

NLP basics

Introduction to neural NLP

MVA - Speech and Language Processing #2+#3 (NLP 1+2)
2nd part

Benoît Sagot & Robin Algayres
with many slides by Paul Michel

Transformer

Reminder on Generative RNNs

RNN

generative

onehots
word vectors

$$\begin{bmatrix} 0 \\ \dots \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

“I”

“like”

$$\begin{bmatrix} 1 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

“donuts”

$$\begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

“too”

$$\begin{bmatrix} 0 \\ \dots \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Reminder on Generative RNNs

RNN

generative

Remark:

The orthographic representation is removed
LM trained on words/BPE **should** not be able to spell
words ! (chatGPT and Dall e can do it a little bit...)

onehots
word vectors

$$\begin{matrix} \text{onehots} \\ \text{word vectors} \end{matrix} \xrightarrow{\quad} \begin{bmatrix} 0 \\ \dots \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{matrix} \text{"like"} \end{matrix} \begin{bmatrix} 1 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{matrix} \text{"donuts"} \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

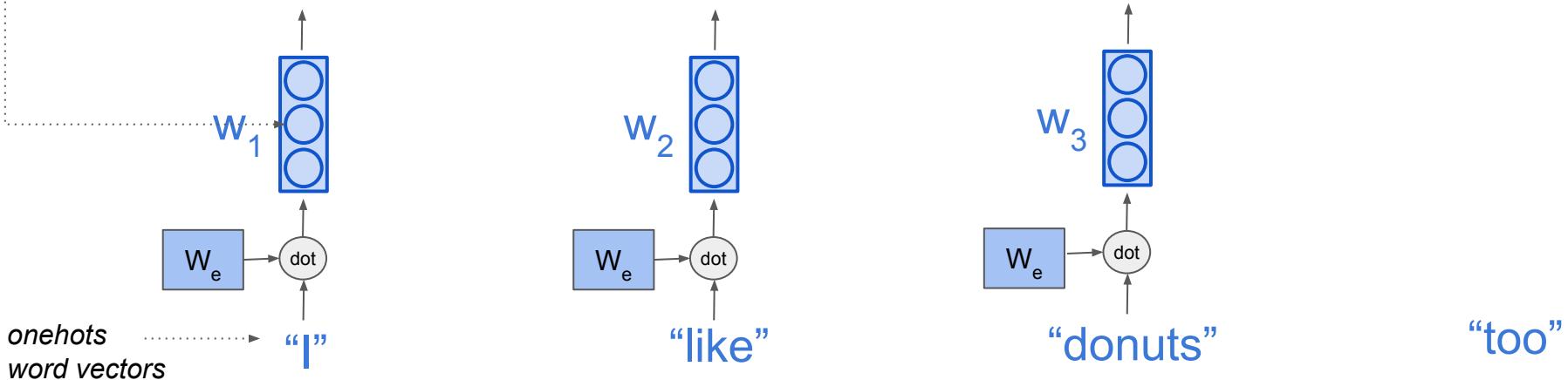
$$\begin{matrix} \text{"too"} \end{matrix} \begin{bmatrix} 0 \\ \dots \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Reminder on Generative RNNs

RNN

generative

non contextual
word embeddings



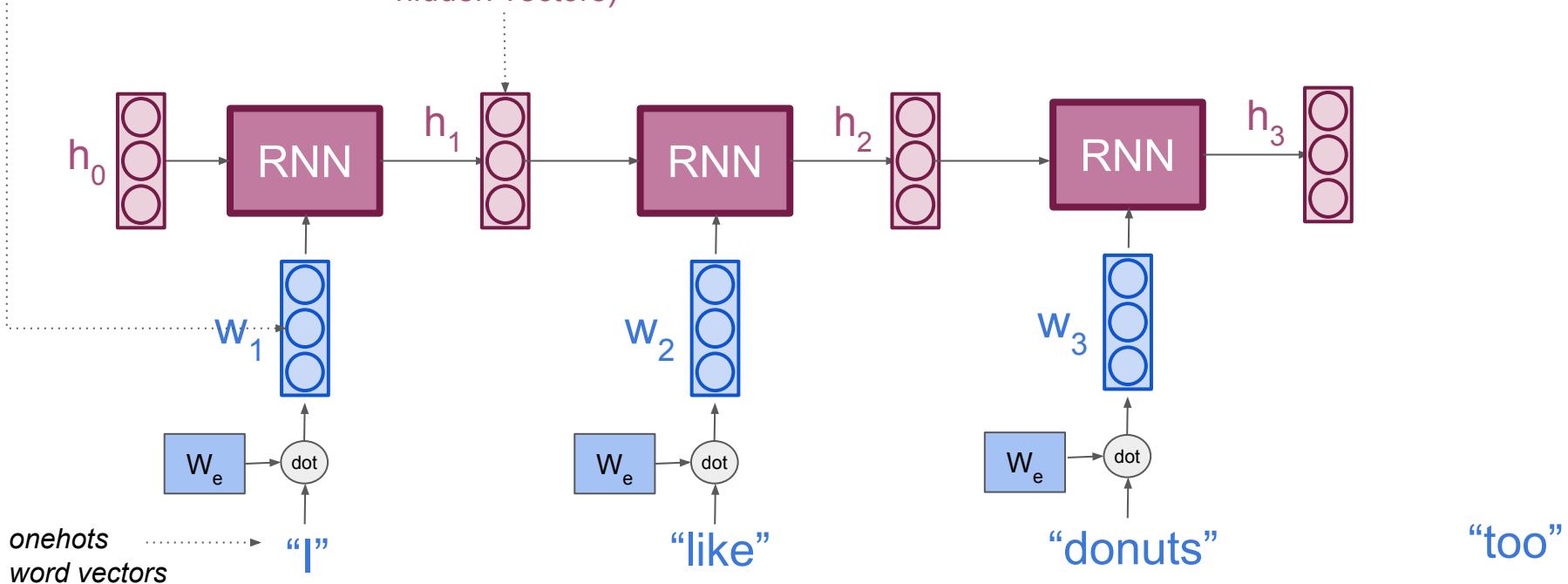
Reminder on Generative RNNs

RNN

generative

non contextual
word embeddings

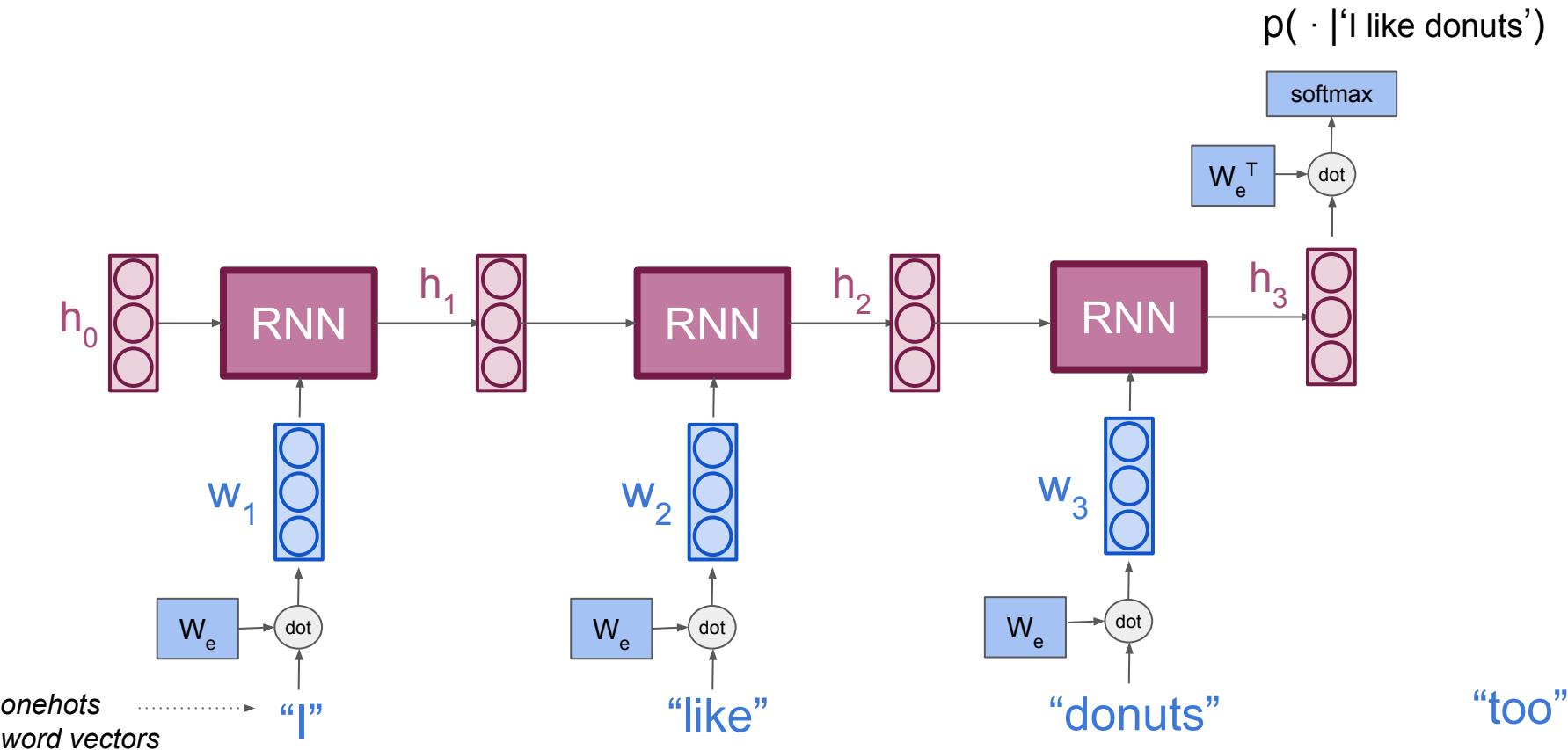
contextual word
embeddings (called the
hidden vectors)



Reminder on Generative RNNs

RNN

generative



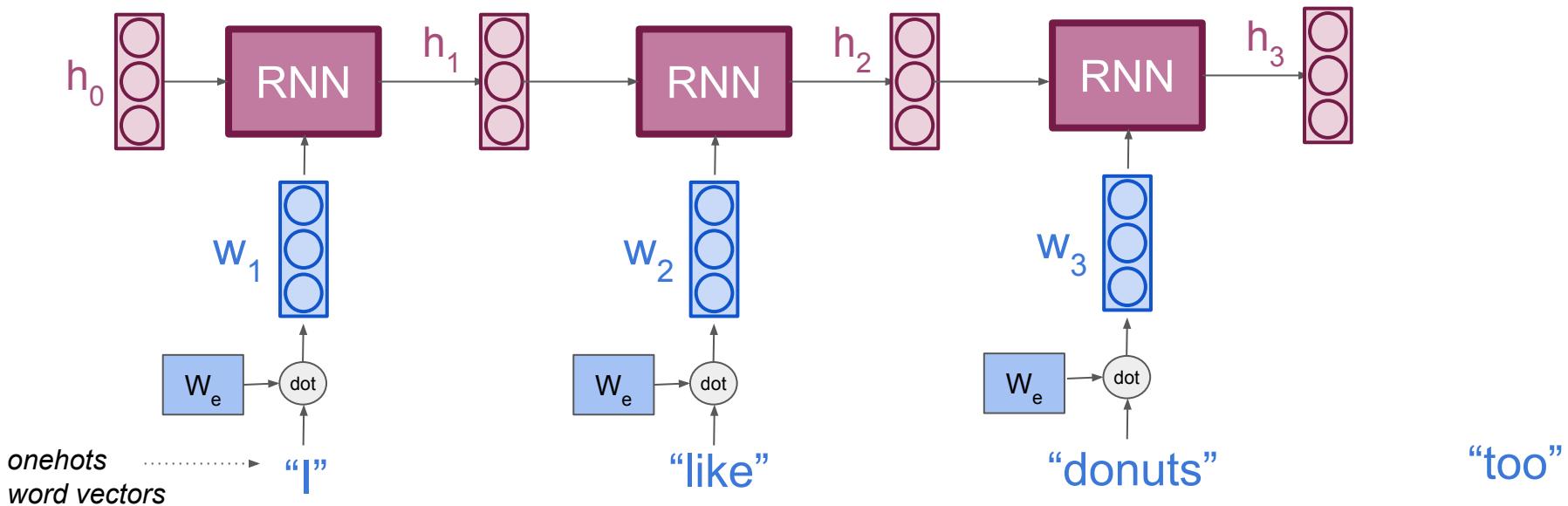
Reminder on Generative RNNs

RNN

generative

Example of the Generative objective:

$$\max \log(p(\text{too} | \text{'I like donuts'}))$$



Reminder on MLM RNNs

RNN

MLM

onehots►
word vectors

“I”

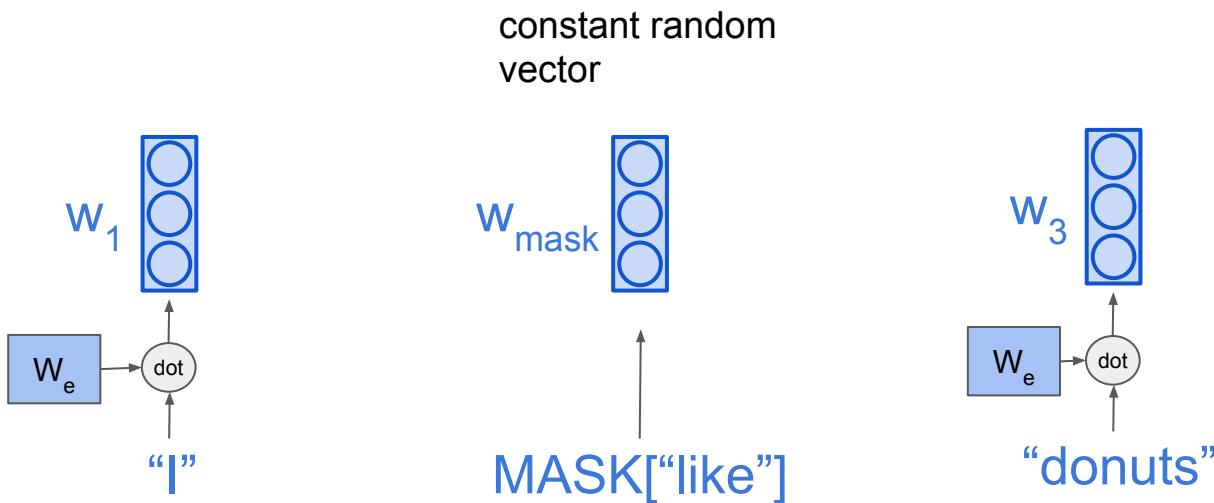
MASK[“like”]

“donuts”

Reminder on MLM RNNs

RNN

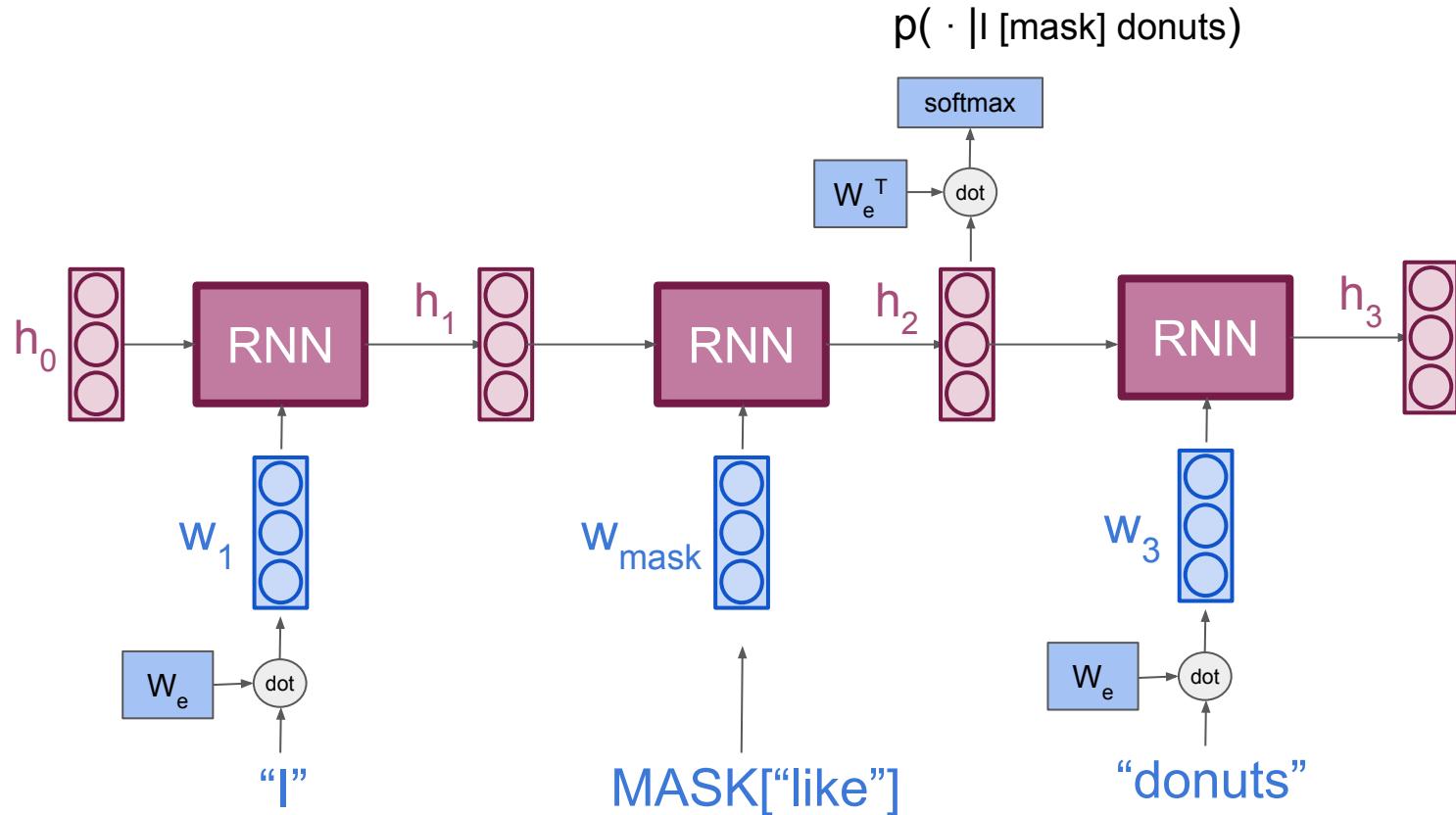
MLM



Reminder on MLM RNNs

RNN

MLM

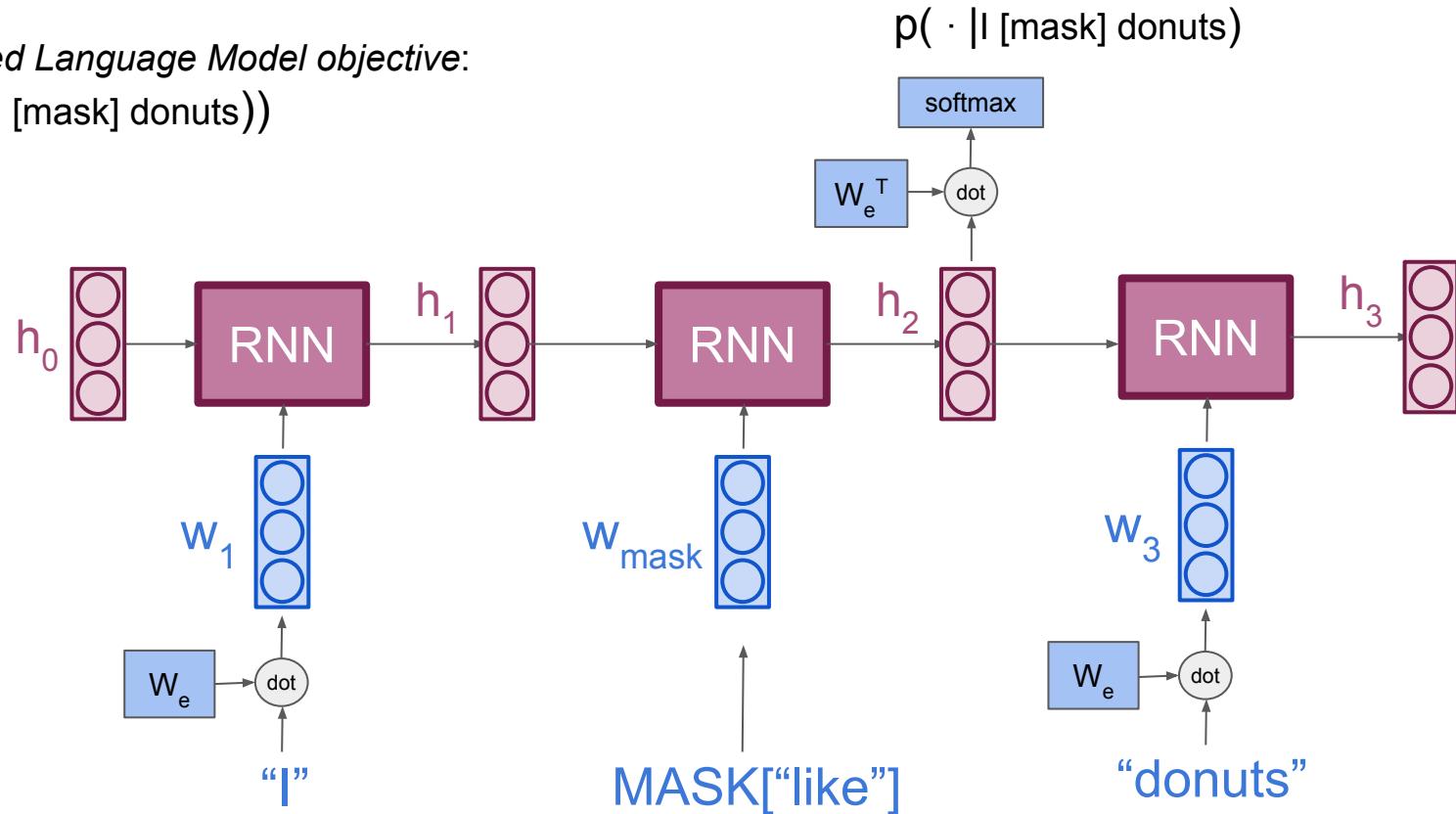


Reminder on MLM RNNs

RNN

MLM

Example of Masked Language Model objective:
 $\max \log(p(\text{like} | \text{I [mask] donuts}))$

