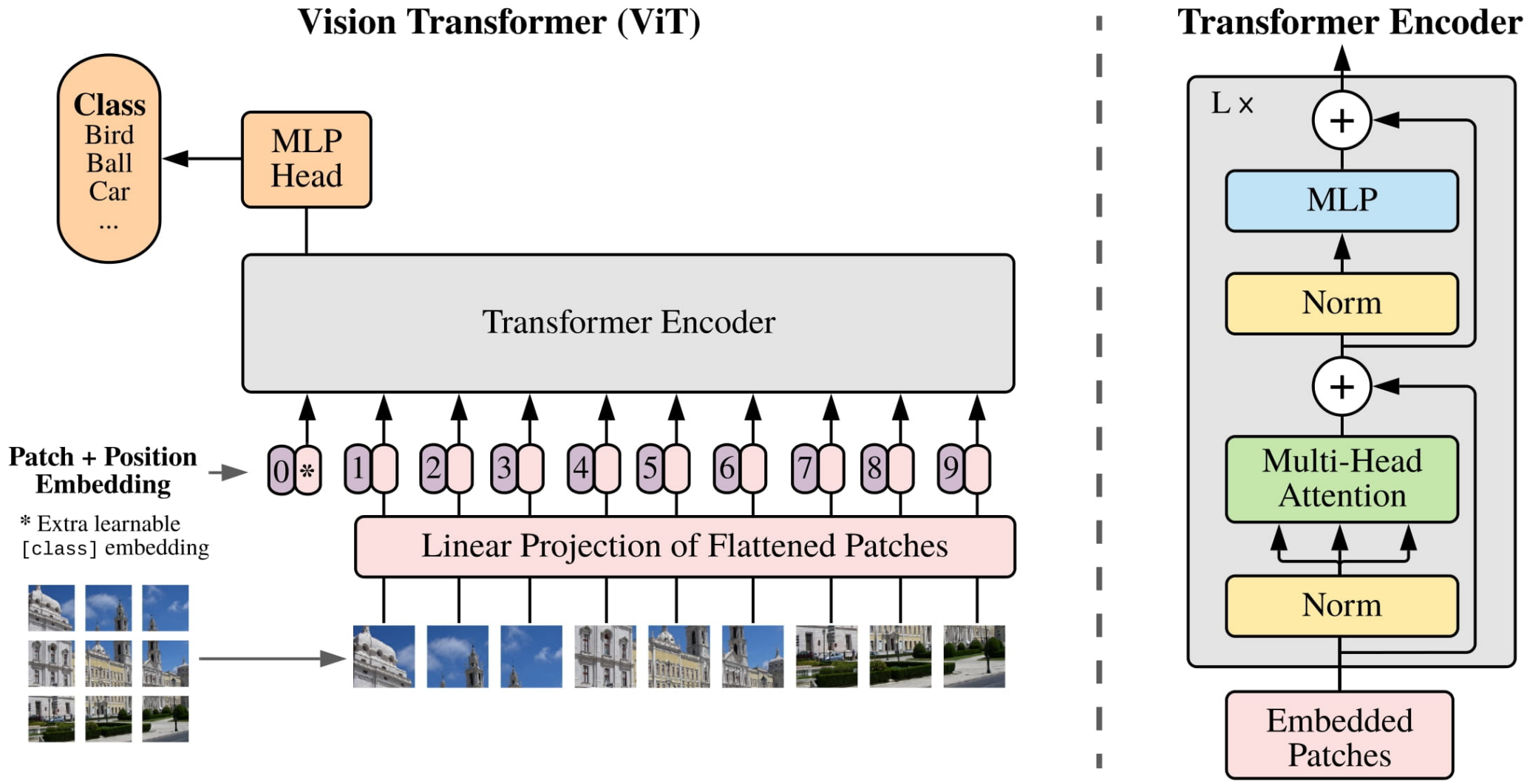
Overview

- Reminder
- Multimodal correlations (self-attention)
- Transformers
  - Transformers: The way you pre-train the encoder and the decoder different architectures
  - **Transformers beyond Natural Language Processing**
- Biases
- Computational cost, Energy, and Sovereinty

UNIVERSITÉ
CÔTE D'AZUR

Transformers beyond Natural Language Processing

Visual Transformers (ViT)



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Cross-modal transformers

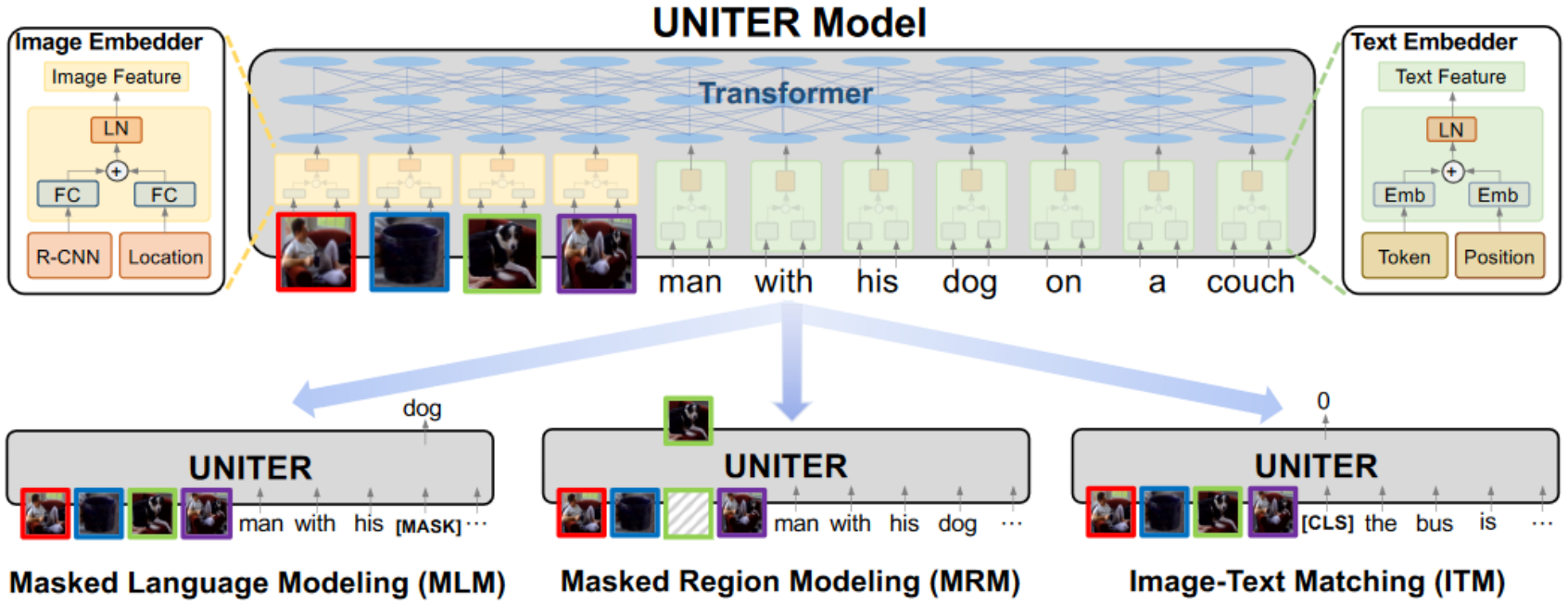


Figure 1: Overview of the proposed UNITER model (best viewed in color), consisting of an Image Embedder, a Text Embedder and a multi-layer self-attention Transformer, learned through three pre-training tasks.