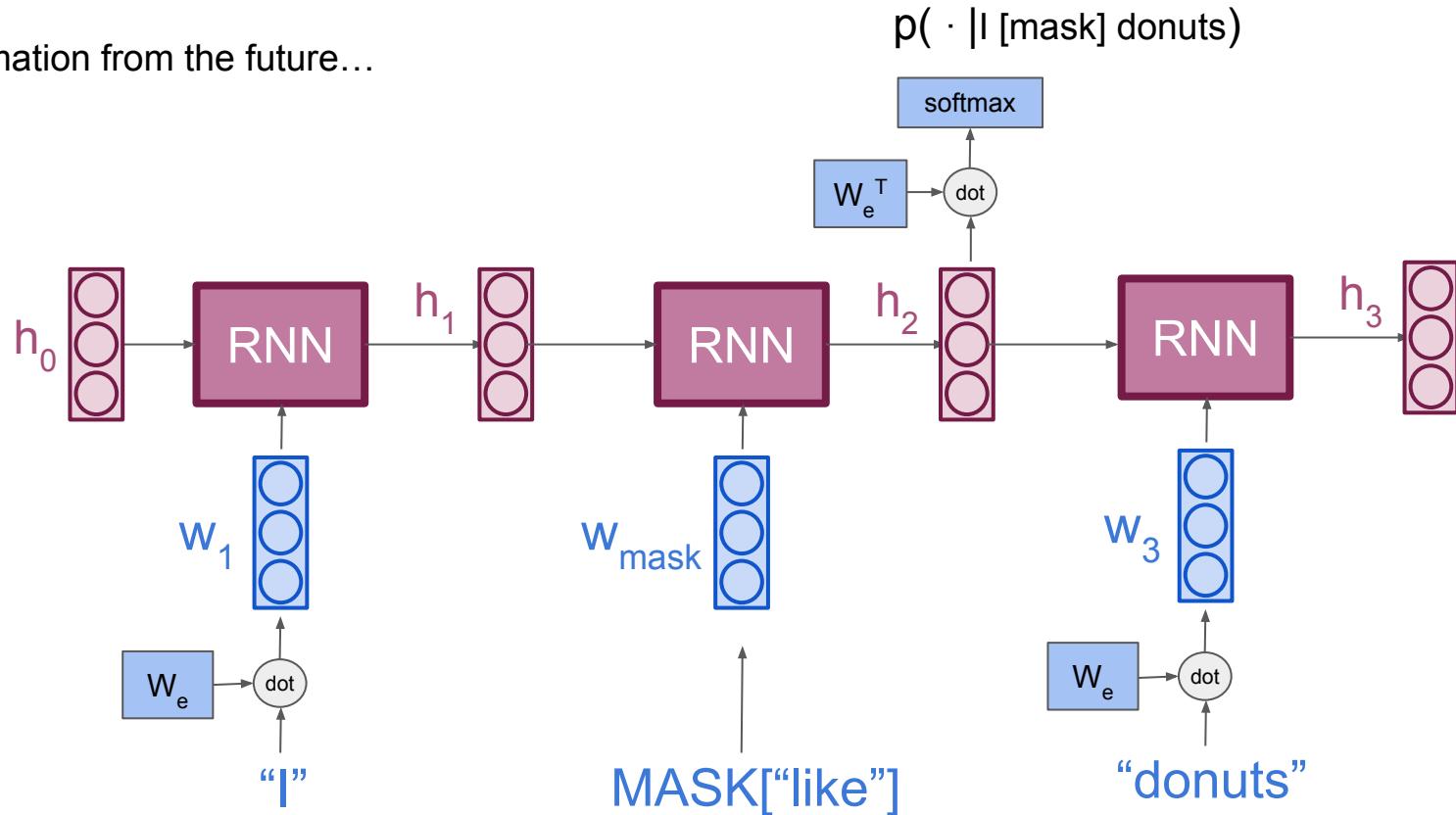


Reminder on MLM RNNs

RNN

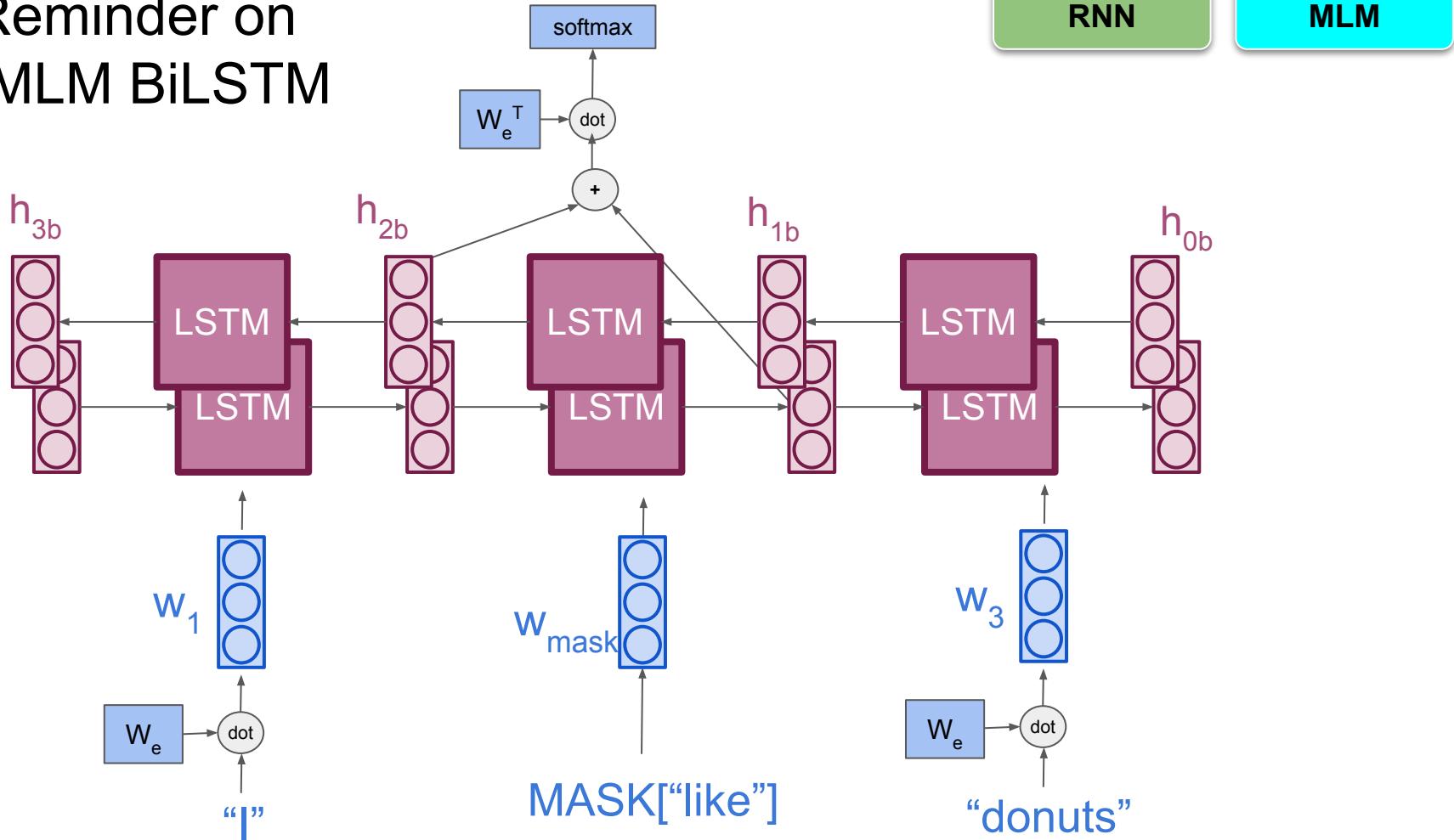
MLM

But we want information from the future...



$p(\cdot | \text{I [mask] donuts})$

Reminder on MLM BiLSTM



RNN

MLM

Transformer Encoder

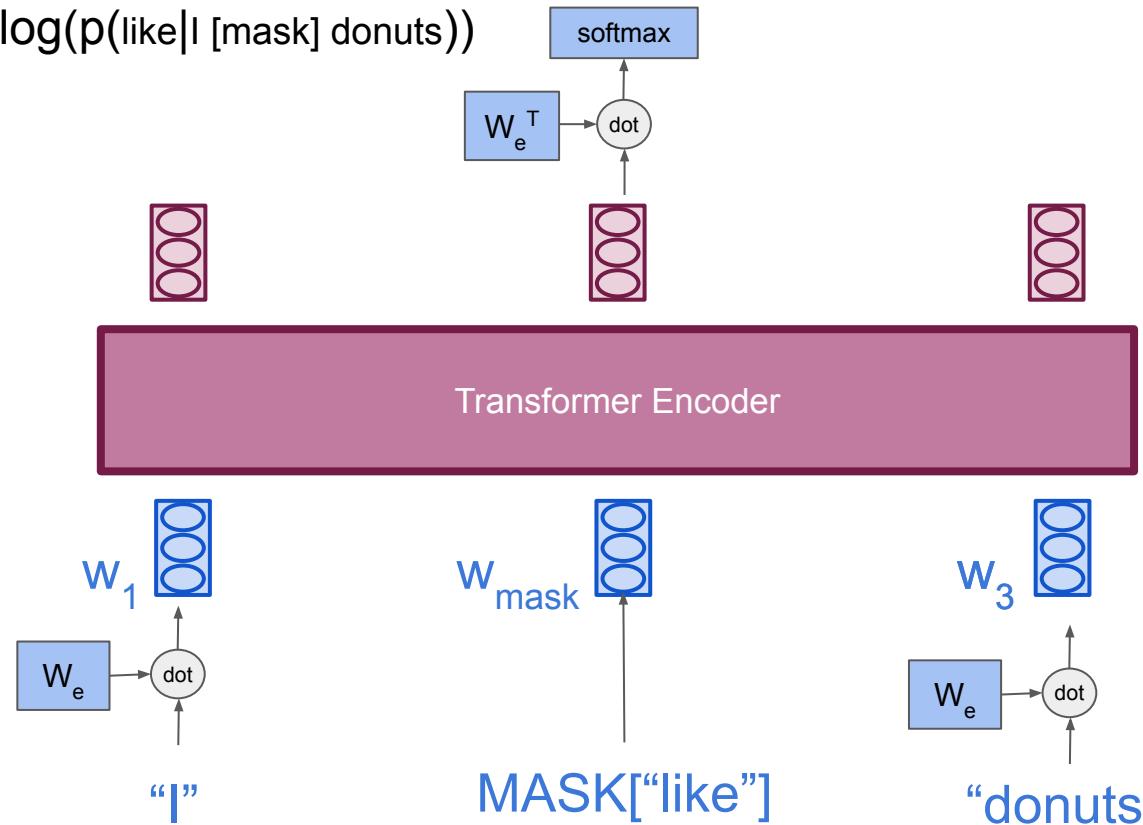
Transformer

MLM

Trained with the MLM objective:

$$\max \log(p(\text{like} | \text{I [mask] donuts}))$$

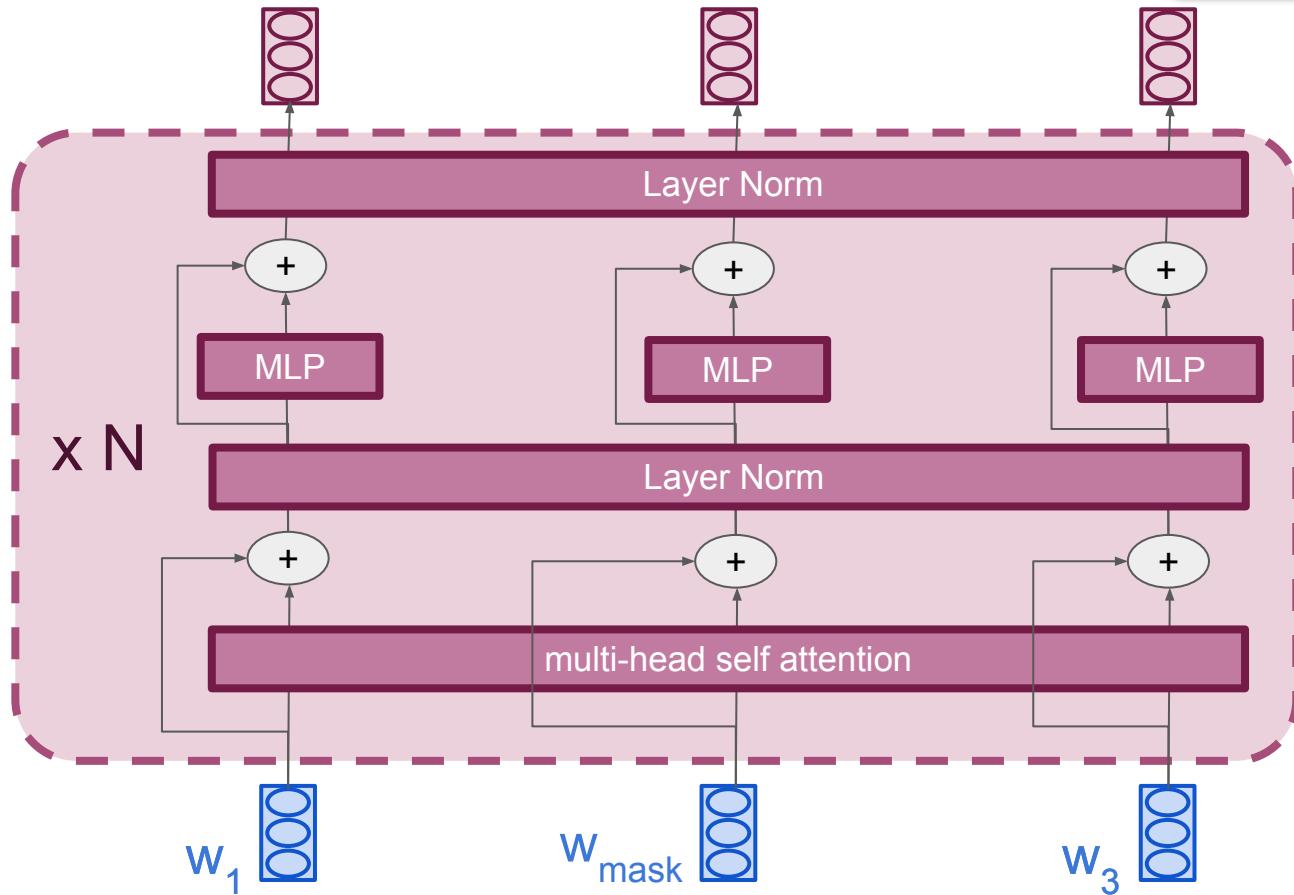
$$p(\cdot | \text{I [mask] donuts})$$



Transformer Encoder

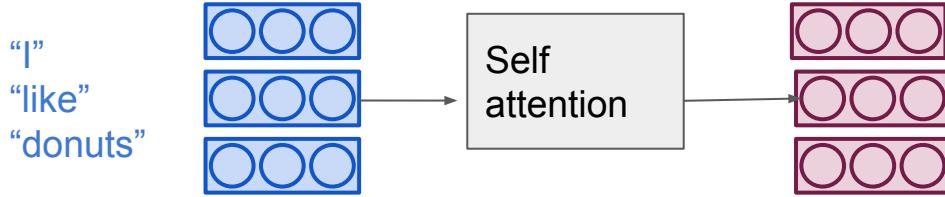
Transformer

MLM



No more causal
architecture. The whole
sentence is processed at
once!

Self-Attention* in 3 steps



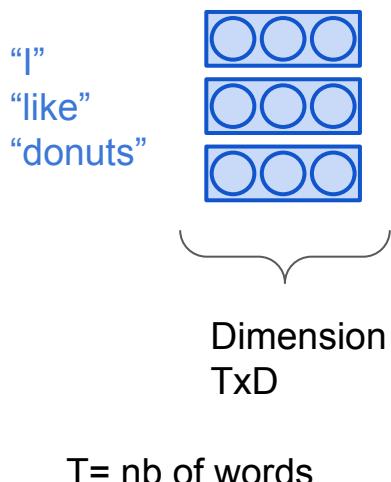
In a sentence, each word should be embedded as a weighted average of all word embeddings from that sentence

Learn the syntactic and semantic dependencies between words

*Vaswani et al. Attention is all you need.