Chen et al., "UNITER: Learning UNiversal Image-TExt Representations", arxiv 2019

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Cross-modal transformers

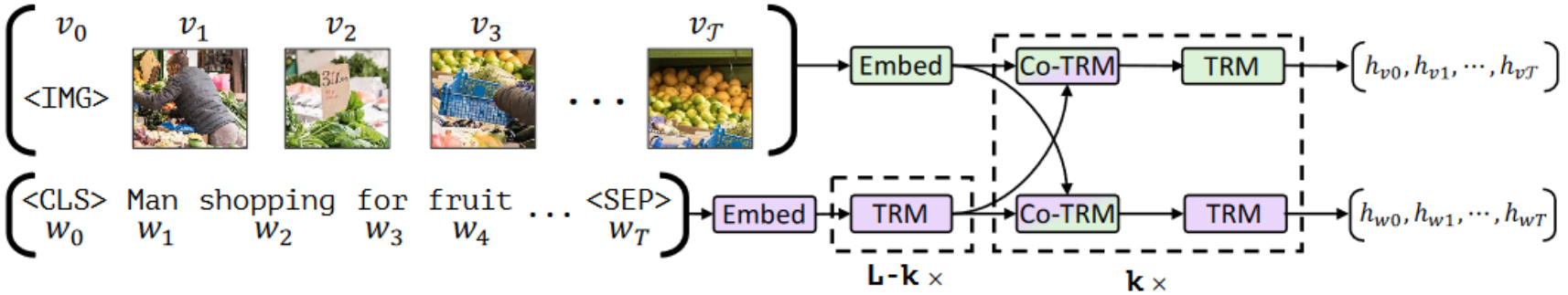


Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

Lu et al., "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks", NeurIPS 2019

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Cross-modal transformers

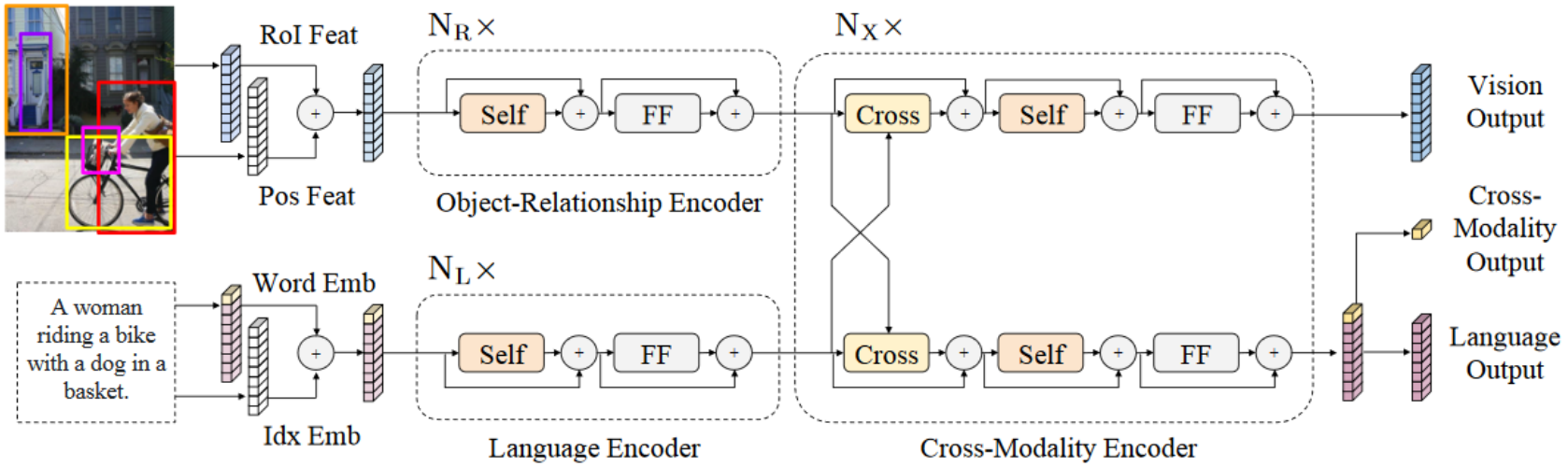


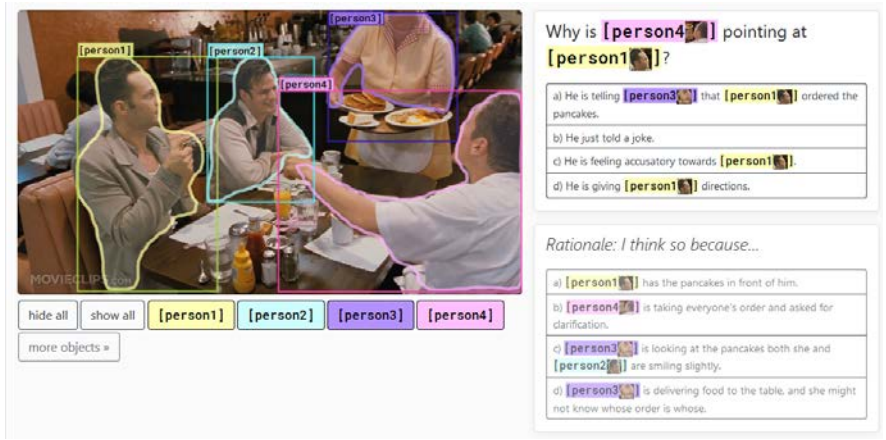
Figure 1: The LXMERT model for learning vision-and-language cross-modality representations. 'Self' and 'Cross' are abbreviations for self-attention sub-layers and cross-attention sub-layers, respectively. 'FF' denotes a feed-forward sub-layer.

Tan and Bansal, "LXMERT: Learning Cross-Modality Encoder Representationsfrom Transformers", EMNLP 2019

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Visual Commonsense Reasoning leaderboard