Self-Attention* in 3 steps

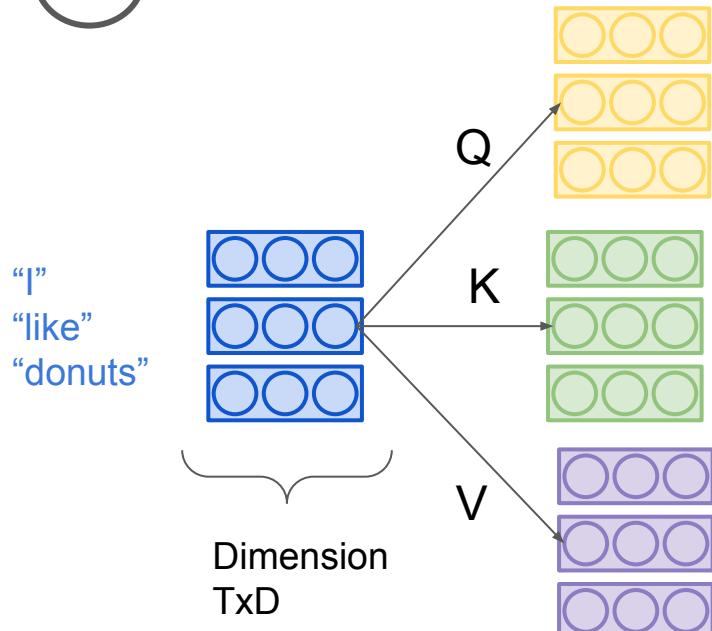
1



Apply 3 fully connected
layers with Query, Key and
Value weight matrices

Self-Attention* in 3 steps

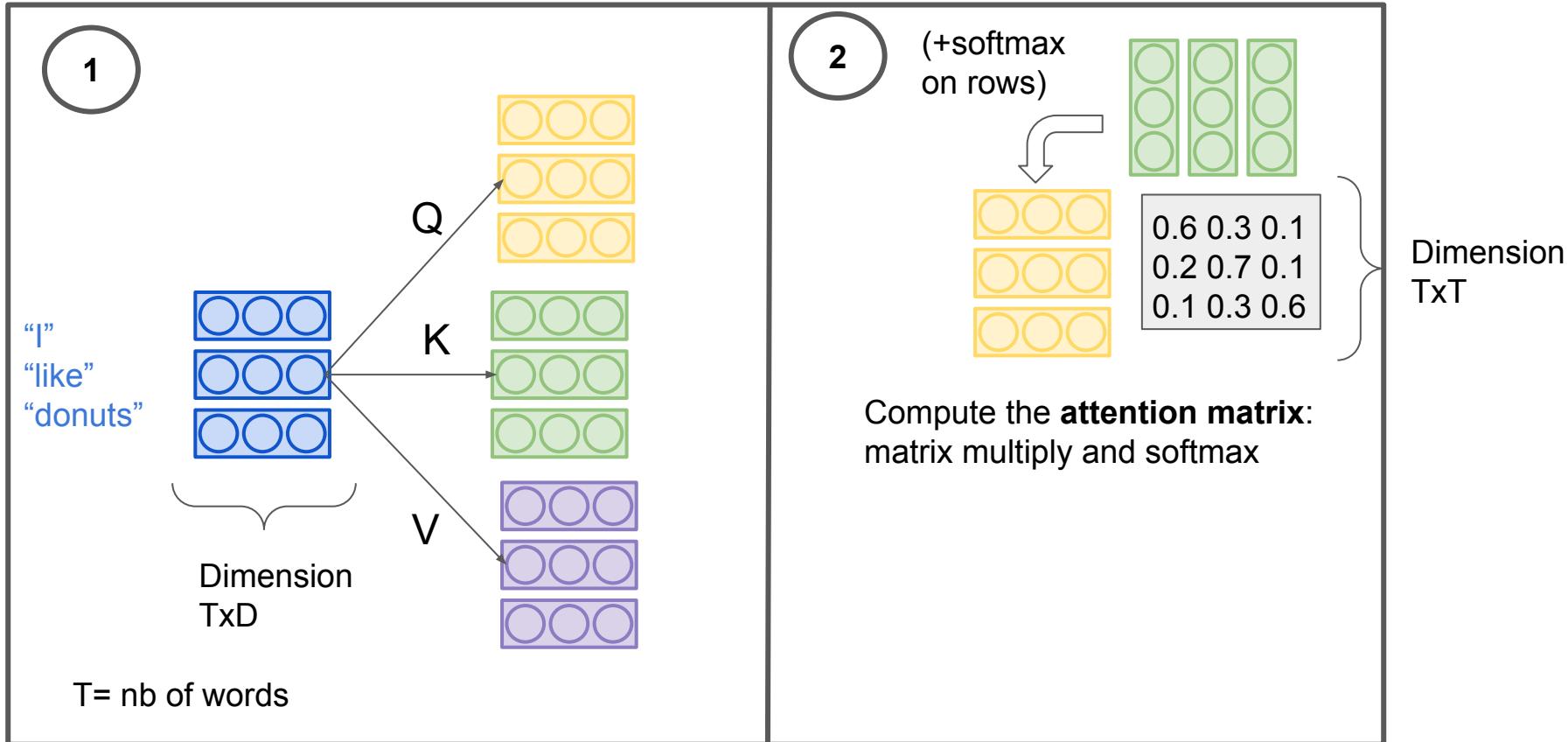
1



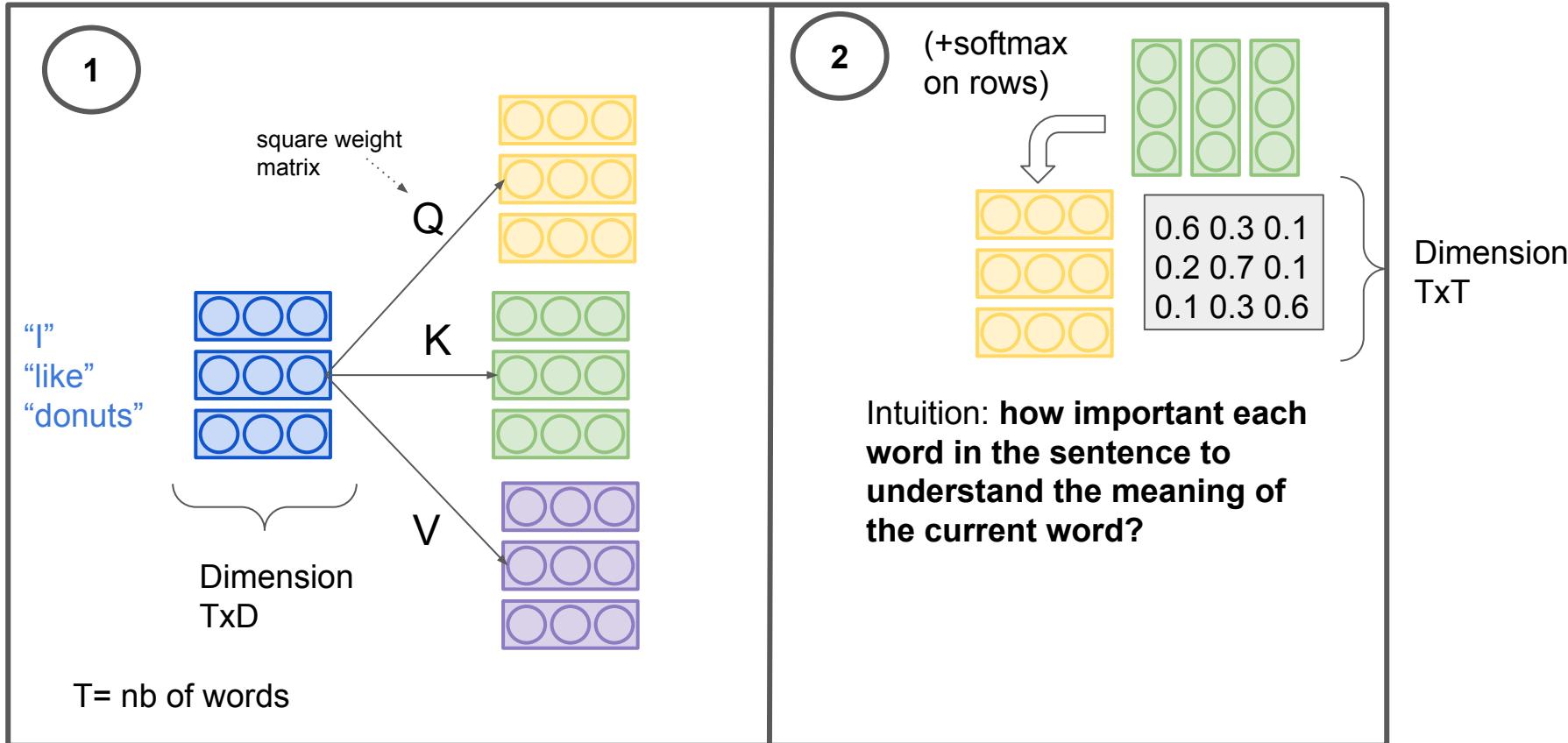
$T = \text{nb of words}$

*Vaswani et al. Attention is all you need.

Self-Attention* in 3 steps

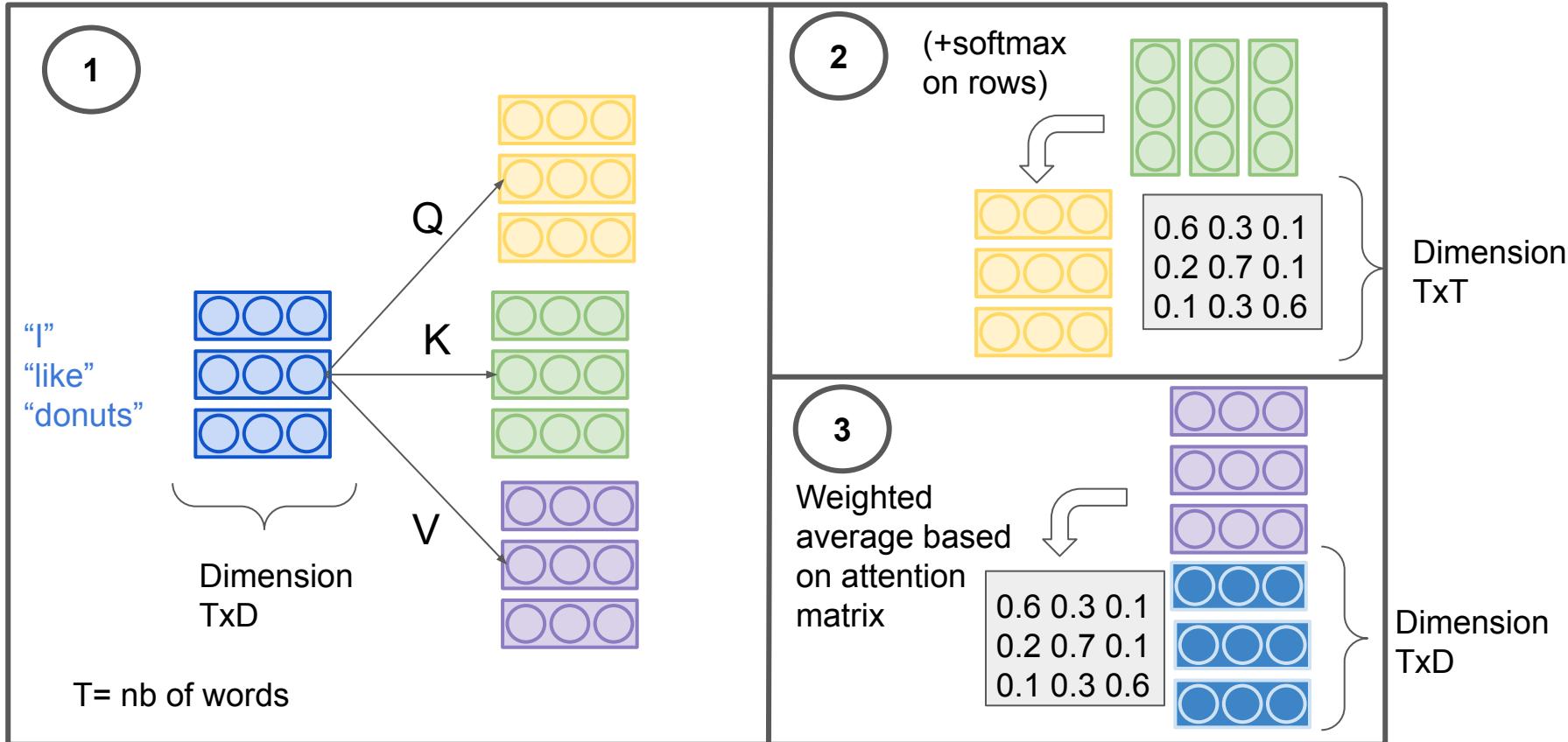


Self-Attention* in 3 steps



*Vaswani et al. Attention is all you need.

Self-Attention* in 3 steps



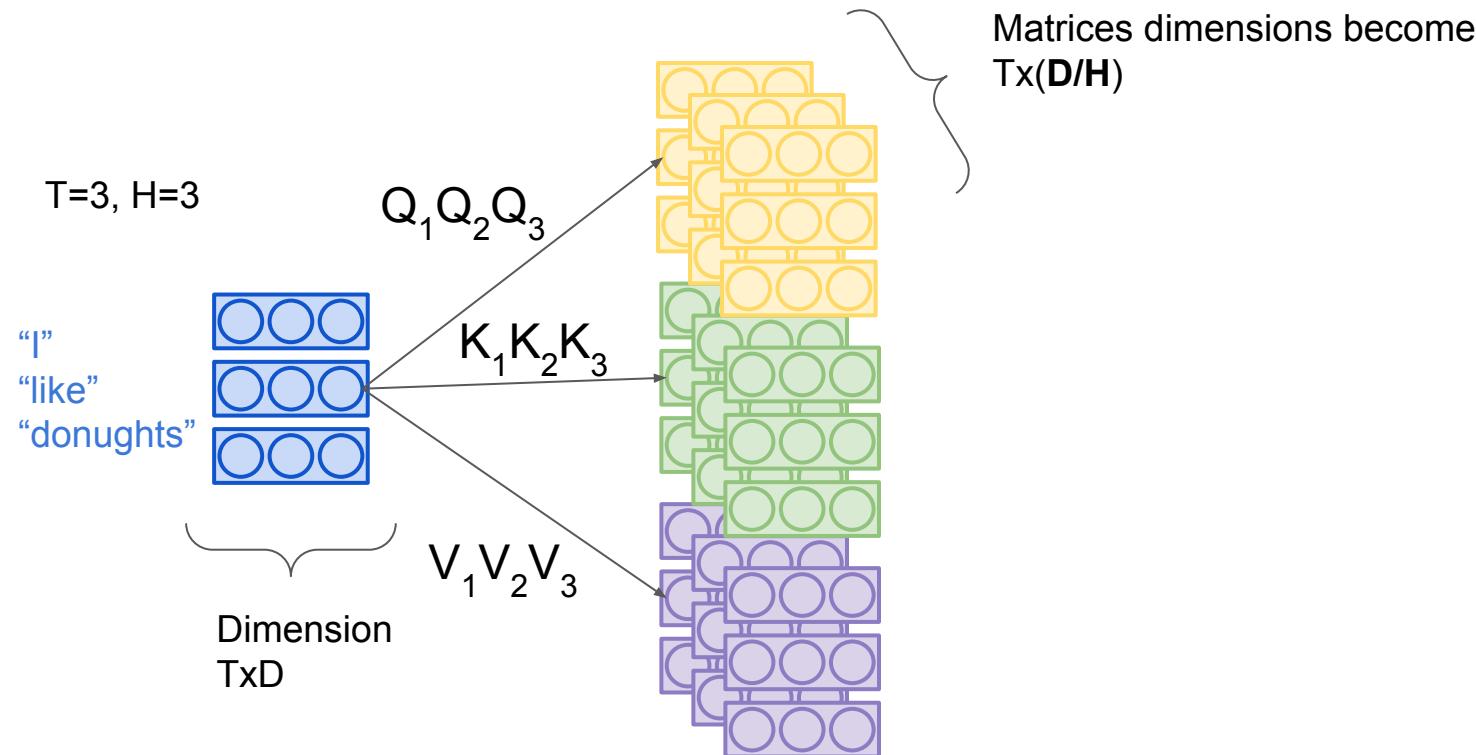
*Vaswani et al. Attention is all you need.

Self-Attention in one formula

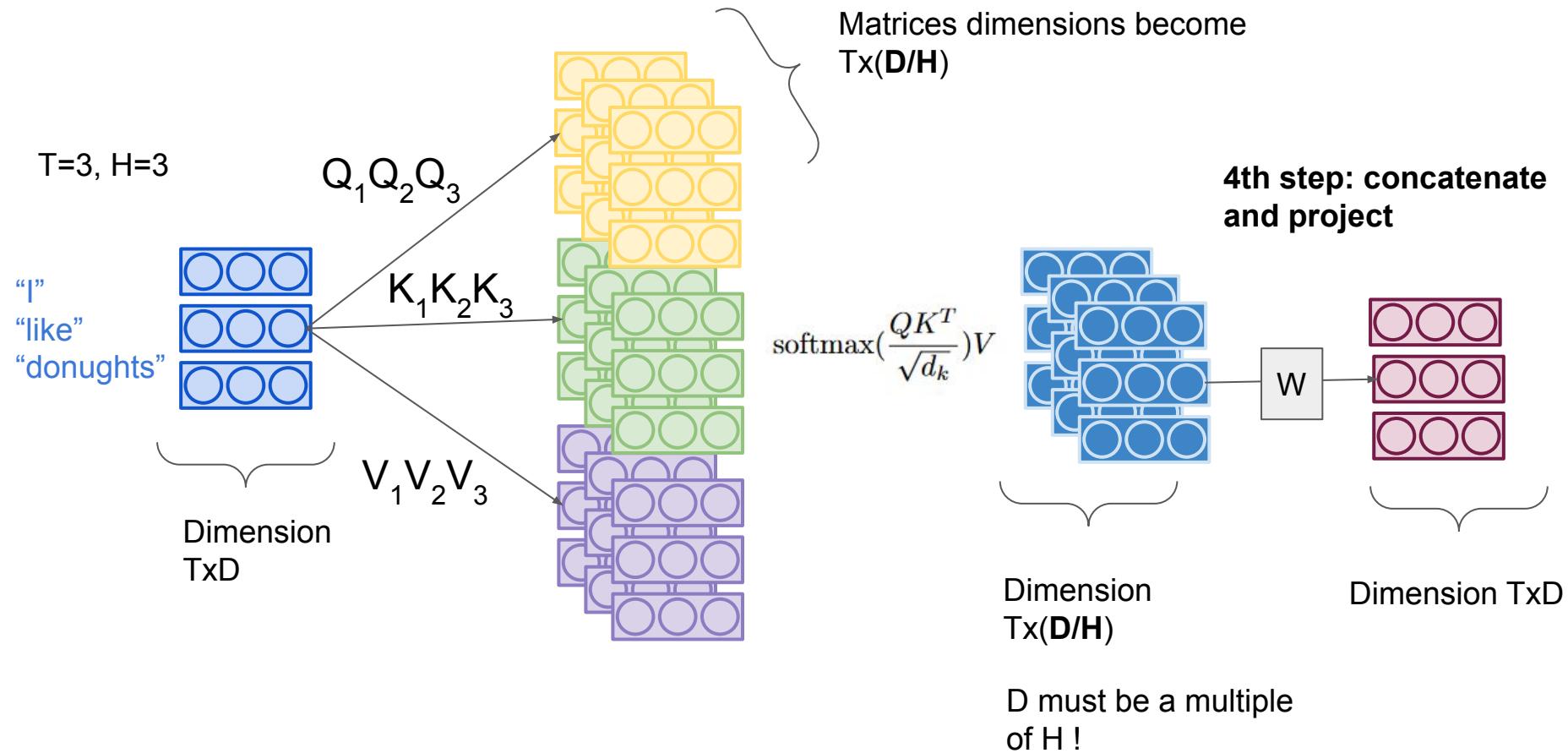
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-head self-attention (4 steps)

Multi-head self-attention (4 steps)



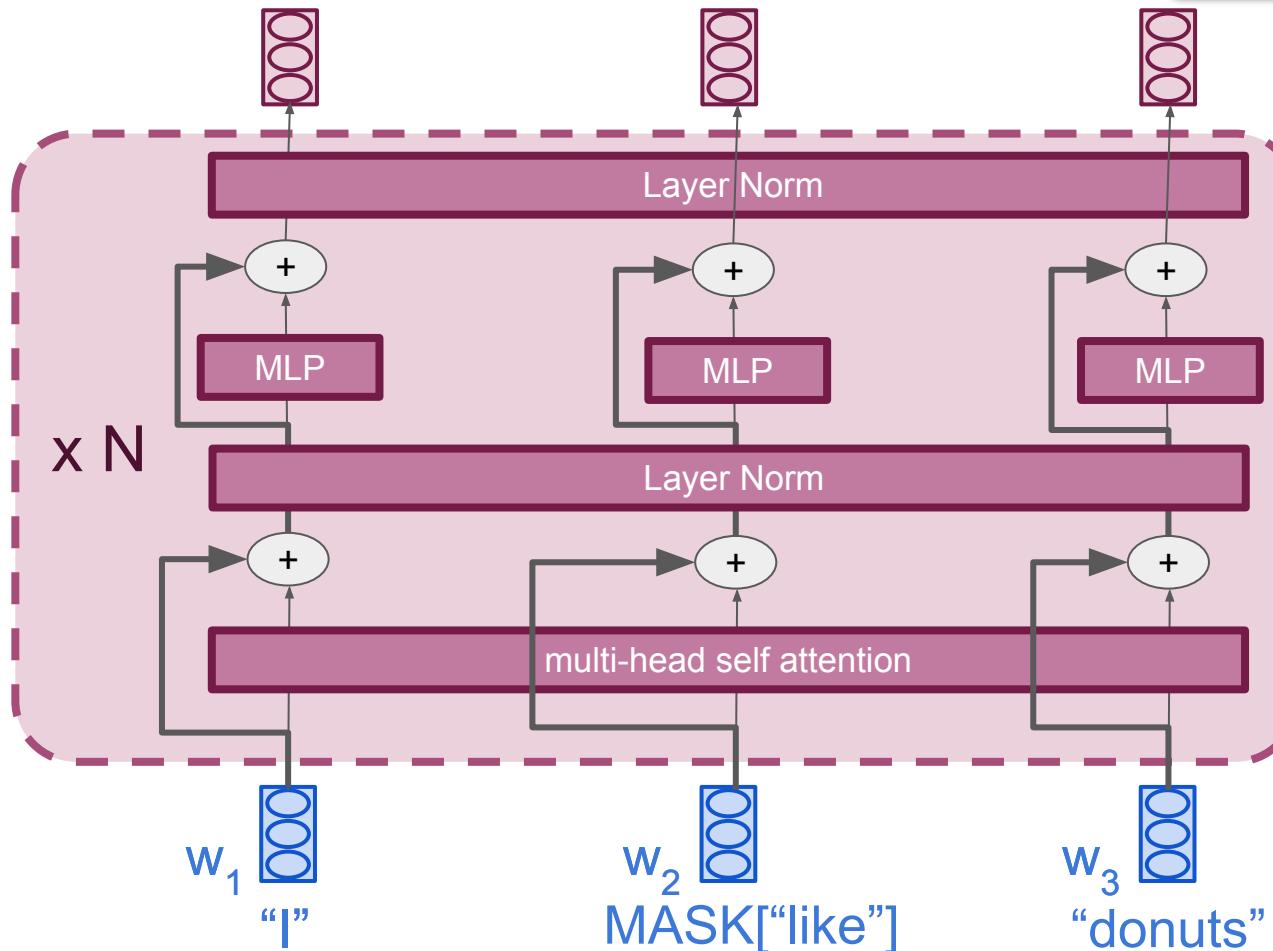
Multi-head self-attention (4 steps)



MLP, skip connections and layer norm

Transformer

MLM

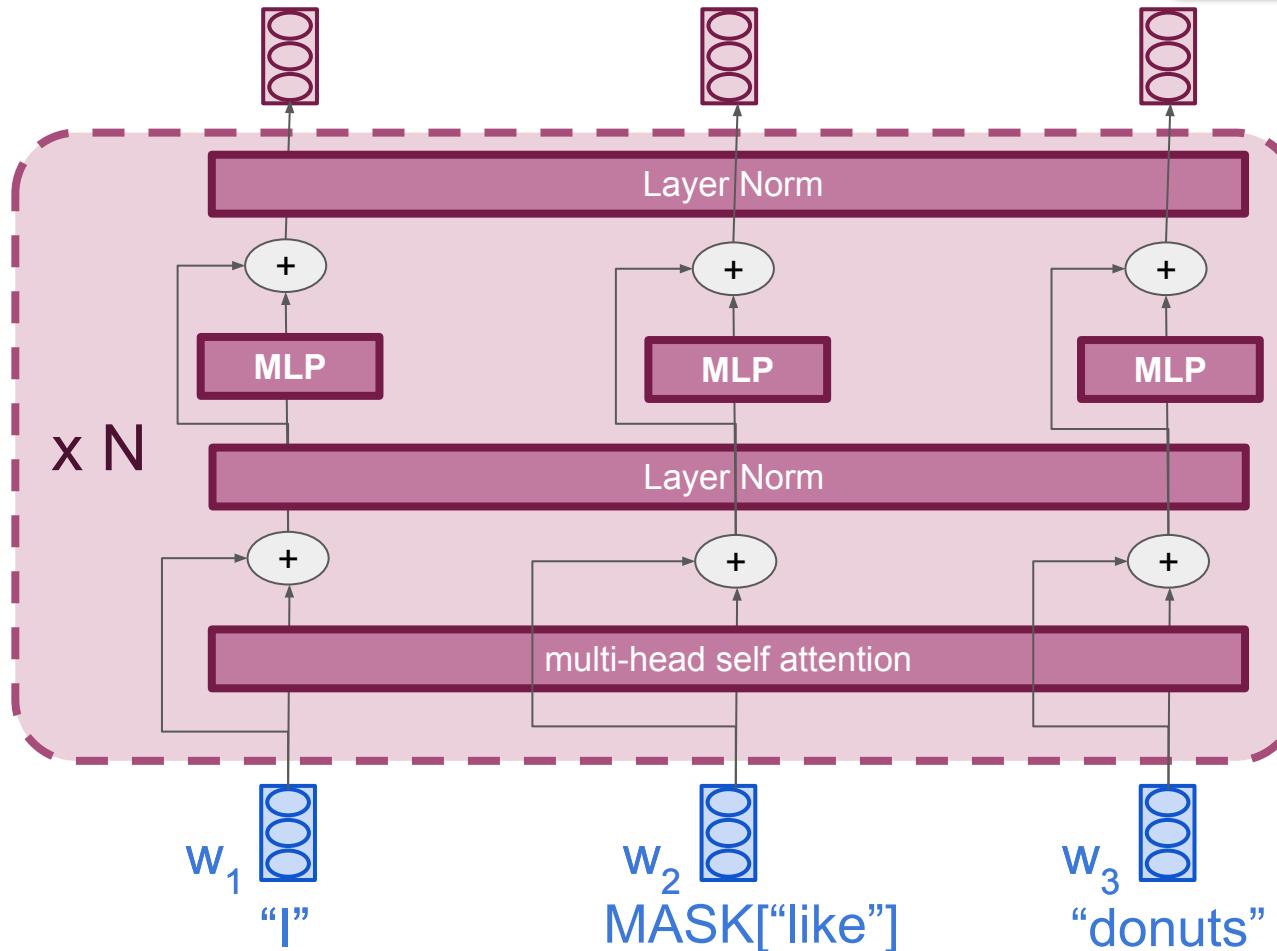


In bold are skip connections:
prevent vanishing gradient.
Necessary in all deep architectures.

MLP, skip connections and layer norm

Transformer

MLM

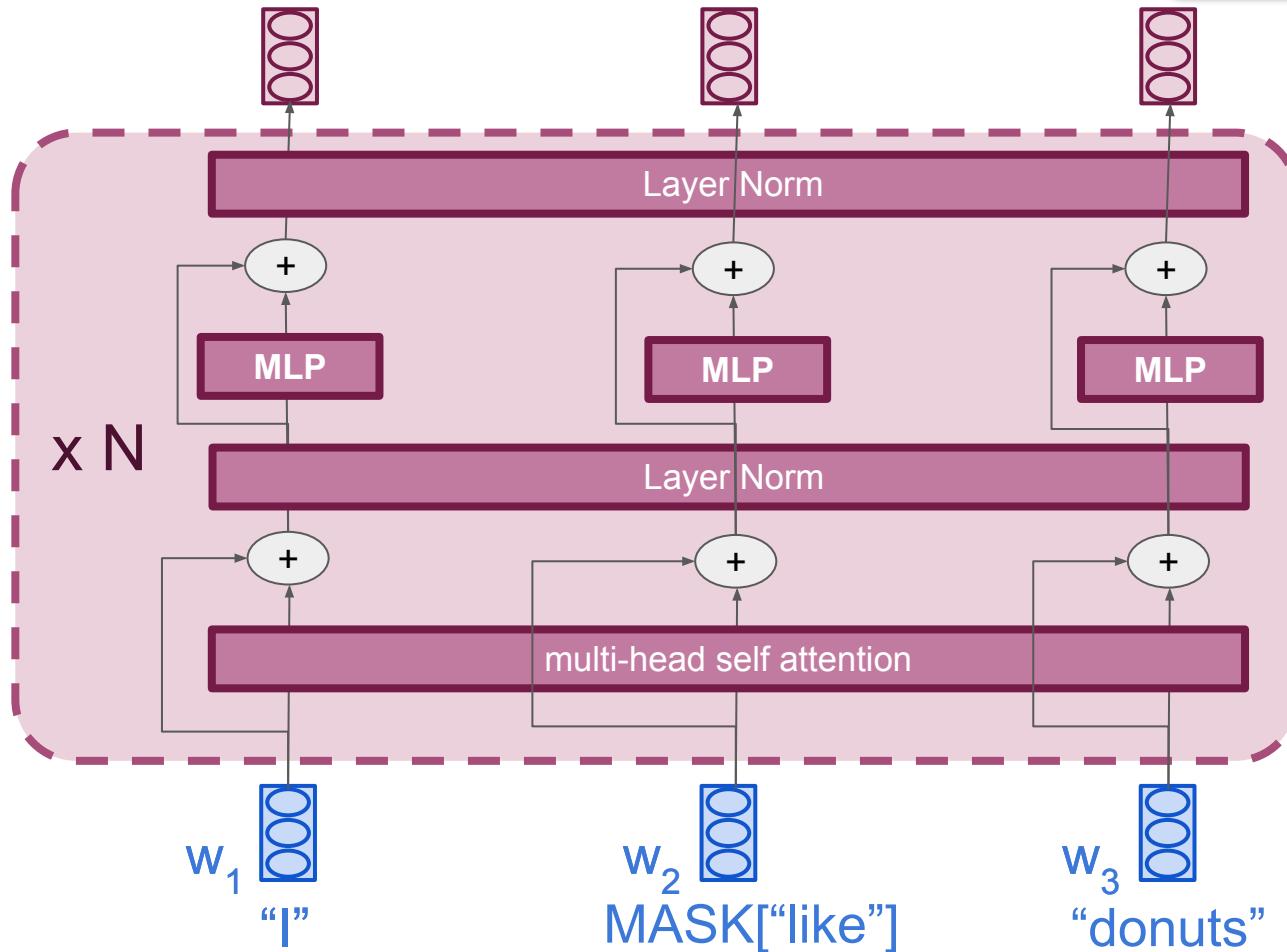


MLP have two
layers with ReLU

MLP, skip connections and layer norm

Transformer

MLM

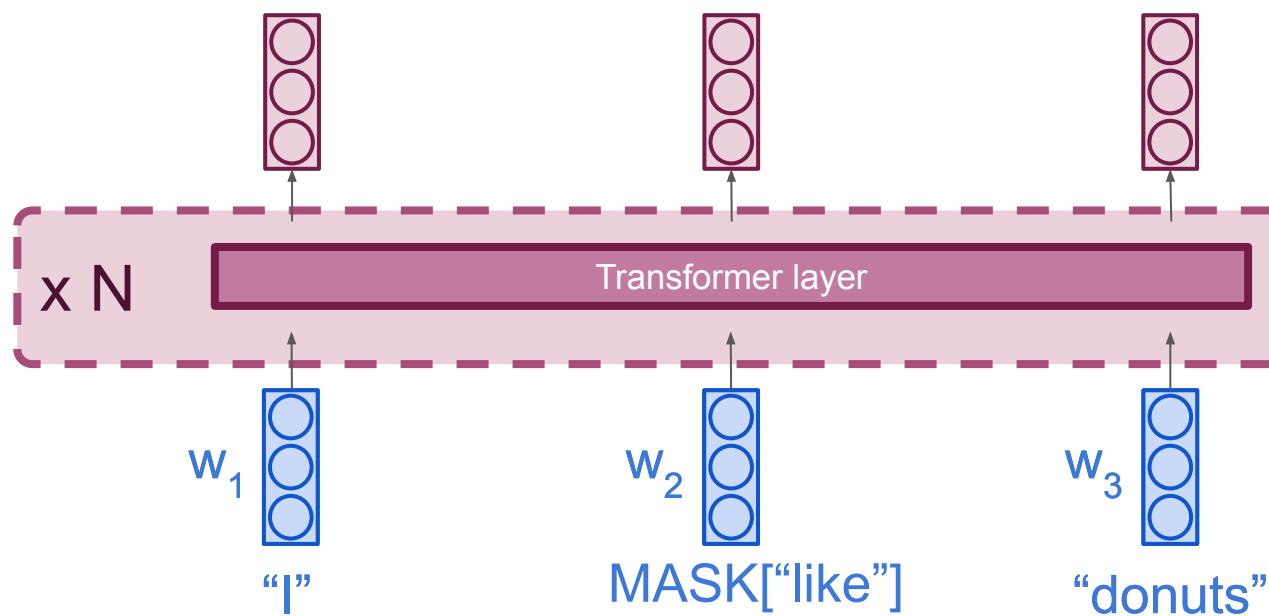


LayerNorm: the batch norm of NLP

How is word order encoded?

Transformer

MLM

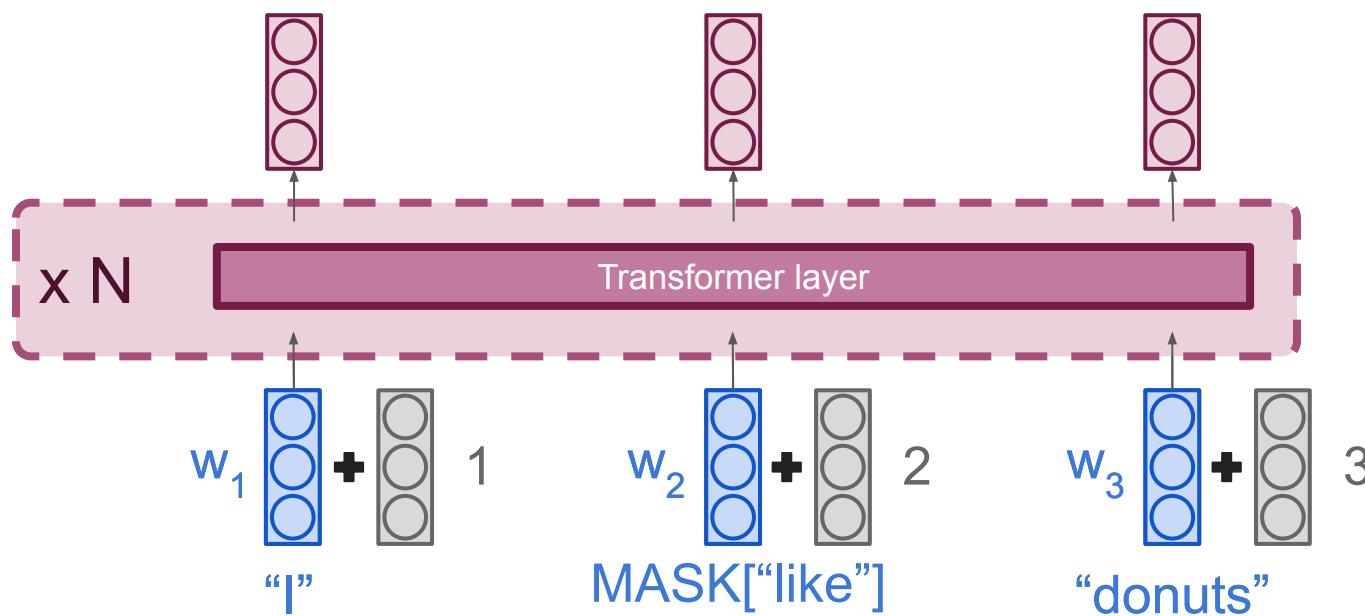


How is word order encoded?

Transformer

MLM

Positional encoding !



Positional embeddings: not that easy

- Unique embedding for each time-step
- The values should be bounded (no indices)
- The distance between pos-embeddings should decay with time
- Distance between two pos-embeddings should be consistent across sentences with different lengths (cannot use a normalised indices)
- Should generalize to longer sentences than seen during training

Example: sinusoidal positional encoding

$$\vec{p}_t = \begin{bmatrix} \sin(\omega_1 \cdot t) \\ \cos(\omega_1 \cdot t) \\ \\ \sin(\omega_2 \cdot t) \\ \cos(\omega_2 \cdot t) \\ \vdots \\ \sin(\omega_{d/2} \cdot t) \\ \cos(\omega_{d/2} \cdot t) \end{bmatrix}_{d \times 1}$$
$$\omega_k = \frac{1}{10000^{2k/d}}$$

Required: $t \ll T_{\max}$ (here $T_{\max} = 10000$)

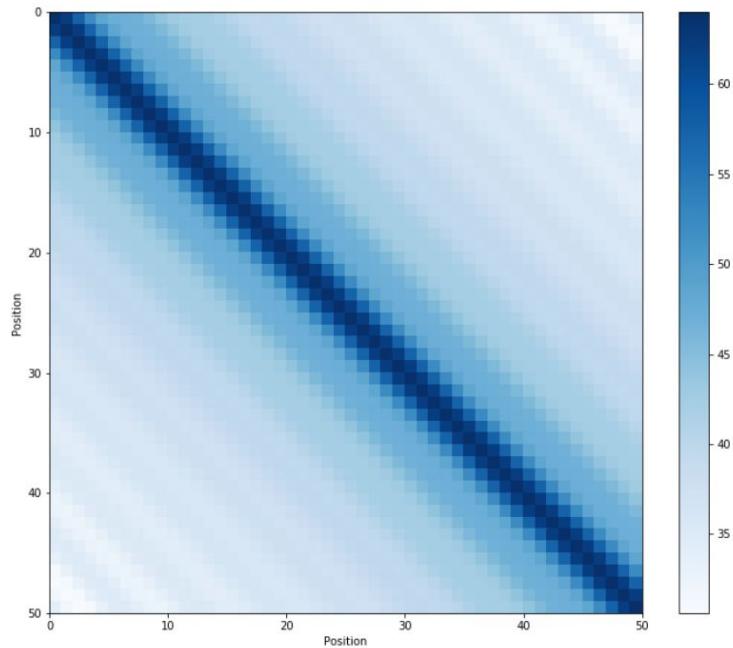
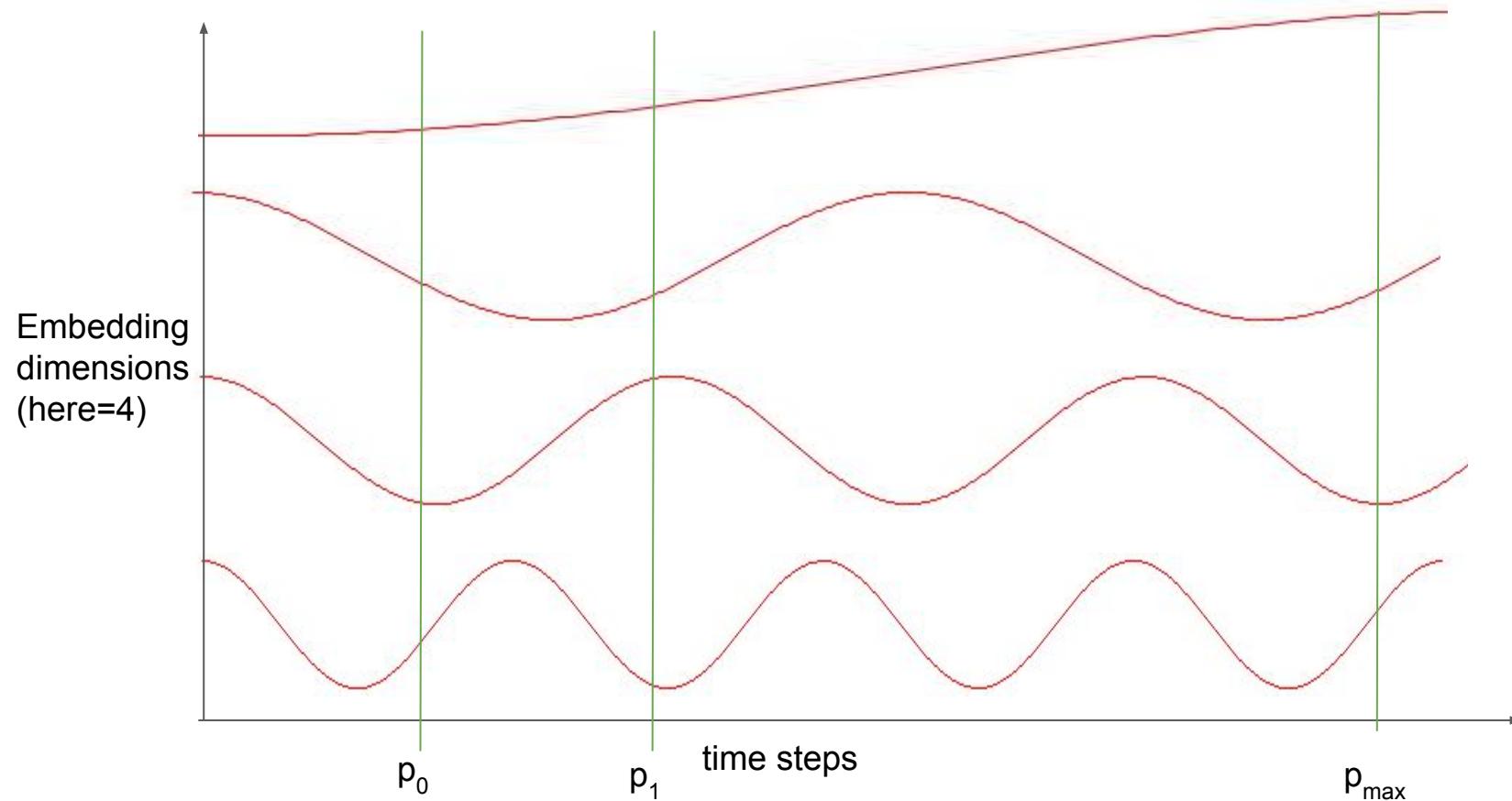


Figure 3 - Dot product of position embeddings for all time-steps

Visual interpretation



Positional encoding

If you use the Pytorch implementation, you need to implement it yourself
(huggingface has already included it)

```
class PositionalEncoding(nn.Module):

    def __init__(self, d_model: int, dropout: float = 0.1, max_len: int = 5000):
        super().__init__()
        self.dropout = nn.Dropout(p=dropout)

        position = torch.arange(max_len).unsqueeze(1)
        div_term = torch.exp(torch.arange(0, d_model, 2) * (-math.log(10000.0) / d_model))
        pe = torch.zeros(max_len, 1, d_model)
        pe[:, 0::2] = torch.sin(position * div_term)
        pe[:, 1::2] = torch.cos(position * div_term)
        self.register_buffer('pe', pe)

    def forward(self, x: Tensor) -> Tensor:
        """
        Args:
            x: Tensor, shape [seq_len, batch_size, embedding_dim]
        """
        x = x + self.pe[:x.size(0)]
        return self.dropout(x)
```

https://pytorch.org/tutorials/beginner/transformer_tutorial.html

Positional Embeddings

Can also be learned with convolution layers, embedding matrix,...