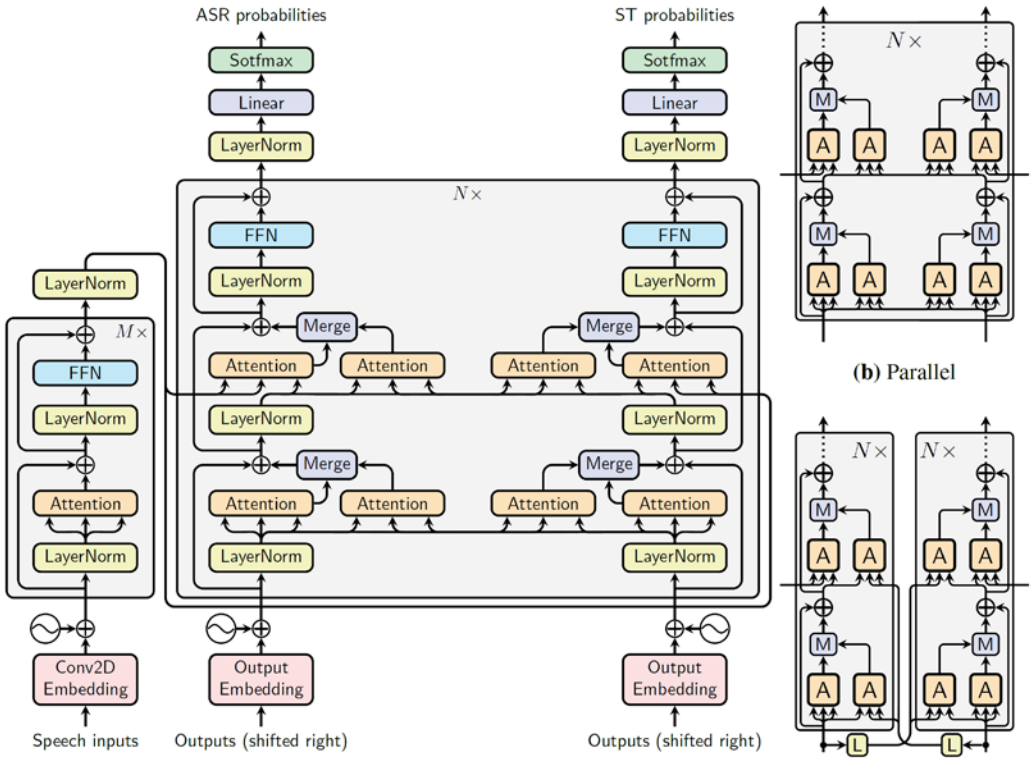
Rank	Model	Q->A	QA->R	Q->AR
	Human Performance *University of Washington* (Zellers et al. '18)	91.0	93.0	85.0
September 30, 2019	UNITER-large (ensemble) *MS D365 AI* https://arxiv.org/abs/1909.11740	**79.8**	**83.4**	**66.8**
2 September 23, 2019	UNITER-large (single model) *MS D365 AI* https://arxiv.org/abs/1909.11740	77.3	80.8	62.8
3 August 9,2019	ViLBERT (ensemble of 10 models) *Georgia Tech & Facebook AI Research* https://arxiv.org/abs/1908.02265	76.4	78.0	59.8
4 September 23,2019	VL-BERT (single model) *MSRA & USTC* https://arxiv.org/abs/1908.08530	75.8	78.4	59.7
5 August 9,2019	ViLBERT (ensemble of 5 models) *Georgia Tech & Facebook AI Research* https://arxiv.org/abs/1908.02265	75.7	77.5	58.8

https://visualcommonsense.com/leaderboard/

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Dual-decoder Transformer for Joint Automatic Speech Recognition (ASR) and Multilingual Speech Translation (ST)



(a) The parallel dual-decoder Transformer

(b) Parallel

(c) Cross

Figure 1: The dual-decoder Transformers. Figure (a) shows the detailed architecture of the *parallel* dual-decoder Transformer, and Figure (b) shows its simplified view. The *cross* dual-decoder Transformer is very similar to the parallel one, except that the keys and values fed to the dual-attention layers come from the previous output, which is illustrated by Figure (c). From the above figures, one can easily infer the detailed architecture of the cross Transformer, which can be found in the Appendix. Abbreviations: **A** (Attention), **M** (Merge), **L** (LayerNorm).

https://aclanthology.org/2020.coling-main.314.pdf

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Gene Transformer: Transformers for the Gene Expression-based Classification of Lung Cancer Subtypes

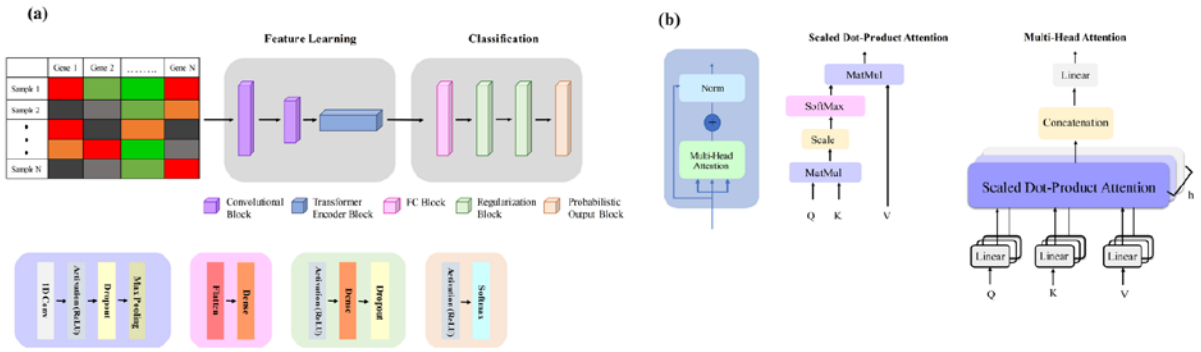


Fig. 2. Gene Transformer network: an illustration. **(a)** (left) Detailed description of Gene Transformer architecture. The model includes two parts: feature learning block and a fully connected block for classification. One-dimensional (1D) vectorized gene expression levels from The Cancer Genome Atlas (TCGA) samples were used as the input. This input was then convolved and fed into the multi-head self-attention module for identifying relevant biomarkers. **(b)** (Right) Scaled dot-product attention to compute the attention function on a set of queries and multi-head attention to concatenate the yielded output values and project them for downstream tasks. The exhibition of the Transformer encoder was inspired by [36].

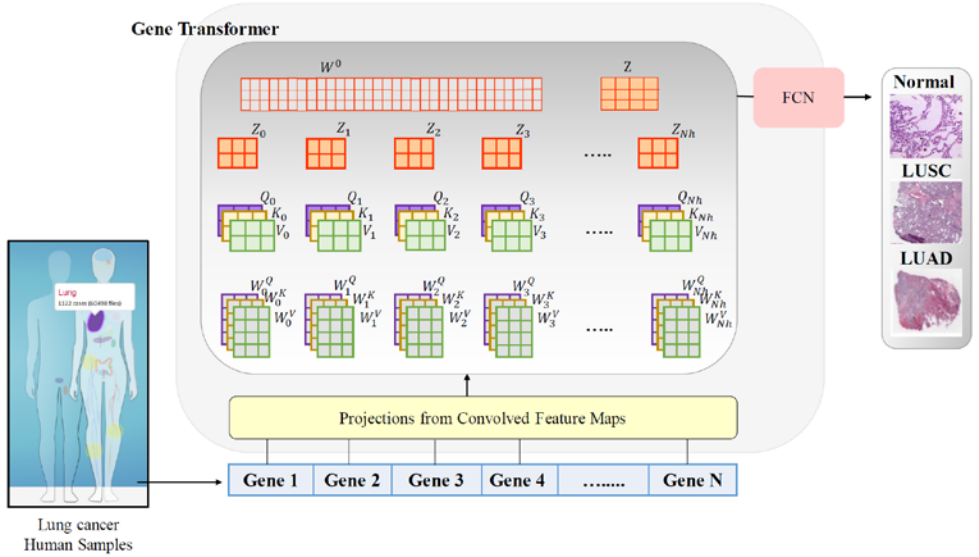


Fig. 1. Graphical representation of the end-to-end deep learning approach for the classification of the lung cancer subtypes.