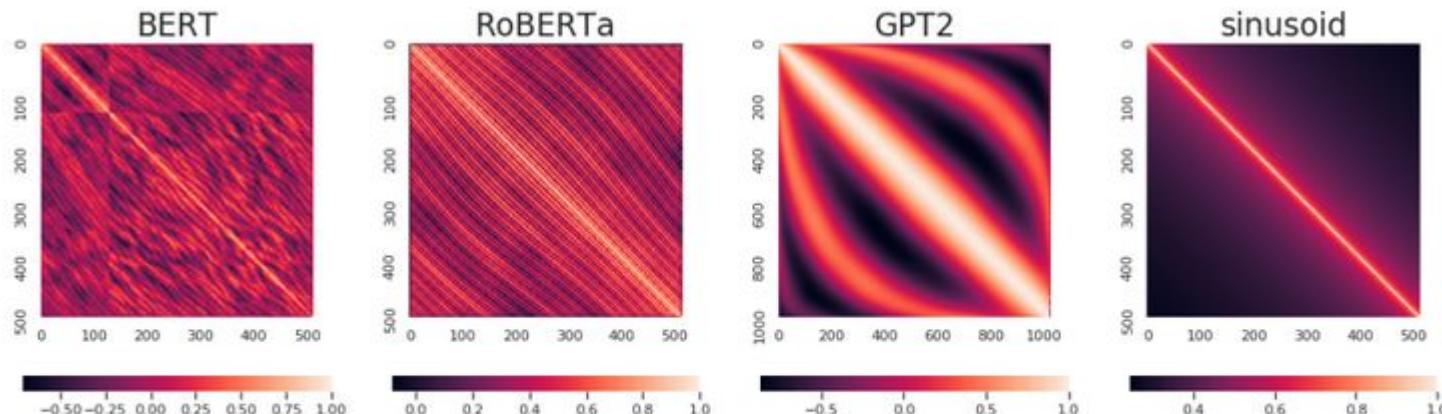


Figure 1: Visualization of position-wise cosine similarity of different position embeddings. Lighter in the figures denotes the higher similarity.

Number of operations in a one layer transformer encoder ?

For a sentence of n words, and a transformer encoder of embedding dimension d

$$\begin{aligned}\phi_{MSA}(n, d) &= \phi_{qkv}(n, d) + \phi_A(n, d) + \phi_O(n, d) + \phi_{proj}(n, d) \\ &= 3nd^2 + n^2d + n^2d + nd^2 \\ &= 4nd^2 + 2n^2d.\end{aligned}$$

(multi-head self-attention is computed in four steps)

$$\phi_{MLP}(n, d) = \phi_{fc1}(n, d) + \phi_{fc2}(n, d) = 4nd^2 + 4nd^2 = 8nd^2.$$

(Fc1: D->4D and Fc2: 4D->D)

$$\phi_{BLK}(n, d) = \phi_{MSA}(n, d) + \phi_{MLP}(n, d) = 12nd^2 + 2n^2d.$$

(total number of operation)

So where is the bottleneck ? sentence length or embedding dimension?

Number of operations in a one layer transformer encoder ?

$$\begin{aligned}\phi_{MSA}(n, d) &= \phi_{qkv}(n, d) + \phi_A(n, d) + \phi_O(n, d) + \phi_{proj}(n, d) \\ &= 3nd^2 + n^2d + n^2d + nd^2 \\ &= 4nd^2 + 2n^2d.\end{aligned}$$

(multi-head self-attention is computed in four steps, see previous slide)

$$\phi_{MLP}(n, d) = \phi_{fc1}(n, d) + \phi_{fc2}(n, d) = 4nd^2 + 4nd^2 = 8nd^2. \quad (\text{Fc1: D} \rightarrow 4\text{D} \text{ and Fc2: } 4\text{D} \rightarrow \text{D})$$

$$\phi_{BLK}(n, d) = \phi_{MSA}(n, d) + \phi_{MLP}(n, d) = 12nd^2 + 2n^2d. \quad (\text{total number of operation})$$

So where is the bottleneck ? sentence length or embedding dimension?
It depends! Most of the time, $n \gg d$ so n is the bottleneck
But Large LM have $d=10^4$, so if $n=1000$, d is the bottleneck

Transformer Decoder

Transformer Encoder = a transformer used in MLM mode

Transformer Decoder = a transformer used in Generative mode

How to do that ?

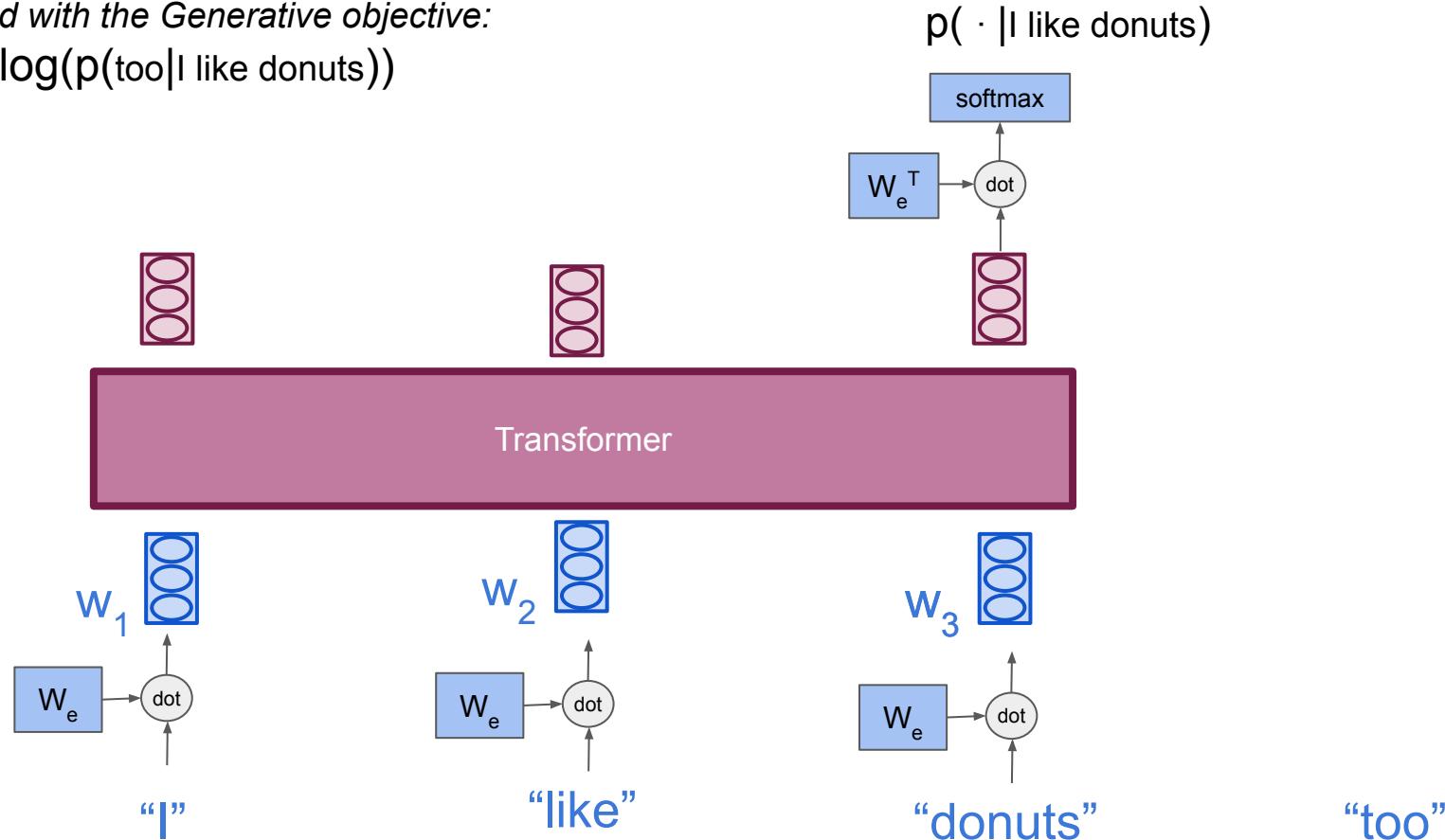
Same architecture as encoder with one **small** yet **crucial** change

Transformer Decoder

Transformer

generative

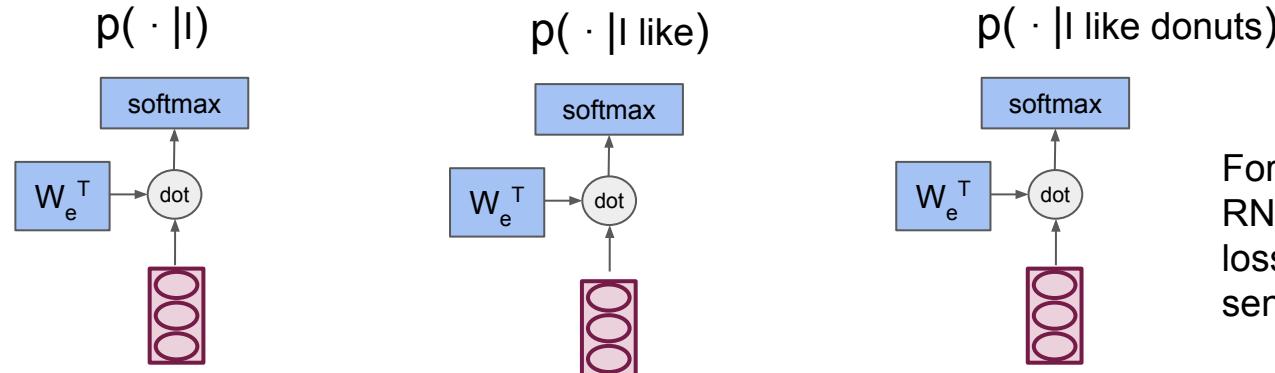
Trained with the Generative objective:
 $\max \log(p(\text{too} | \text{I like donuts}))$



Transformer Decoder

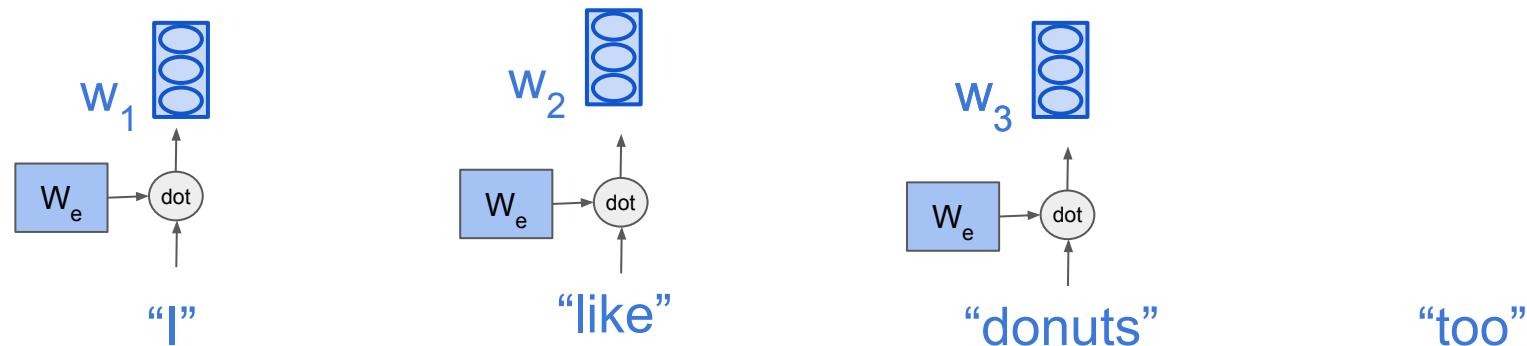
Transformer

generative



For faster training, like for RNNs, we compute the loss over all words of the sentence at once.

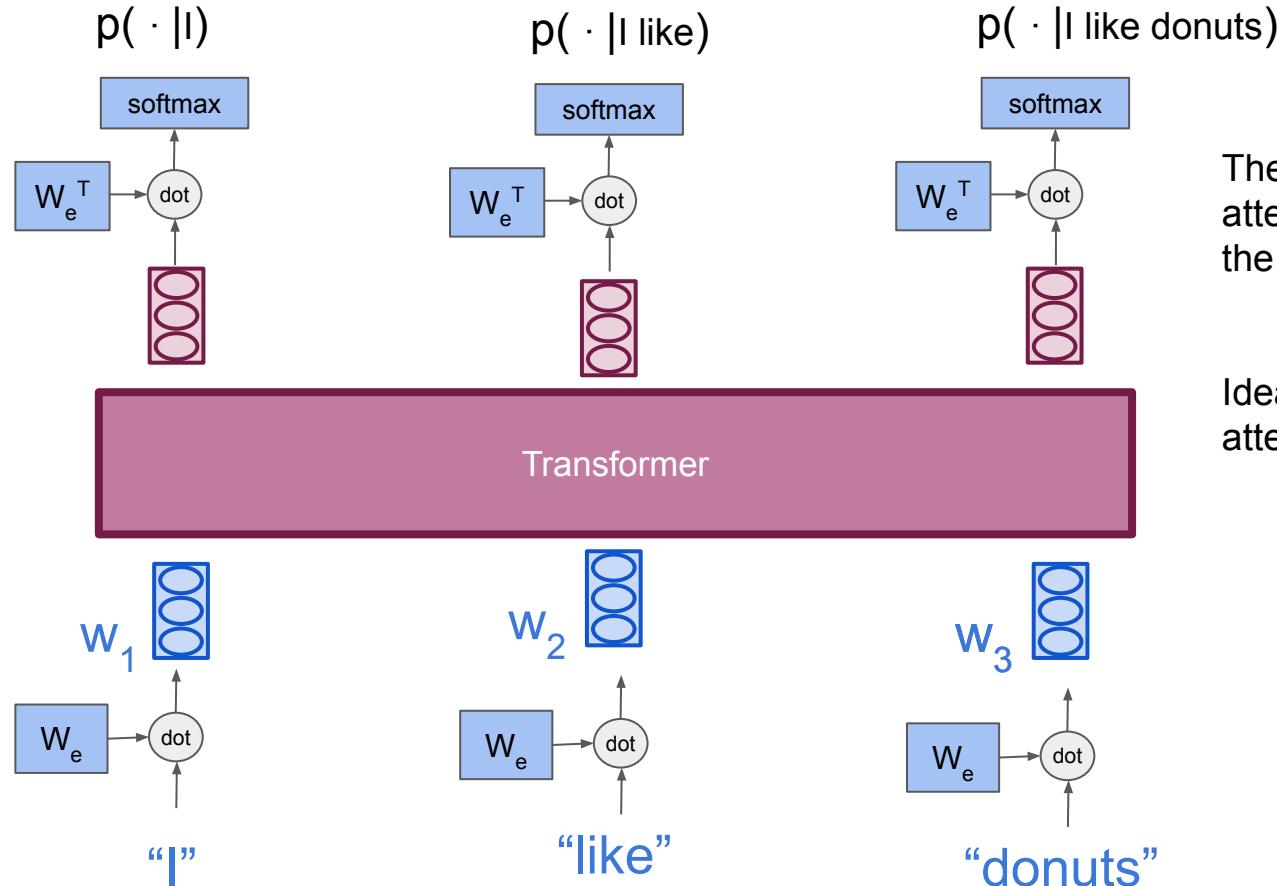
But then we have a problem right?



Transformer Decoder

Transformer

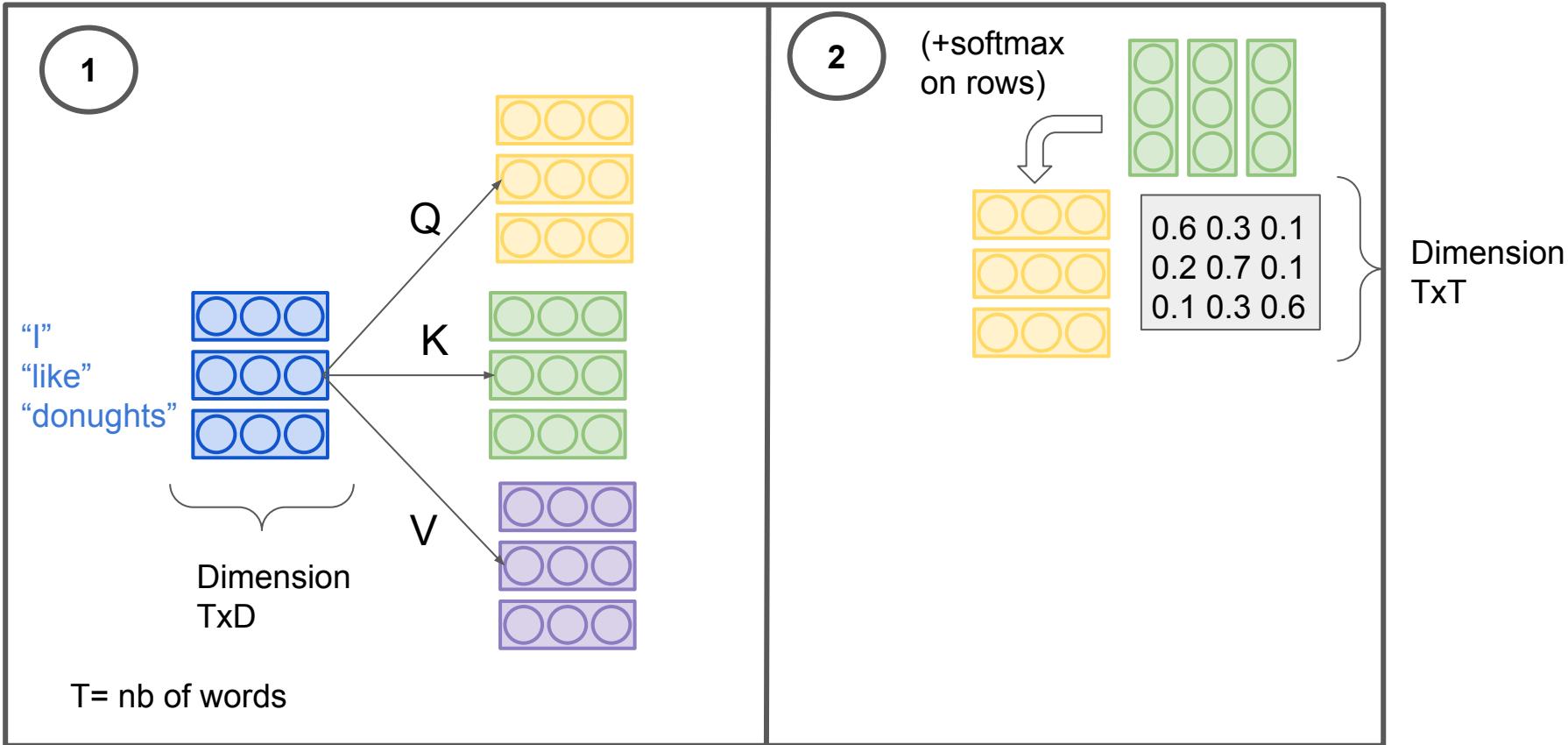
generative



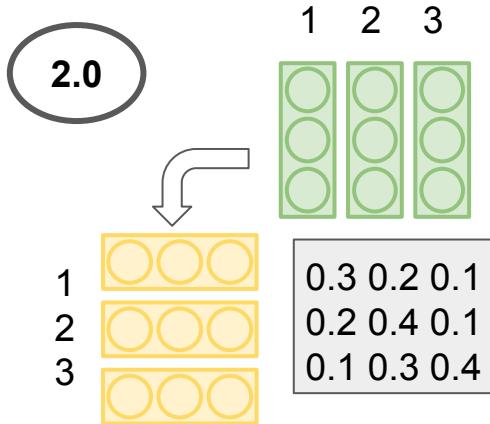
The model can cheat by attending to the future of the sentence !

Idea: causal mask on attention !