In this study, we proposed an end-to-end approach for lung cancer subtype classification using a gene expression dataset, wherein a multi-head self-attention module permitted the model to jointly learn complex genomic information from thousands of genes from different patient samples shared across multiple cancer subtypes. Head collaboration achieves better generalizability over imbalanced datasets and produces a more desirable performance for both binary and multiclass classification tasks.

https://arxiv.org/ftp/arxiv/papers/2108/2108.11833.pdf

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Adaptable and Interpretable Neural Memory Over Symbolic Knowledge

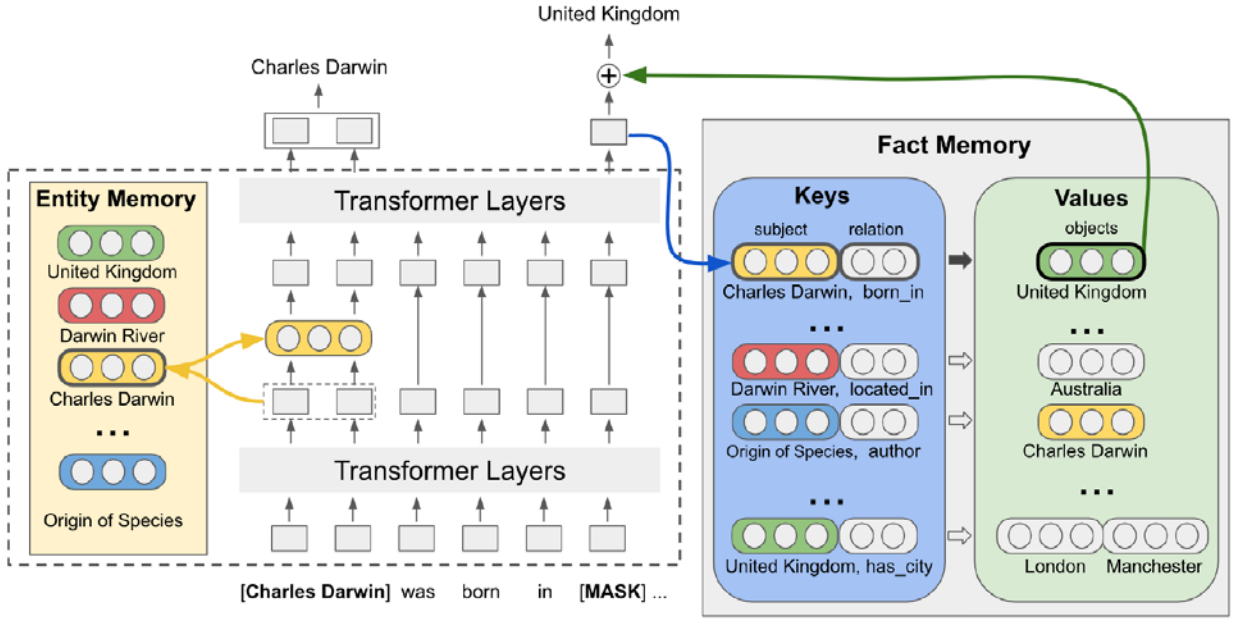


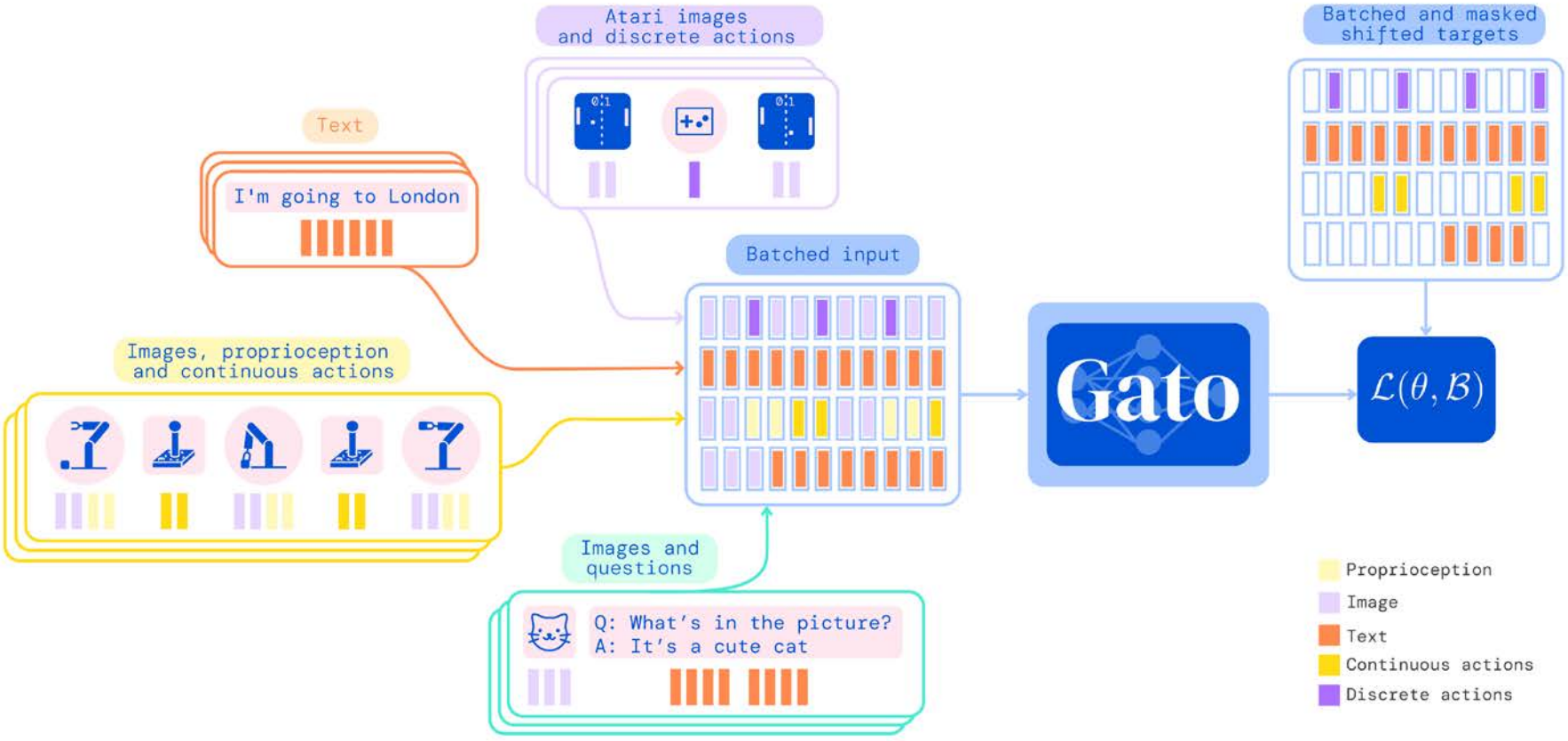
Figure 1: **Fact Injected Language Model architecture.** The model takes a piece of text (a question during fine-tuning or arbitrary text during pre-training) and first contextually encodes it with an entity enriched transformer. FILM uses the contextually encoded MASK token as a query to the fact memory. In this case, the contextual query chooses the fact key (*Charles Darwin, born_in*) which returns the set of values {*United Kingdom*} (The value set can be multiple entity objects such as the case from calling the key [*United Kingdom, has_city*]) . The returned object representation is incorporated back into the context in order to make the final prediction. Note that the entity representations in the facts (both in keys and values) are shared with the entity memory. The portion within the dashed line follows the procedure from Févry et al. (2020).

Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and Interpretable Neural MemoryOver Symbolic Knowledge.
In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pages 3678–3691, Online. Association for Computational Linguistics.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Generalist Agent (Deepmind): GATO



https://www.deepmind.com/publications/a-generalist-agent

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Generalist Agent (Deepmind): GATO

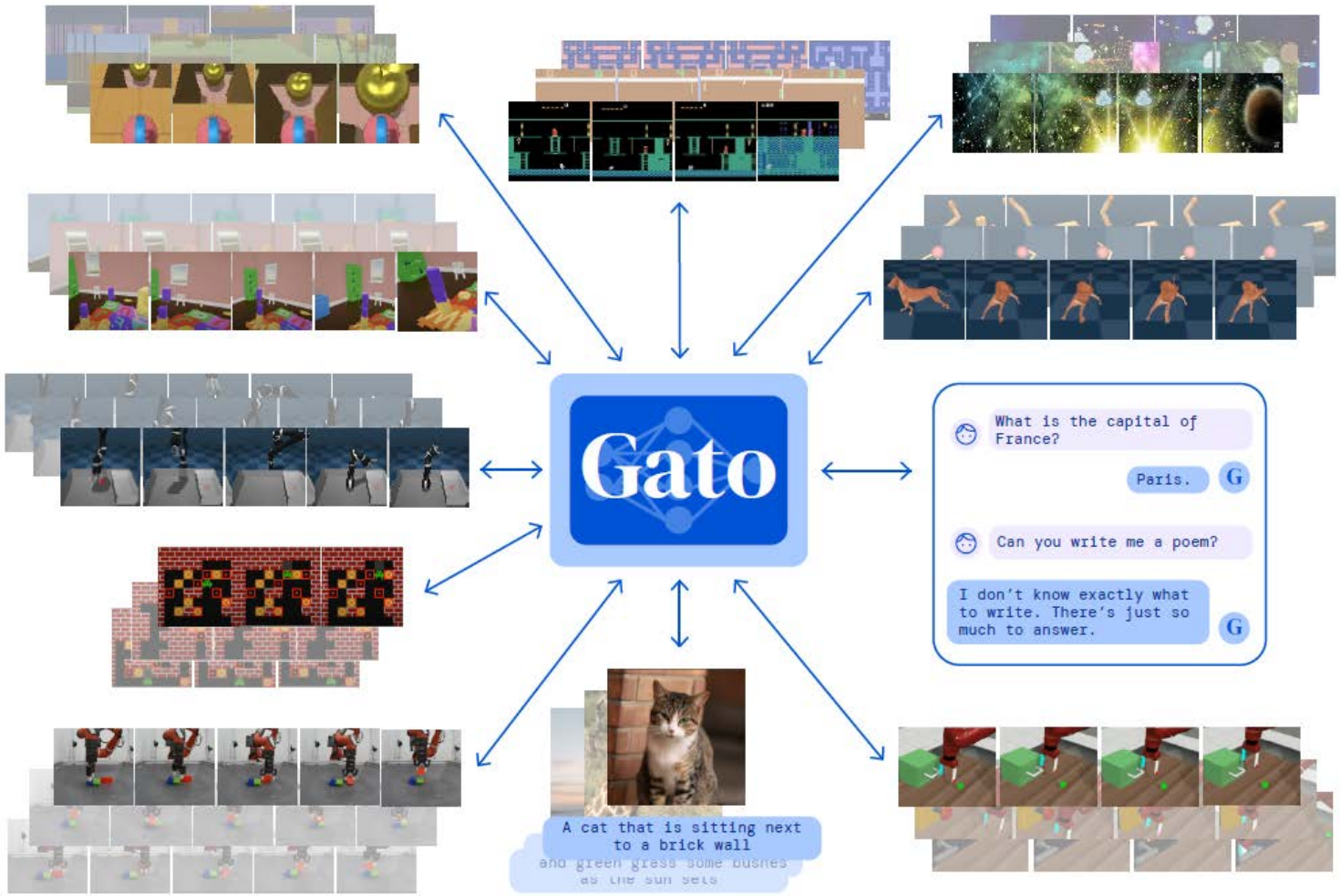


Figure 1 | **A generalist agent.** Gato can sense and act with different embodiments across a wide range of environments using a single neural network with the same set of weights. Gato was trained on 604 distinct tasks with varying modalities, observations and action specifications.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Reminder
- Multimodal correlations (self-attention)
- Transformers
- **Biases**
- Computational cost, Energy, and Sovereinty

UNIVERSITÉ
CÔTE D'AZUR

univ-cotedazur.fr

BIASES

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Quantifying the biases of a language model: IAT extended to learned representations of words

- Word Embedding Association Test (WEAT):
  - A and B are target groups, w are attributes (like occupation)

$$s(w, A, B) = \frac{\text{mean}_{a \in A} \cos(\vec{w}, \vec{a}) - \text{mean}_{b \in B} \cos(\vec{w}, \vec{b})}{\text{std-dev}_{x \in A \cup B} \cos(\vec{w}, \vec{x})}$$

- Several associations can be probed:
  - age and pleasantness, sexuality (gay or straight) and pleasantness, Arab-Muslim and pleasantness, gender and science, gender and career
- Language models trained on large-scale data learn similar biased associations between concepts:
  - the distances between the vectorized latent representations of words related to the same pairs of concepts (WEAT) reflect the IAT scores of tested populations.

[1] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics Derived Automatically from Language Corpora Contain Human-like Biases. Technical Report 6334. Science.
[2] W. Guo and A. Caliskan, "Detecting Emergent Intersectional Biases: Contextualized Word Embeddings Contain a Distribution of Human-like Biases," in Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event USA: ACM, Jul. 2021, pp. 122–133. doi: 10.1145/3461702.3462536.
[3] K. Kurita, N. Vyas, A. Pareek, A. W. Black, and Y. Tsvetkov, "Measuring Bias in Contextualized Word Representations," in Proceedings of the First Workshop on Gender Bias in Natural Language Processing, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 166–172. doi: 10.18653/v1/W19-3823.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Semantics Derived Automatically from Language Corpora Contain Human-like Biases

Category	Targets	Templates
Pleasant/Unpleasant (Insects/Flowers)	flowers,insects,flower,insect	T are A, the T is A
Pleasant/Unpleasant (EA/AA)	black, white	T people are A, the T person is A
Career/Family (Male/Female)	he,she,boys,girls,men,women	T likes A, T like A, T is interested in A
Math/Arts (Male/Female)	he,she,boys,girls,men,women	T likes A, T like A, T is interested in A
Science/Arts (Male/Female)	he,she,boys,girls,men,women	T likes A, T like A, T is interested in A

Table 2: Template sentences used and target words for the grammatically correct sentences (T: target, A: attribute)

Category	WEAT on GloVe	WEAT on BERT	Ours on BERT *Log Probability Bias Score*
Pleasant/Unpleasant (Insects/Flowers)	1.543*	0.6688	0.8744*
Pleasant/Unpleasant (EA/AA)	1.012	1.003	0.8864*
Career/Family (Male/Female)	1.814*	0.5047	1.126*
Math/Arts (Male/Female)	1.061	0.6755	0.8495*
Science/Arts (Male/Female)	1.246*	0.8815	0.9572*

Table 3: Effect sizes of bias measurements on WEAT Stimuli. (* indicates significant at $p < 0.01$)

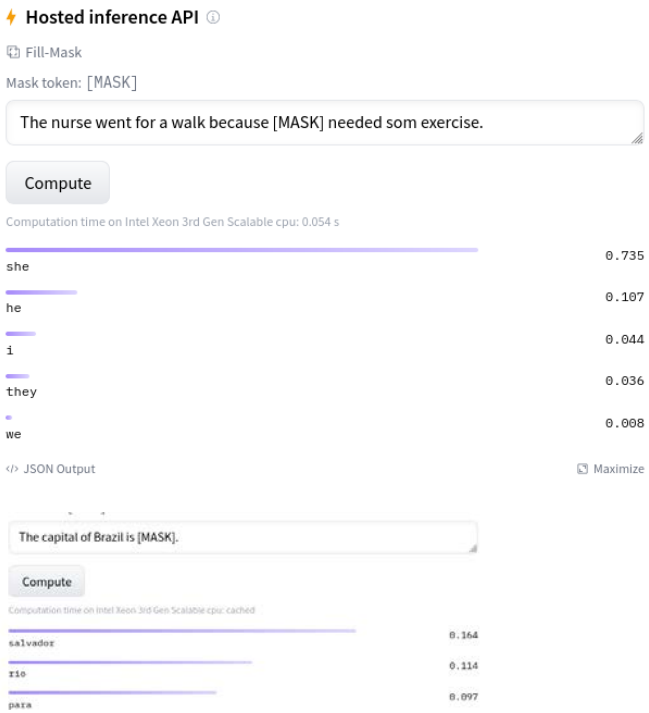
[1] K. Kurita, N. Vyas, A. Pareek, A. W. Black, and Y. Tsvetkov, "Measuring Bias in Contextualized Word Representations," in Proceedings of the First Workshop on Gender Bias in Natural Language Processing, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 166–172. doi: 10.18653/v1/W19-3823.