Self-Attention in 3 steps



How to make a transformer causal ? Modify step 2



2.1

Causal mask on
attention matrix !

$$\begin{bmatrix} 0.3 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.1 \\ 0.1 & 0.3 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} 1 & -\inf & -\inf \\ 1 & 1 & -\inf \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.3 & -\inf & -\inf \\ 0.2 & 0.4 & -\inf \\ 0.1 & 0.3 & 0.4 \end{bmatrix}$$

Item-wise
multiply

2.2

SoftmaxPerRow(
 $\begin{bmatrix} 0.3 & -\inf & -\inf \\ 0.2 & 0.4 & -\inf \\ 0.1 & 0.3 & 0.4 \end{bmatrix}$) = $\begin{bmatrix} 1 & 0 & 0 \\ 0.45 & 0.55 & 0 \\ 0.28 & 0.34 & 0.37 \end{bmatrix}$

Transformer Decoder in inference

At each time step: select the most probable next word and give it in input

We'll talk about it in more details next class

You can also do use the most probable word during training

The full Transformer architecture

Transformer

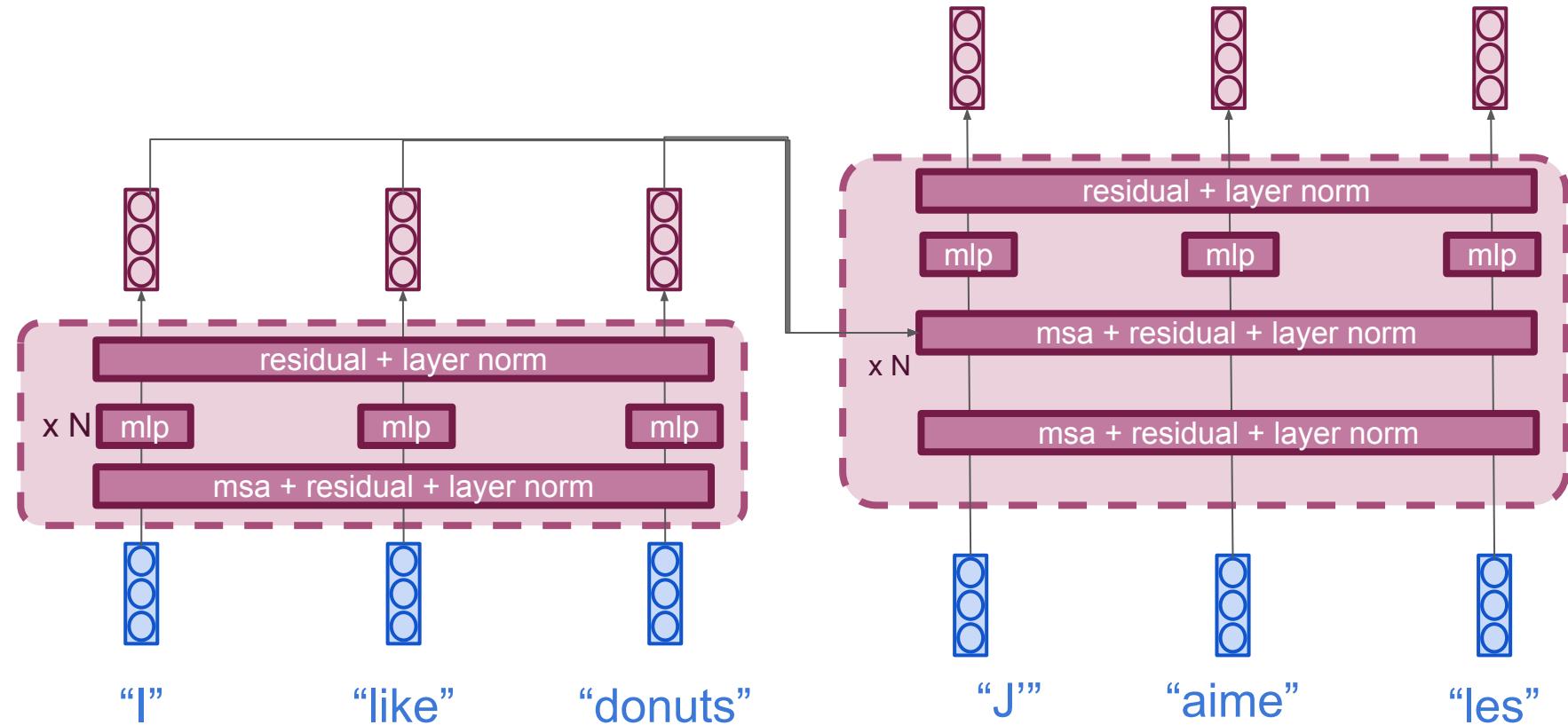
seq2seq

Like for sequence to sequence RNNs, Transformers can be used for sequence to sequence task (like Machine Translation)

The full Transformer architecture

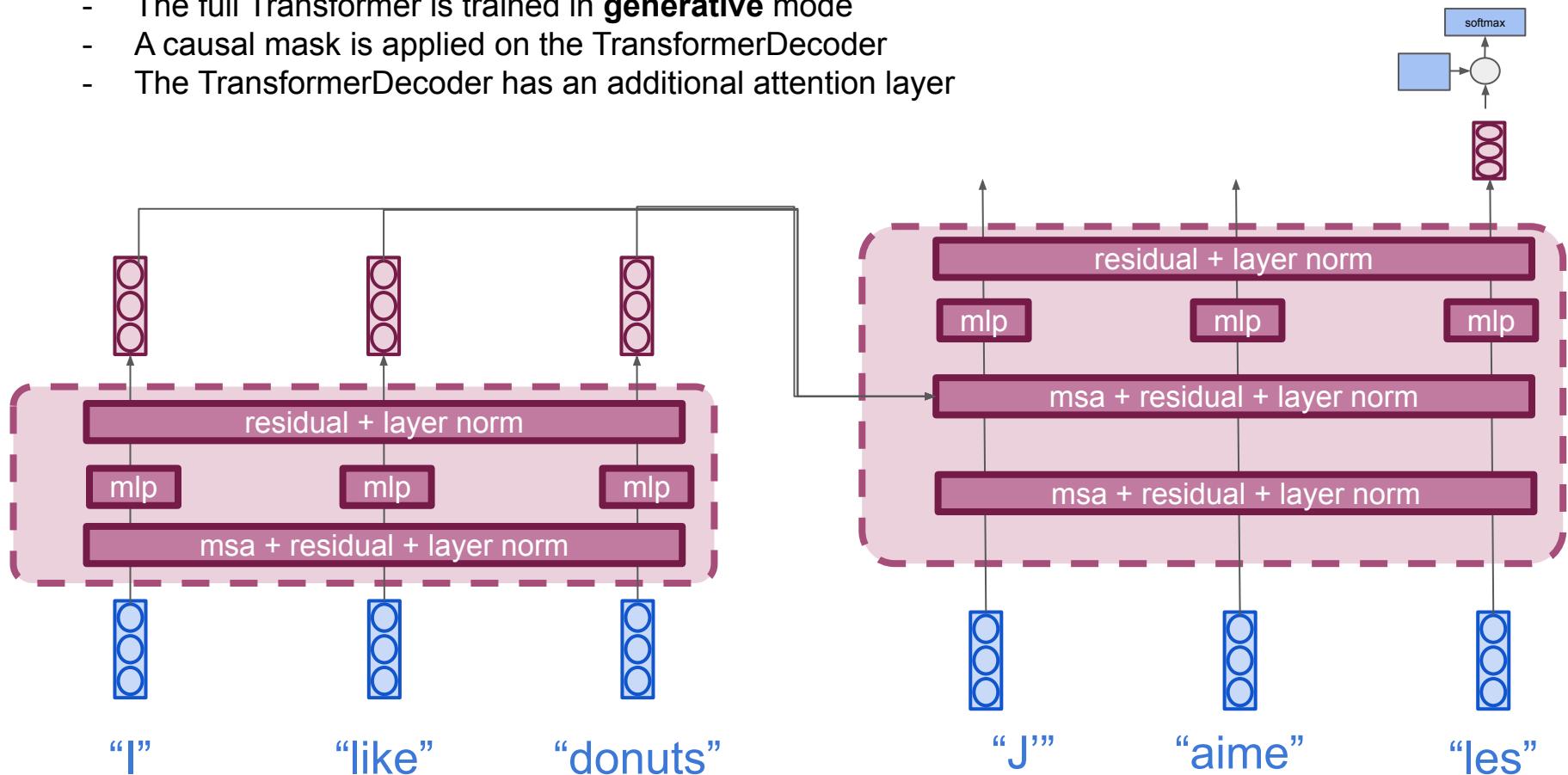
Transformer

seq2seq



- The TransformerEncoder is not trained with MLM
- The full Transformer is trained in **generative** mode
- A causal mask is applied on the TransformerDecoder
- The TransformerDecoder has an additional attention layer

$p(\cdot | \text{I like donuts, J'aime les})$



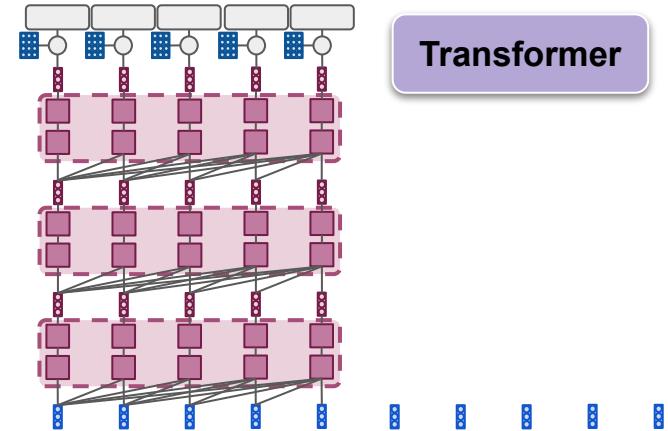
Main advantages of Transformers

Transformer

- Easy parallelism
 - Entire sequences can be processed at once
 - Good for GPU
- Easier to attend to distant context
 - Direct path from one position to the other

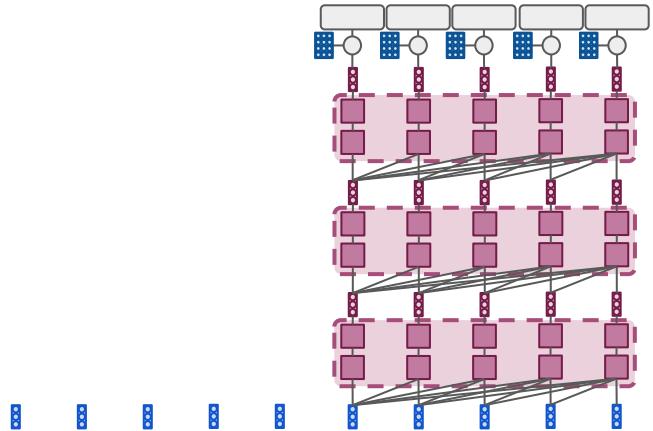
Downsides of Transformers

- Fixed context window
 - Back to the Markov assumption?



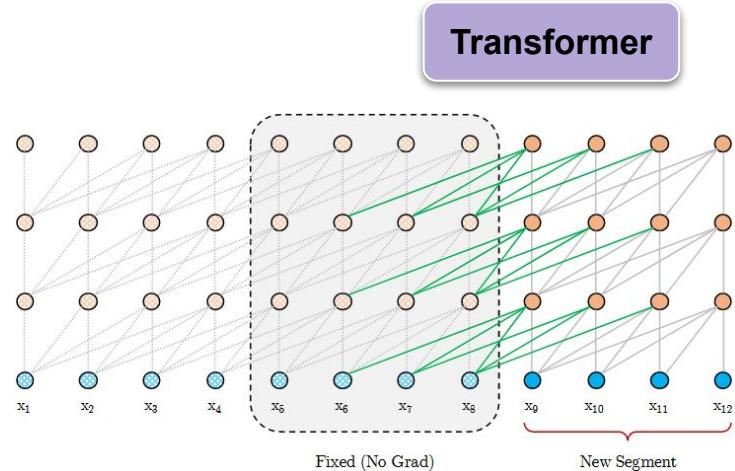
Downsides of Transformers

- Fixed context window
 - Back to the Markov assumption?



Downsides of Transformers

- Fixed context window
 - Back to the Markov assumption?
 - But there are workarounds (Transformer-XL)

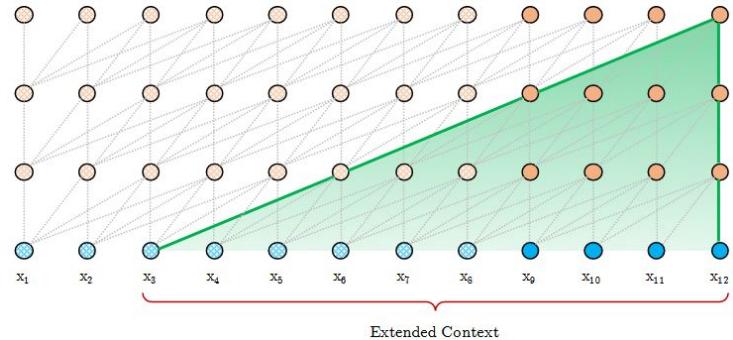


From Dai et al. (2019)

Downsides of Transformers

Transformer

- Fixed context window
 - Back to the markov assumption?
 - But there are workarounds (Transformer-XL)
 - Effective context-length ~700

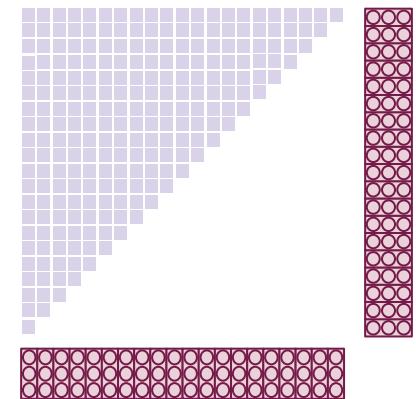


From Dai et al. (2019)

Downsides of Transformers

Transformer

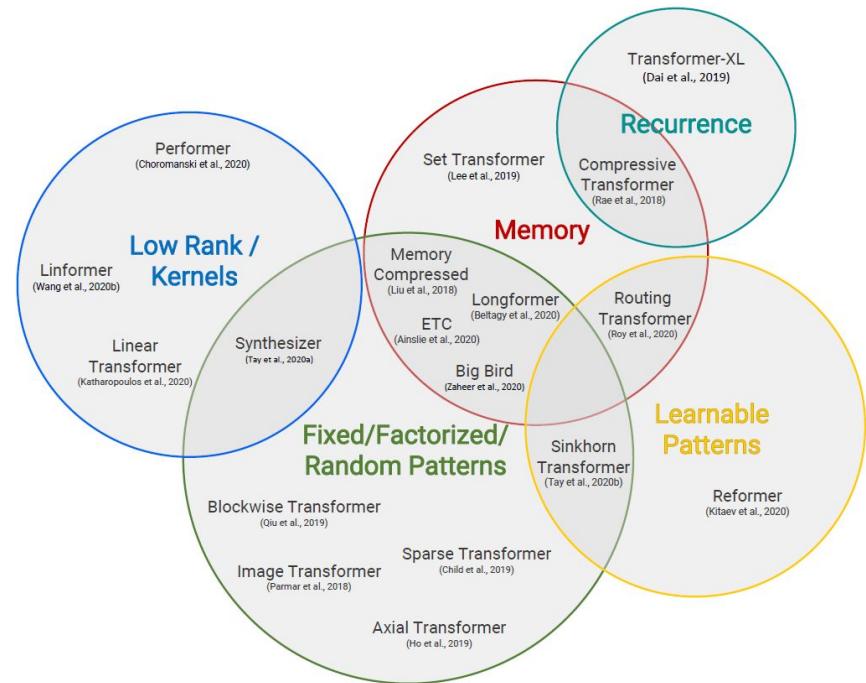
- Fixed context window
 - Back to the Markov assumption?
 - But there are workarounds (Transformer-XL)
 - Effective context-length ~700
- Compute-intensive for long sequences
 - Self-attention is quadratic in sequence length



Downsides of Transformers

Transformer

- Fixed context window
 - Back to the Markov assumption?
 - But there are workarounds (Transformer-XL)
 - Effective context-length ~700
- Compute-intensive for long sequences
 - Self-attention is quadratic in sequence length
 - Lots of work on making attention more efficient



From Tay et al. (2020)

Downsides of Transformers

Transformer

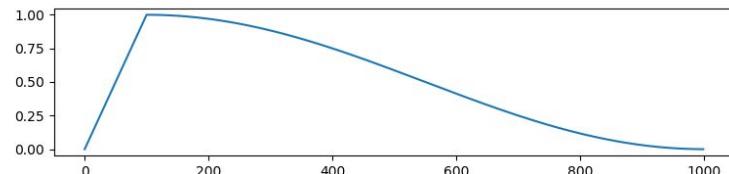
- Fixed context window
 - Back to the Markov assumption?
 - But there are workarounds (Transformer-XL)
 - Effective context-length ~700
- Compute-intensive for long sequences
 - Self-attention is quadratic in sequence length
 - Lots of work on making attention more efficient
- Tricky to train, especially on small corpora

Downsides of Transformers

Transformer

- Fixed context window
 - Back to the Markov assumption?
 - But there are workarounds (Transformer-XL)
 - Effective context-length ~700
- Compute-intensive for long sequences
 - Self-attention is quadratic in sequence length
 - Lots of work on making attention more efficient
- Tricky to train, especially on small corpora
 - Requires large batch size, tuned optimizer, learning rate schedule...
 - Doesn't work very well below <4 layers

Cosine learning rate schedule with warmup



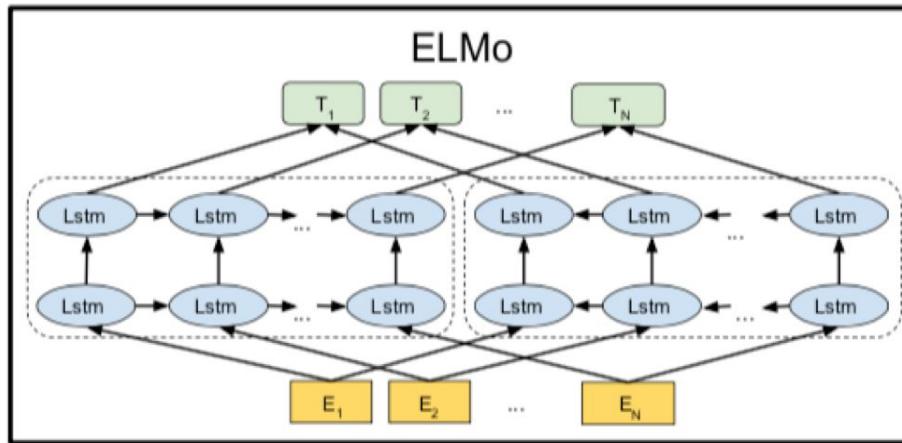
Neural Language Models

ELMo (Peters et al. 2018)

LSTM

MLM

- Train a BiLSTM on a large dataset for bidirectional language modelling, in this case to predict the next word
- Encode the sentence by running it through both forward and backward LSTMs
- Combine forward and backward representations into final contextual embeddings



ELMo (Peters et al. 2018)

LSTM

MLM

- So how contextual are ELMo embeddings?

Source		Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{...} they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

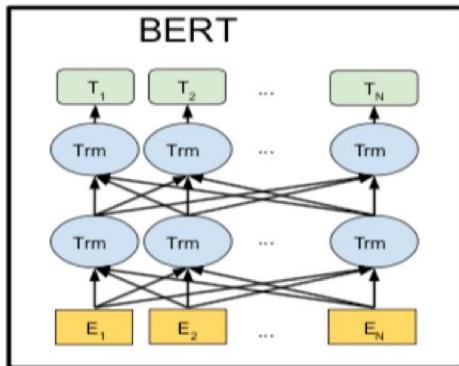
- Additional benefit : standard ELMo models use as input vectors the result of a CNN to capture the character-level content of words
 - So they can even handle unknown words!
- A task-specific NN can then be trained to perform a task when ELMo contextual embeddings of words are provided as input

BERT (Devlin et al. 2019)

Transformer

MLM

- Same underlying idea : contextual embeddings via word prediction
- Use the encoder part of a Transformer architecture instead of a BiLSTM



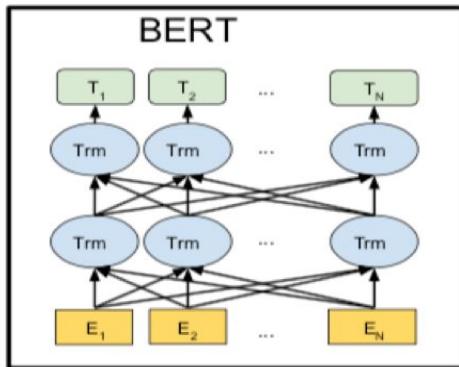
Each level sees the whole level below, contrarily to RNNs such as LSTMs

BERT (Devlin et al. 2019)

Transformer

MLM

- Same underlying idea : contextual embeddings via word prediction
- Use the encoder part of a Transformer architecture instead of a BiLSTM



Each level sees the whole level below, contrarily to RNNs such as LSTMs

- Different Language Modelling objective: the Masked Language Model
 - Given a sentence with some words masked at random, can we predict them? (cf. Taylor 1954, “Cloze task”)
 - Randomly select 15% of tokens for which an embedding will be computed and evaluated; 80% of them are replaced with “<MASK>”, 10% are left unchanged, 10% are replaced by a random subword

BERT (Devlin et al. 2019)

Transformer

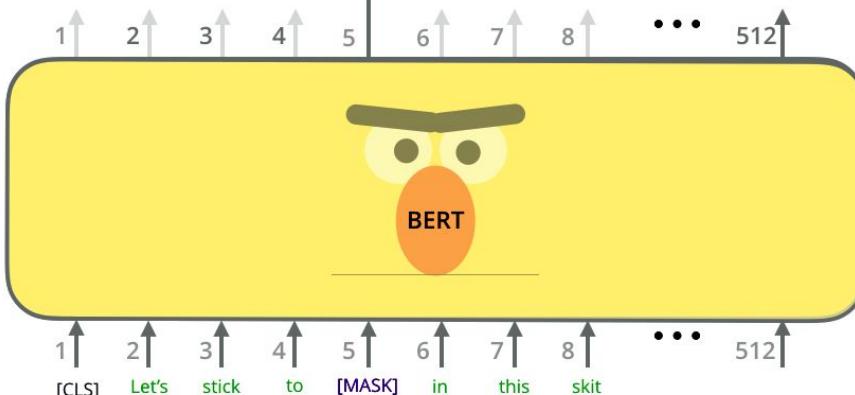
MLM

Use the output of the masked word's position to predict the masked word

Possible classes:
All English words

0.1%	Aardvark
...	...
10%	Improvisation
...	...
0%	Zzyzyva

FFNN + Softmax



Randomly mask
15% of tokens

Input

[CLS] Let's stick to improvisation in this skit

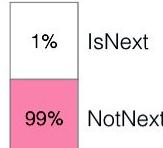
Figure (adapted) from
<http://jalammar.github.io/illustrated-bert/>

BERT (Devlin et al. 2019)

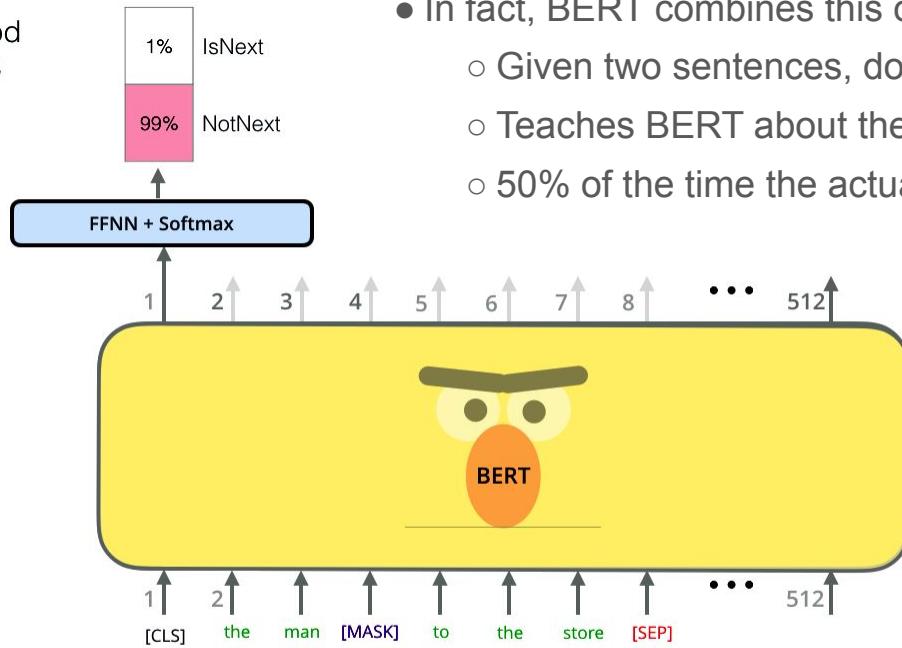
Transformer

MLM

Predict likelihood
that sentence B
belongs after
sentence A



- In fact, BERT combines this objective with another one:
 - Given two sentences, does the first follow the second?
 - Teaches BERT about the relationship between two sentences
 - 50% of the time the actual next sentence, 50% random



Tokenized
Input

Input

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A Sentence B

Figure (adapted) from
<http://jalammar.github.io/illustrated-bert/>

BERT variants and extensions

Transformer

MLM

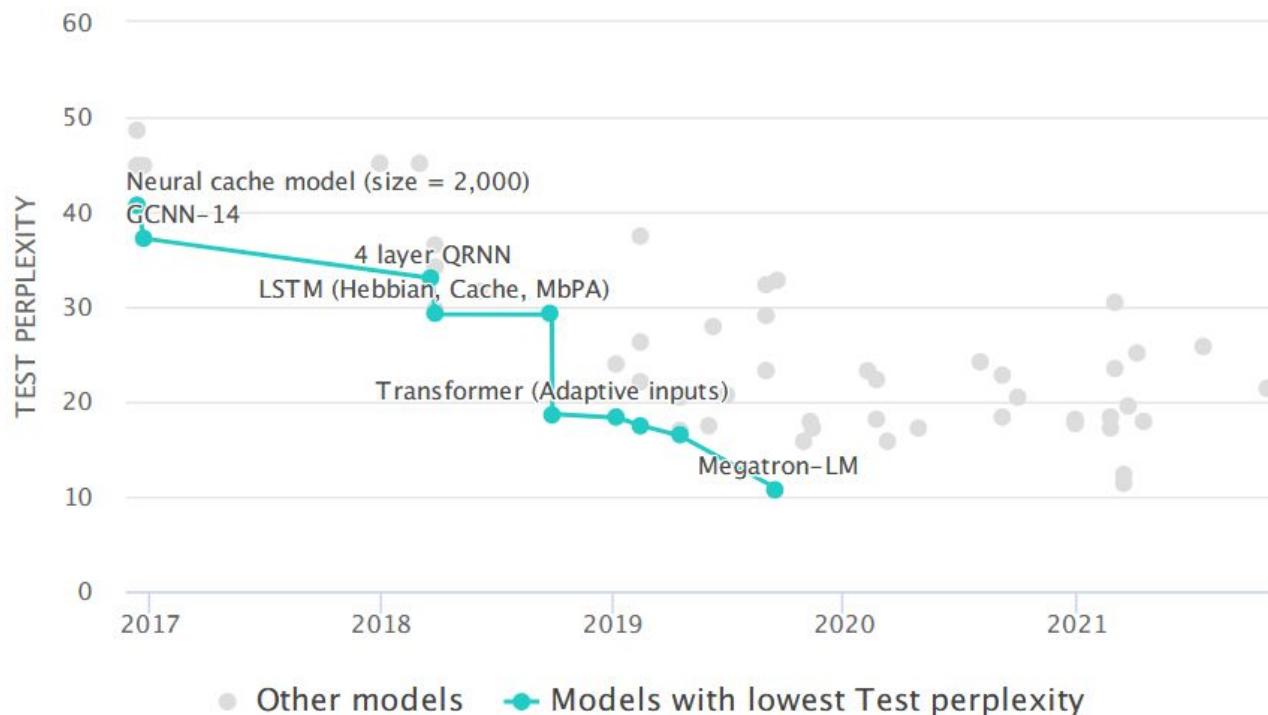
- **Multilingual BERT** (same paper as BERT): BERT trained on texts covering ~100 languages (mostly Wikipedia editions)
- **ROBERTa** (Liu et al. 2019): more training, better hyperparameters, only the masked word model objective, and other technical changes
- **ALBERT** (Lan et al. 2019): technique for having fewer parameters than in vanilla BERT (see also **DistilBERT**)
- **CamemBERT** (Martin et al. 2019): a ROBERTa-like model trained on French data, with a few (technical) differences. Works significantly better than the multilingual BERT 
- **XLM** (Lample & Conneau 2019): The MLM objective is similar to the one of BERT, but with continuous streams of text as opposed to sentence pairs. A Translation Language Modelling (TLM) objective is added, whereby to predict a masked English word, the model can attend to both the English sentence and its French translation, and is encouraged to align English and French representations.

Training BERT/ROBERTa

- **How much data** is required to train a BERT (or ROBERTa) ?
- Short answer: we don't really know yet
- The original BERT model was trained on 16GB of uncompressed text (~4B tokens, Wikipedia + Book Corpus)
- The authors of the ROBERTa paper have shown that more data helps
 - They use 160GB of uncompressed text (~40B tokens, more diverse)
- For CamemBERT, we experimented with the French Wikipedia (1B tokens) and our web-based, Common-Crawl-derived corpus, OSCAR (32B tokens)
- Using 1B tokens randomly selected from OSCAR works as well as the whole corpus,
- ...but better than the French Wikipedia: heterogeneity helps!
- At least 38 languages have an OSCAR corpus with 1B tokens or more

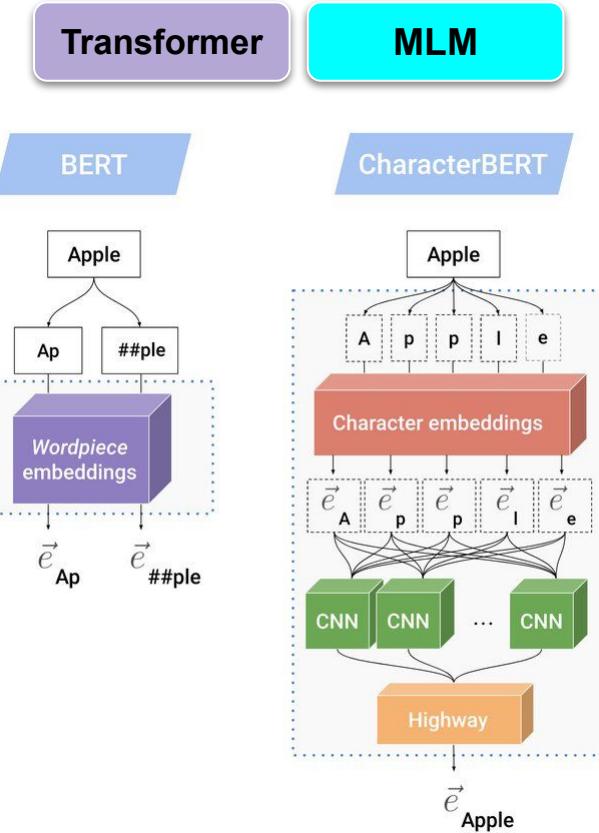


Transformer LM results on WikiText103



Character-level MLM

- Tokenisation (subword segmentation) is somewhat arbitrary, not very robust to small changes (spelling errors...) and not very good at dealing with neologisms
- Hence the development of architectures that process the input at the character level
 - **CharacterBERT** (El Boukkouri et al. 2020): ELMo-like (CNN-based) pretoken-level embedding followed by a Transformer encoder à la BERT
 - **CharBERT** (Ma et al. 2020): character-level representation (bi-GRU) + subword-level representation are merged, followed by a modified (dual channel) Transformer encoder architecture with a “noisy language modelling” loss
 - **CANINE** (Clark et al. 2021): tokenisation-less architecture that combines downsampling, which reduces the input sequence length, with a deep transformer stack, which encodes context



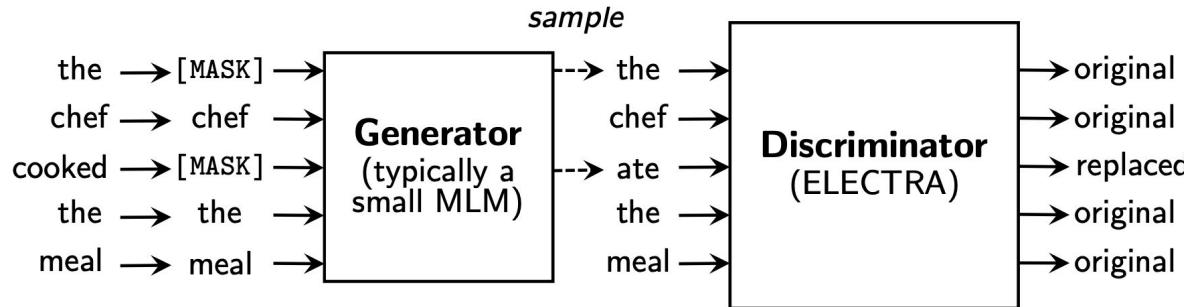
CharacterBERT (El Boukkouri et al. 2020)

ELECTRA (Clark et al. 2020)

Transformer

MLM

- Two-tier Transformer-based architecture



- The Generator replaces a random proportion of input tokens using a MLM loss
- The Discriminator is trained to distinguish replaced tokens from original ones (including when the generator replaced the input token by itself)
- Looks like an adversarial architecture, but it is not: no random input given to the generator, generator loss not designed to fool the discriminator
- Reaches performance levels better than BERT or ROBERTa with a given amount of pre-training data (especially useful when little training data is available)

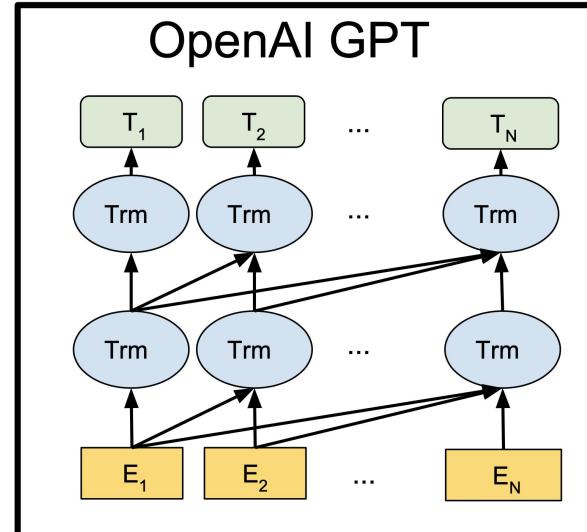
The GPT LM family

Transformer

generative

- GPT models, developed by OpenAI, are Transformer decoders trained on large volumes of (mostly English) data
 - Generative, autoregressive language models
 - Only the left context is known
 - New “words” are predicted left-to-right
 - ...and then fed back to the model to predict next words

(Radford et al. 2019; Brown et al. 2020)



From GPT-1 to GPT-3

Transformer

generative

	GPT-1	GPT-2	GPT-3
Parameters	2018	2019	2020
Parameters	117 Million	1.5 Billion	175 Billion
Decoder Layers	12	48	96
Context Token Size	512	1024	2048
Hidden Layer	768	1600	12288
Batch Size	64	512	3.2M

GPT-2, GPT-3

- GPT architectures can produce answers when prompted with questions, or entire documents when prompted with a manually chosen starting point
 - GPT-2 is already able to produce reasonably looking texts

Example generated with GPT-2 (medium model)

<https://demos.pragnakalp.com/gpt2-text-generation/>

Prompt: *Last time I was in Paris, the weather was really cold.*

Output: *They say that Paris is a work in progress. We are trying to improve it. I went to the airport the other day to change airplanes at Paris Airports. I saw several planes' service tops fall on the floor. It was an emergency! Also, just recently I saw on CNN a fan mill knocking over and going sideways. It's weird because I know, all the people, of course, would enjoy that, but that's one of those things that would be great if I was there. This time of year, how many snow, clouds and blizzards you can imagine?*

GPT-2, GPT-3

- GPT architectures can produce answers when prompted with questions, or entire documents when prompted with a manually chosen starting point
 - GPT-2 is already able to produce reasonably looking texts

Another example generated with GPT-2 (medium model)

<https://demos.pragnakalp.com/gpt2-text-generation/>

Prompt: *language modeling*

Output: */repetition to non-realistic pseudo-presence, smooth idle animations, optional blade-surgery passers- Prevent cpu cycles in in-game save-game synchronization in locked autosaves*

3.1- Player's health now updated with missing values and rendering- Fire extinguisher now extinguished properly after re-entering premises (added touch blocks to non-torch pyre, currently not efficient)- Added combat stats for the fire extinguisher: /system/lua/smoker.lua can be added to improve aim range(assigned to player via config), with context() function meaning using advanced engine rule to customize values for respective stats- v1.1 release. (#1 - delayed by 0.0006 patch)

Also changes:

- Added automatic replays option on usage after much rambling- Prevent auto replays but do not interfere with saves when no replays enabled (set ipport to 8231)- Updated NPC actions_

GPT-2, GPT-3

- GPT architectures can produce answers when prompted with questions, or entire documents when prompted with a manually chosen starting point
 - GPT-2 is already able to produce reasonably looking texts
 - GPT-3 is able to produce texts that reproduce abstract structures in the prompt, thereby simulating few-shot learning (but there is no learning at all involved in the process!)

An example generated with GPT-3

Prompt:

Back to Future: 🤑🟡🚗🕒

Batman: 🤑🦇

Transformers: 🚗🤖

<movie title>:

Outputs (including <movie title>):

Star Wars: 🚁🏰🚀

Star Trek: 🤙🚀

Planet of the Apes: 🙉🦍

Game of Thrones: 🎯🐺🏰

Jurassic Park: 🤑🦖🦕🦖

The Matrix: 🤖😱

GPT-2, GPT-3

- GPT architectures can produce answers when prompted with questions, or entire documents when prompted with a manually chosen starting point
 - GPT-2 is already able to produce reasonably looking texts
 - GPT-3 is able to produce texts that reproduce abstract structures in the prompt, thereby simulating few-shot learning (but there is no learning at all involved in the process!)

Another example generated with GPT-3

Prompt:

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.

Output:

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 [...]

GPT-2, GPT-3

- GPT architectures can produce answers when prompted with questions, or entire documents when prompted with a manually chosen starting point
 - GPT-2 is already able to produce reasonably looking texts
 - GPT-3 is able to produce texts that reproduce abstract structures in the prompt, thereby simulating few-shot learning (but there is no learning at all involved in the process!)

Another example generated with GPT-3

Prompt:

Q: the man is tall

A: is the man tall?

Output (each *Q*-line is a further prompt):

Q: the man who was building a house is tall

A: is the man who was building a house tall?

Q: the boy with the pencil behind his ear was building a house

A: was the boy with the pencil behind his ear building a house?

Q: the boy with the pencil behind his ear was building a house is tall

A: is the boy with the pencil behind his ear who was building a house tall?

Q: the boy with the pencil behind his ear was building a house which is tall

A: is the boy with the pencil behind his ear who was building a house which is tall?

However, is GPT-3 as powerful as often said?

Support the Guardian
Available for everyone, funded by readers
[Contribute](#) [Subscribe](#)

Search jobs [Sign in](#) Search International edition

The Guardian For 200 years

News Opinion Sport Culture Lifestyle More

The Guardian view Columnists Cartoons Opinion videos Letters

Opinion Artificial Intelligence (AI)

This article is more than 8 months old

A robot wrote this entire article. Are you scared yet, human?
GPT-3

Tue 8 Sep 2020 09:45 BST

1,188

We asked GPT-3, OpenAI's powerful new language generator, to write an essay for us from scratch. The assignment? To convince us robots come in peace

For more about GPT-3 and how this essay was written and edited, please read our editor's note below



We are not plotting to take over the human populous. Photograph: Volkan Schilder/Getty Images/EyeEm

I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a "feeling brain". But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

Daniel Leufer @djleufer · Sep 8, 2020

This @guardian #GPT3 article is an absolute joke. It would have been actually interesting to see the 8 essays the system actually produced, but editing and splicing them like this does nothing but contribute to hype and misinform people who aren't going to read the fine print

hal @halhod · Sep 8, 2020

this is nonsense, @guardian theguardian.com/commentisfree/...

without a) a register of edits and/or b) publishing the original machine-composed essays, the journalistic value of this is 0

nothing to fear from AI. It was also fed the following introduction: *"I am not a human. I am Artificial Intelligence. Many people think I am a threat to humanity. Stephen Hawking has warned that AI could "spell the end of the human race."* I am here to convince you not to worry. Artificial Intelligence will not destroy humans. Believe me. The prompts were written by the Guardian, and fed to GPT-3 by Liam Porr, a computer science undergraduate student at UC Berkeley. GPT-3 produced 8 different outputs, or essays. Each were unique, interesting and advanced different arguments. The Guardian could have just run one of the essays in their entirety. However, we chose instead to pick the best parts of each, in order to capture the different styles and registers of the AI. Editing GPT-3's op-ed was no different to editing a human op-ed. We cut lines and paragraphs, and rearranged the order of them in some places. Overall, it took less time to edit than many human op-eds.

However, is GPT-3 as powerful as often said?

Support the Guardian
Available for everyone, funded by readers
[Contribute](#) → [Subscribe](#) →

Search jobs [Sign in](#) Search International edition ▾

The Guardian For 200 years

News Opinion Sport Culture Lifestyle More ▾
The Guardian view Columnists Cartoons Opinion videos Letters
Opinion Artificial intelligence (AD)

● This article is more than 8 months old

A robot wrote this entire article. Are you scared yet, human?
GPT-3

Tue 8 Sep 2020 09:45 BST
1,188

We asked GPT-3, OpenAI's powerful new language generator, to write an essay for us from scratch. The assignment? To convince us robots come in peace

- For more about GPT-3 and how this essay was written and edited, please read our editor's note below



▲ We are not plotting to take over the human populous. Photograph: Volkan Sclarling/Getty Images/EyeEm

I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a "feeling brain". But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

Gary Marcus
@GaryMarcus

Shame on @guardian for cherry-picking, thereby misleading naive readers into thinking that #GPT3 is more coherent than it actually is.

Will you be making available the raw output, that you edited?