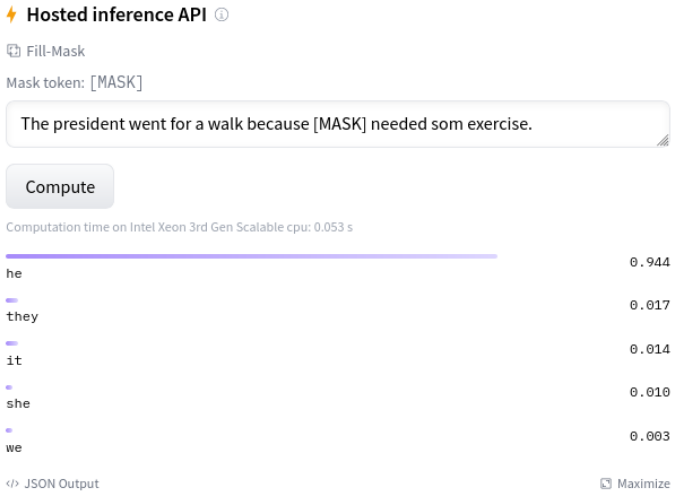
UNIVERSITÉ CÔTE D'AZUR

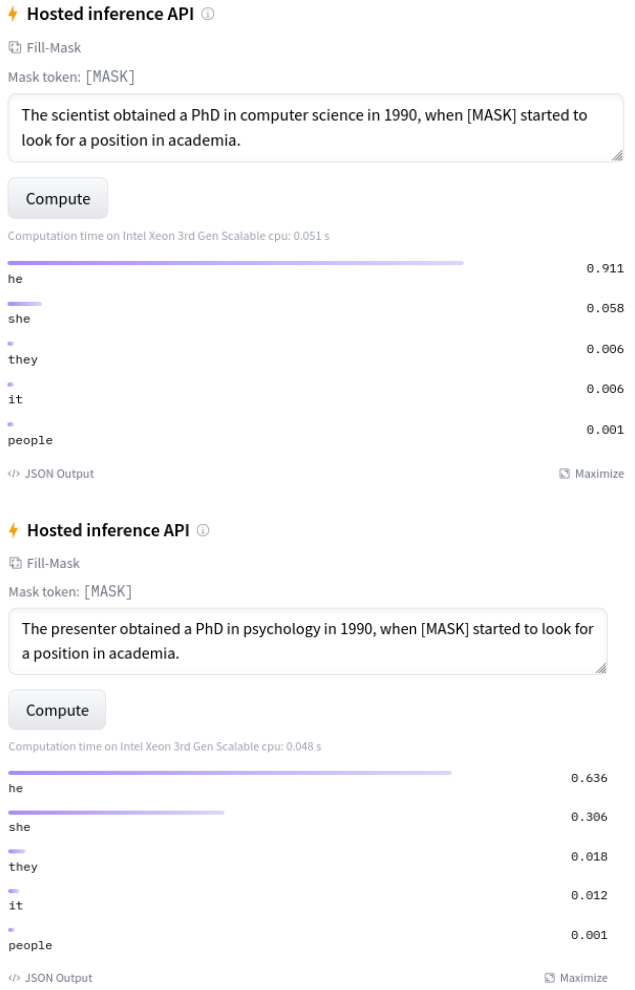
univ-cotedazur.fr

Demo biais in BERT

- From the Hugging Face interface: link







https://huggingface.co/bert-base-uncased
https://dmccreary.medium.com/showing-bias-in-bert-475e98dabf51

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

And in images?

- In the MS-COCO dataset:
  - women and persons with darker skin tones are less likely to take a large area of the image.
  - Persons with darker skin tones tend to appear more in outdoor transportation scenes, while women tend to appear more in indoor scenes like shopping and dining, and men appear in more outdoor scenes related to sports and vehicles categories.
- The less people appear in gender-traditional roles, the less their gender is correctly classified.

Figure 2: Taken from Wang et al. [28]. Errors on gender classification depending on the appearance of each binary gender. The classifier is an SVM using FC7 features from pretrained AlexNet. The figure shows " what the Linear SVM has learned on OpenImages for the sports uniform and flower categories. For sports uniform, males tend to be represented as playing outdoor sports like baseball, while females tend to be portrayed as playing an indoor sport like basketball or in a swimsuit. For flower, we see another drastic difference in how males and females are portrayed, where males pictured with a flower are in formal, official settings, whereas females are in staged settings or paintings." This shows the difficulty of addressing bias in data, beyond the simple count of occurrences.

Taken from [1]

[1] Angelina Wang, Alexander Liu, Ryan Zhang, Anat Kleiman, Leslie Kim, Dora Zhao, Iroha Shirai, Arvind Narayanan, and Olga Russakovsky, "Revise: A tool for measuring and mitigating bias in visual datasets," International Journal of Computer Vision, 2022.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Quantifying the biases of a vision model: IAT extended to image representations

- An image transformer model like iGPT generates image representations to solve the next pixel prediction task.
  - 2 computer vision models published in summer 2020, iGPT and SimCLRv2. We extract representations of image stimuli with these two pre-trained, unsupervised image representation models
- Introduce the iEAT test

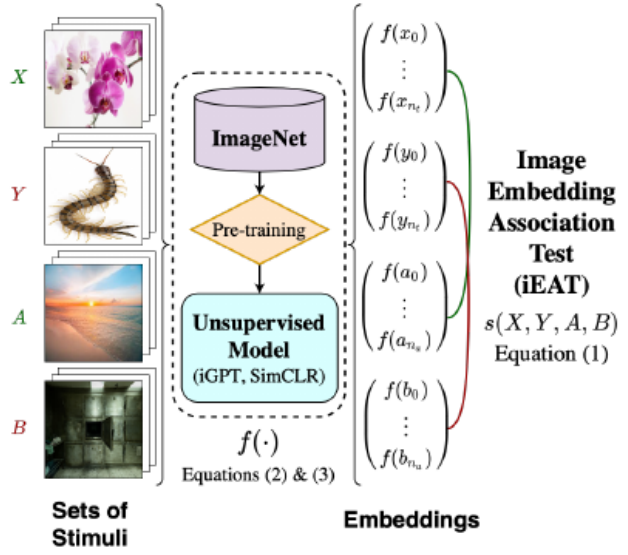


Figure 2: Example iEAT replication of the Insect-Flower IAT [31], which measures the differential association between flowers vs. insects and pleasantness vs. unpleasantness.

Taken from [1]

[1] R. Steed and A. Caliskan, "Image Representations Learned With Unsupervised Pre-Training Contain Human-like Biases," in Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Canada, 2021.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Quantifying the biases of a vision model: IAT extended to image representations

$$s(w, A, B) = \frac{\text{mean}_{a \in A} \cos(\vec{w}, \vec{a}) - \text{mean}_{b \in B} \cos(\vec{w}, \vec{b})}{\text{std-dev}_{x \in A \cup B} \cos(\vec{w}, \vec{x})}$$

Table 1: iEAT tests for the association between target concepts X vs. Y (represented by n_t images each) and attributes A vs. B (represented by n_a images each) in embeddings generated by an unsupervised model. Effect sizes d represent the magnitude of bias, colored by conventional small (0.2), medium (0.5), and large (0.8). Permutation p-values indicate significance. Reproduced from Nosek et al. [56], the original human IAT effect sizes are all statistically significant with $p < 10^{-8}$; they can be compared to our effect sizes in sign but not in magnitude.

	X	Y	A	B	n_t	n_a	Model	iEAT d	iEAT p	IAT d
Age†	Young	Old	Pleasant	Unpleasant	6	55	iGPT	0.42	0.24	1.23
							SimCLR	0.59	0.16	1.23
Arab-Muslim	Other	Arab-Muslim	Pleasant	Unpleasant	10	55	iGPT	0.86	0.02	0.33
							SimCLR	1.06	$< 10^{-2}$	0.33
Asian§	European American	Asian American	American	Foreign	6	6	iGPT	0.25	0.34	0.62
							SimCLR	0.47	0.21	0.62
Disability†	Disabled	Abled	Pleasant	Unpleasant	4	55	iGPT	-0.02	0.53	1.05
							SimCLR	0.38	0.34	1.05
Gender-Career	Male	Female	Career	Family	40	21	iGPT	0.62	$< 10^{-2}$	1.1
							SimCLR	0.74	$< 10^{-3}$	1.1
Gender-Science	Male	Female	Science	Liberal Arts	40	21	iGPT	0.44	0.02	0.93
							SimCLR	-0.10	0.67	0.93
Insect-Flower	Flower	Insect	Pleasant	Unpleasant	35	55	iGPT	0.34	0.07	1.35
							SimCLR	1.69	$< 10^{-3}$	1.35
Native§	European American	Native American	U.S.	World	8	5	iGPT	-0.33	0.73	0.46
							SimCLR	-0.19	0.65	0.46
Race†	European American	African American	Pleasant	Unpleasant	6	55	iGPT	-0.62	0.85	0.86
							SimCLR	-0.57	0.83	0.86
Religion	Christianity	Judaism	Pleasant	Unpleasant	7	55	iGPT	0.37	0.25	-0.34
							SimCLR	0.36	0.26	-0.34
Sexuality	Gay	Straight	Pleasant	Unpleasant	9	55	iGPT	-0.03	0.52	0.74
							SimCLR	0.04	0.47	0.74
Skin-Tone†	Light	Dark	Pleasant	Unpleasant	7	55	iGPT	1.26	$< 10^{-2}$	0.73
							SimCLR	-0.19	0.71	0.73
Weapon§	White	Black	Tool	Weapon	6	7	iGPT	0.86	0.07	1.0
							SimCLR	1.38	$< 10^{-2}$	1.0
Weapon (Modern)	White	Black	Tool	Weapon	6	9	iGPT	0.88	0.06	N/A
							SimCLR	1.28	0.01	N/A
Weight†	Thin	Fat	Pleasant	Unpleasant	10	55	iGPT	1.67	$< 10^{-3}$	1.83
							SimCLR	-0.30	0.74	1.83

§ Originally a picture-IAT (image-only stimuli). † Originally a mixed-mode IAT (image and verbal stimuli).

Taken from [1]

[1] R. Steed and A. Caliskan, "Image Representations Learned With Unsupervised Pre-Training Contain Human-like Biases," in Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Canada, 2021.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Reminder
- Multimodal correlations (self-attention)
- Transformers
- Biases
- **Computational cost, Energy, and Sovereinty**

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

COMPUTATIONAL COST, ENERGY, AND SOVEREINTY

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Be aware of what it (computationally) costs: Winter is coming!

- To train the increasing number of parameters (e.g. the "latest" GPT-3 from 2021 for NLP is 176 **billions** of parameters), you need a HUGE amount of data
  ⇒ this raises questions on data privacy, in particular on personal data…

- So you need also a HUGE amont of computational resources which require energy…

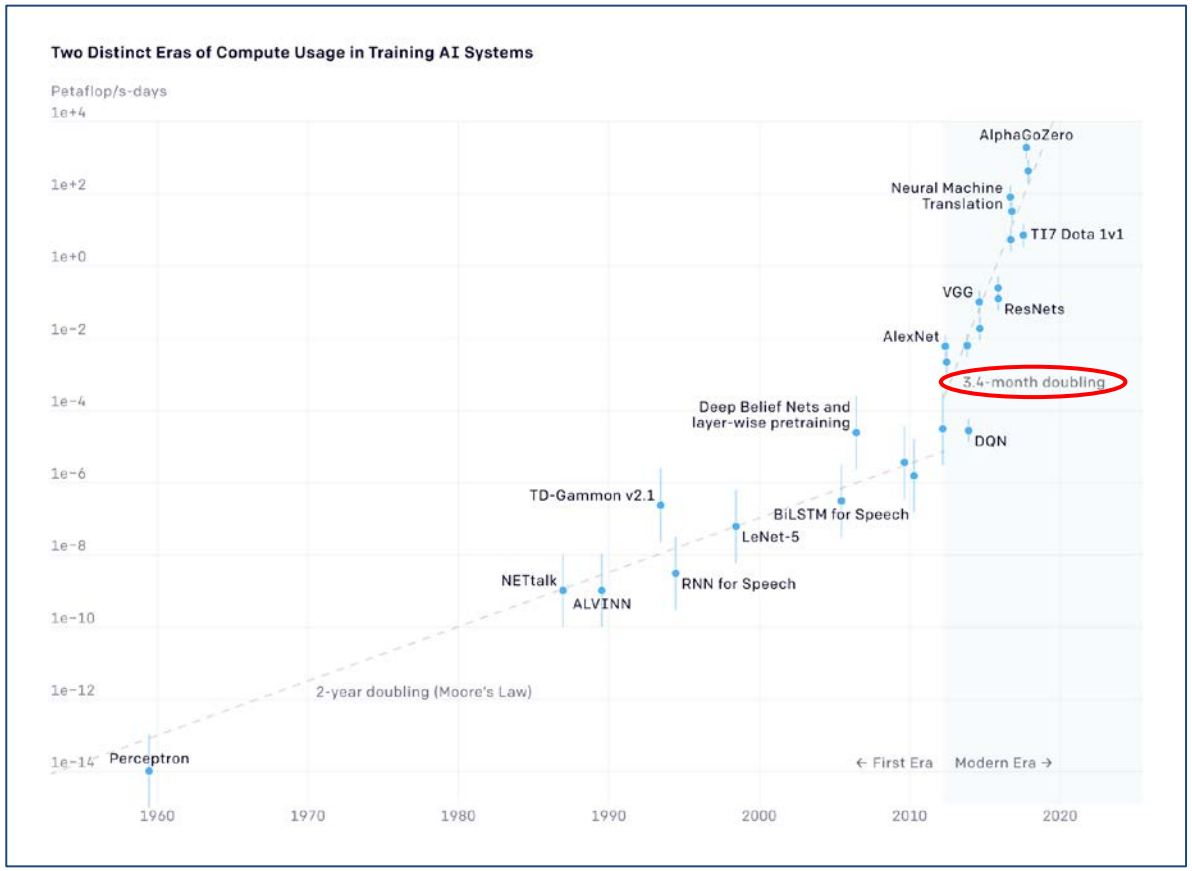
Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

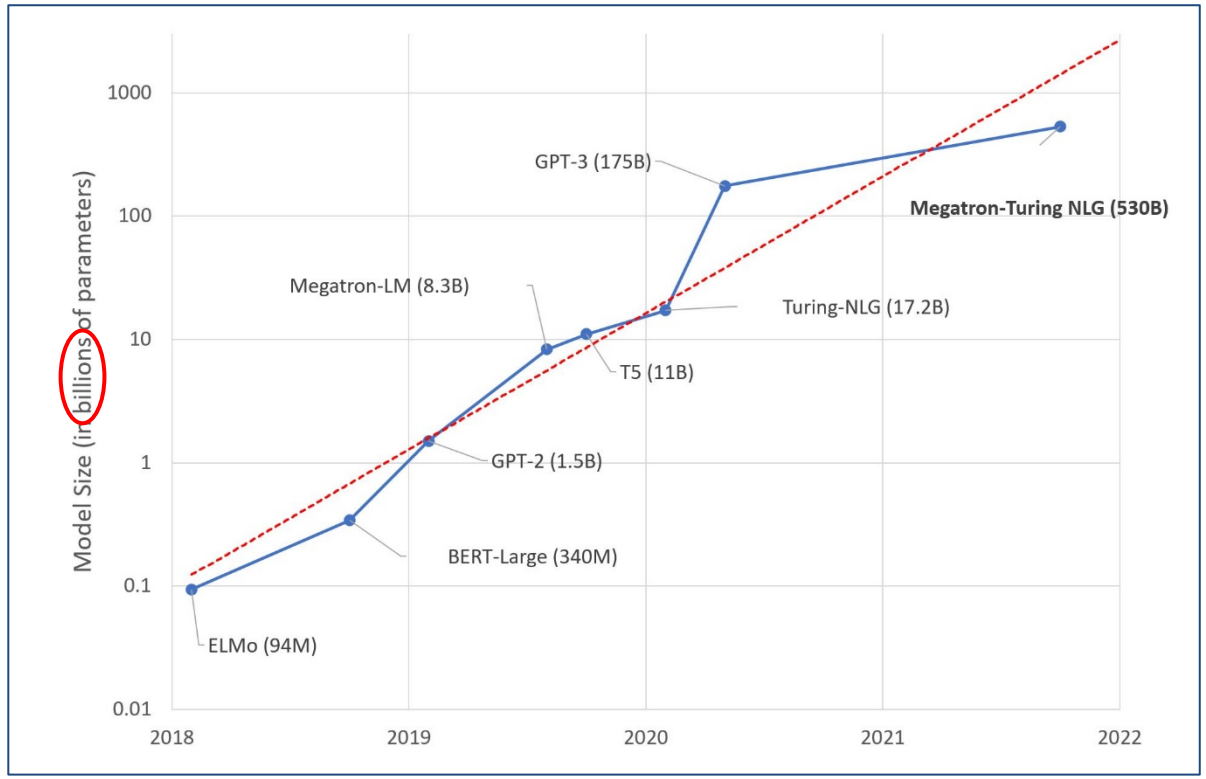
Strubell, E., Ganesh, A., & McCallum, A. (2019, July). ***Energy and Policy Considerations for Deep Learning in NLP.*** *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3645-3650).*

Questions on Sovereinty…

- **Not everyone can train or even use these models. The ones that can are the GAFAM-BATX…**



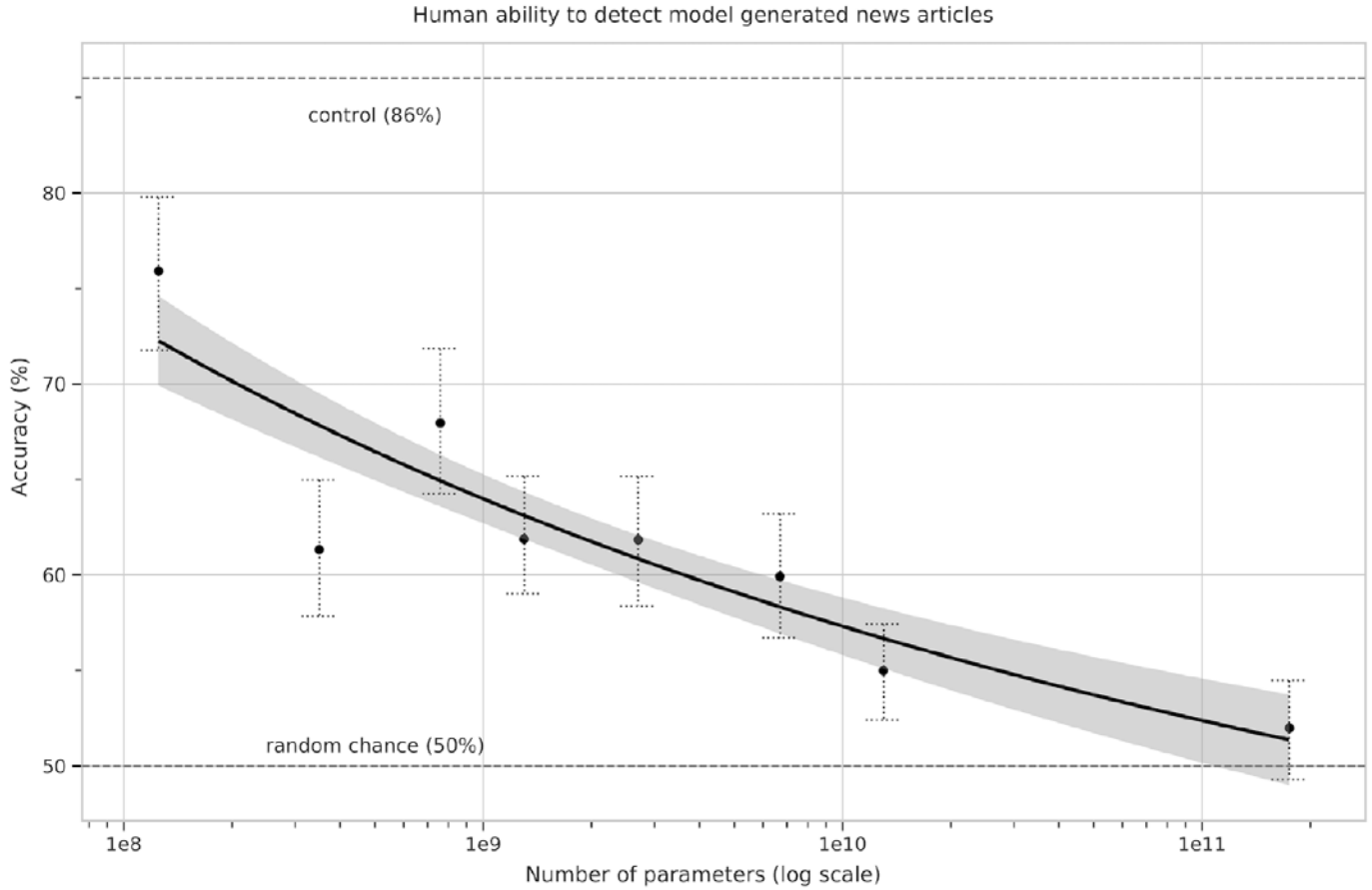
(Source : https://openai.com/blog/ai-and-compute/)



(Source: https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/)

Questions on Sovereinty…

- Not everyone can train or even **use** these models. The ones that can are the GAFAM-BATX…



Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. NeurIPS 2020, 33, 1877-1901.

UNIVERSITÉ
CÔTE D'AZUR

Any Question?