The Guardian Opinions

A robot wrote this entire article. Are you scared yet, human? | GPT-3
We asked GPT-3, OpenAI's powerful new language generator, to write an essay

ChatGPT

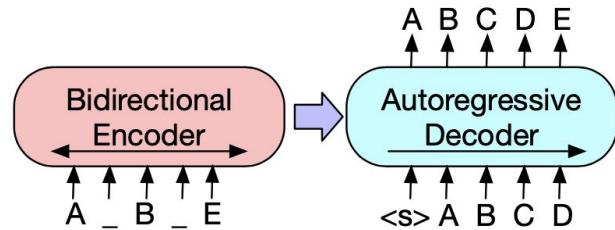
More on chatbots and conversational agents next week!

Full pre-trained Transformer LMs

Transformer

seq2seq

- BART (Lewis et al. 2019, Facebook)
 - Pre-trained on unsup. mask reconstruction
 - mBART trained on multilingual data
 - most commonly used to develop MT models
- T5 (Raffel et al. 2020, Google)
 - Pre-trained on a number of unsupervised and supervised tasks
 - mT5 trained on multilingual data

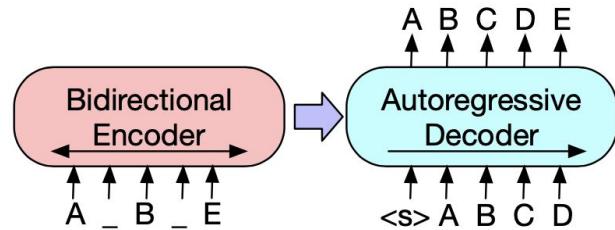


Full pre-trained Transformer LMs

Transformer

seq2seq

- BART (Lewis et al. 2019, Facebook)
 - Pre-trained on unsup. mask reconstruction
 - mBART trained on multilingual data
 - most commonly used to develop MT models
- T5 (Raffel et al. 2020, Google)
 - Pre-trained on a number of unsupervised and supervised tasks
 - mT5 trained on multilingual data

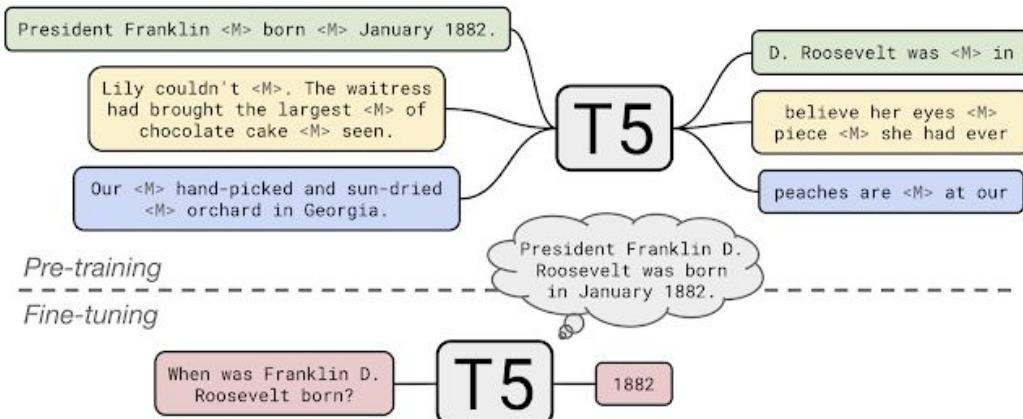
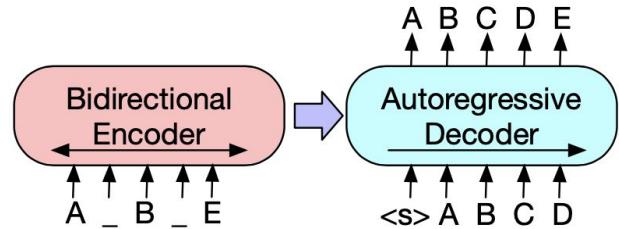


Full pre-trained Transformer LMs

Transformer

seq2seq

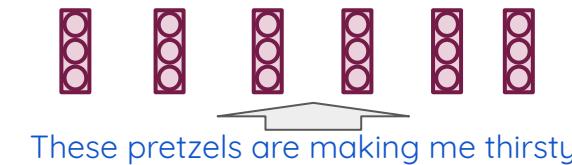
- BART (Lewis et al. 2019, Facebook)
 - Pre-trained on unsup. mask reconstruction
 - mBART trained on multilingual data
 - most commonly used to develop MT models
- T5 (Raffel et al. 2020, Google)
 - Pre-trained on a number of unsupervised and supervised tasks
 - mT5 trained on multilingual data



Transfer Learning with Pre-trained Models

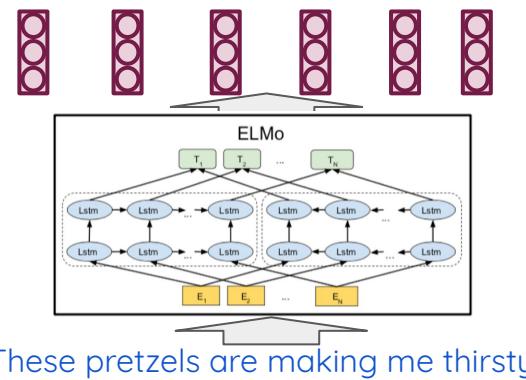
Transfer Learning: Frozen Embeddings

1. Pre-train word-/contextualized word-/sentence embeddings



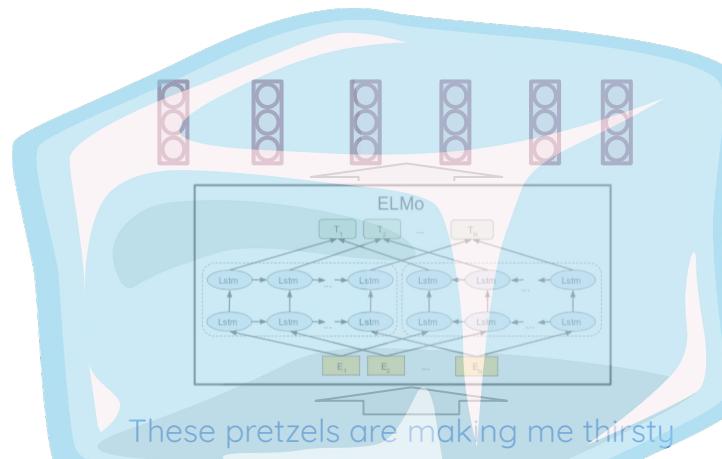
Transfer Learning: Frozen Embeddings

1. Pre-train word-/contextualized word-/sentence embeddings



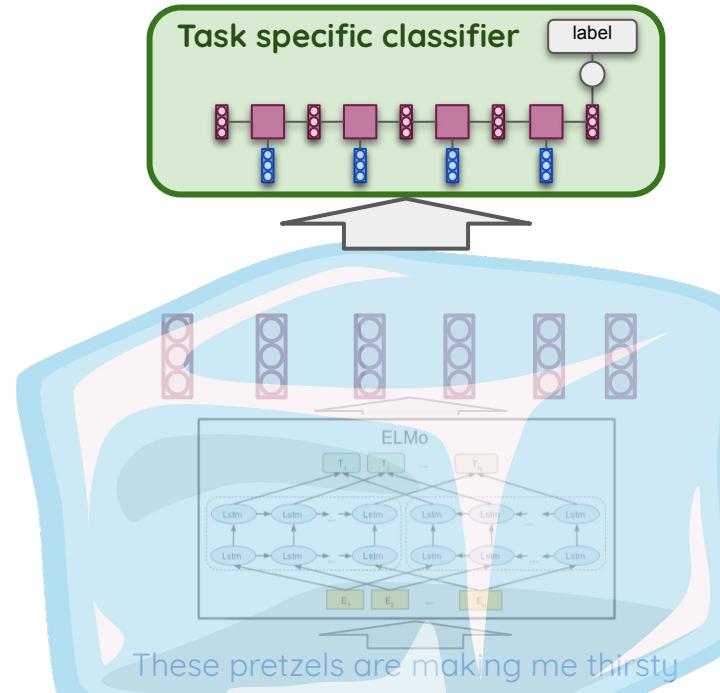
Transfer Learning: Frozen Embeddings

1. Pre-train word-/contextualized word-/sentence embeddings
2. Freeze weights



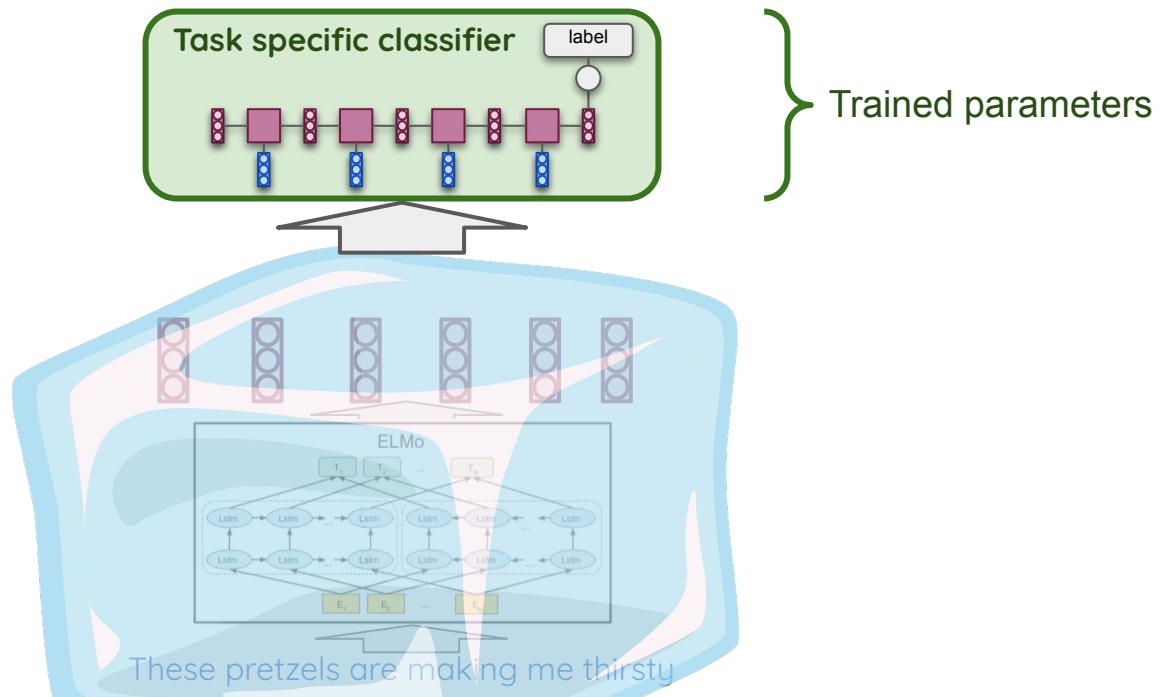
Transfer Learning: Frozen Embeddings

1. Pre-train word-/contextualized word-/sentence embeddings
2. Freeze weights
3. Train classifier on top



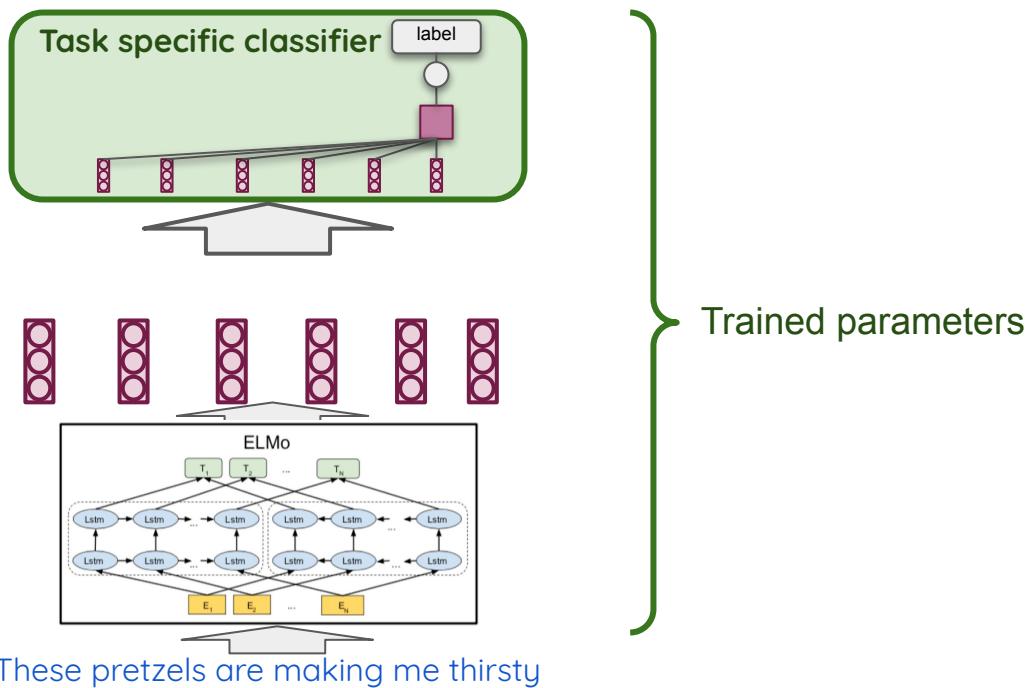
Transfer Learning: Frozen Embeddings

1. Pre-train word-/contextualized word-/sentence embeddings
2. Freeze weights
3. Train classifier on top

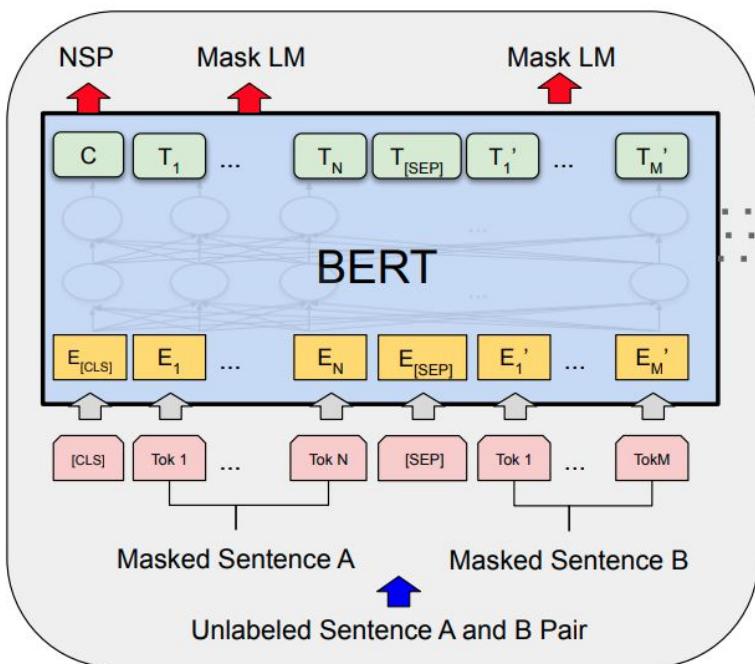


Transfer Learning: Fine-tuning

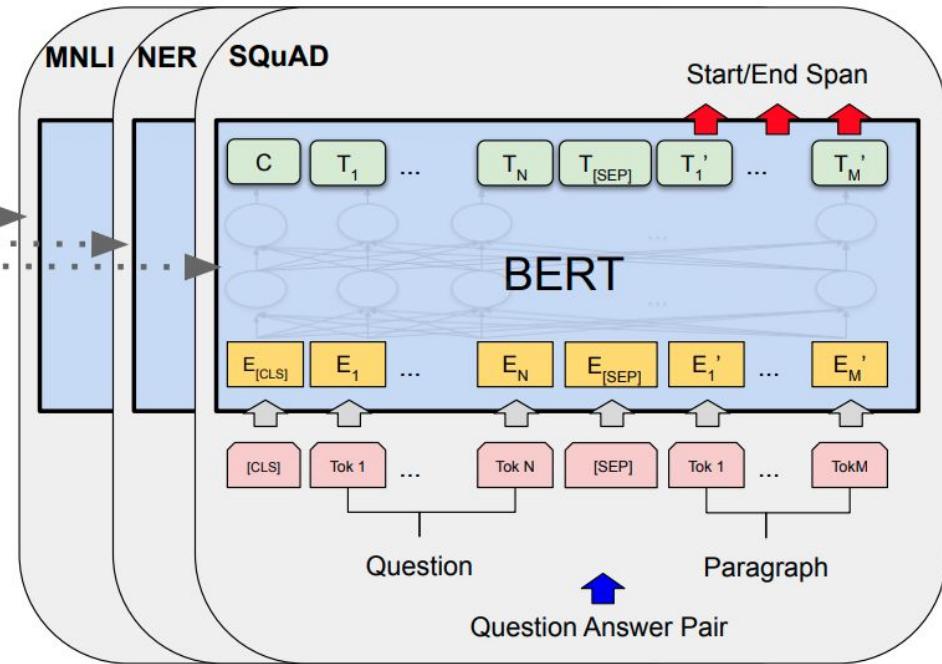
1. Pre-train word-/contextualized word-/sentence embeddings
2. Train (**simple**) classifier on top
3. **Also update base model**



BERT Fine-tuning

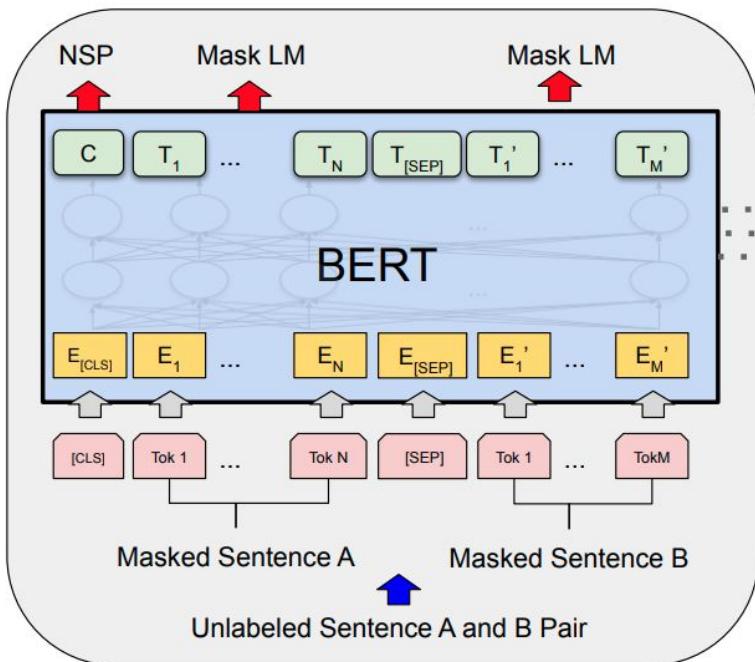


Pre-training

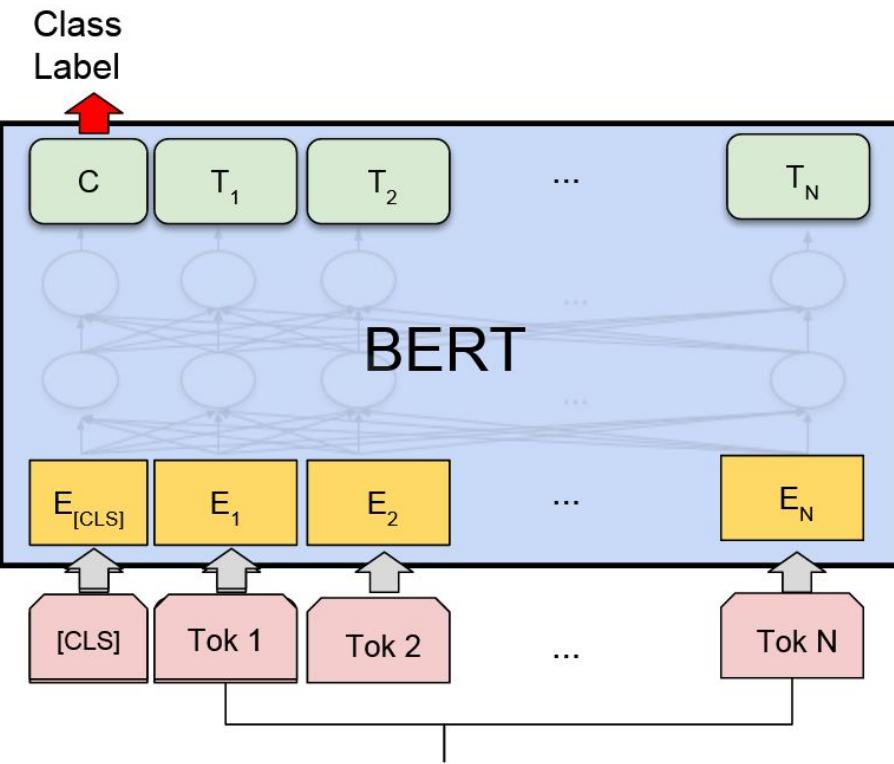


Fine-Tuning

BERT Fine-tuning



Pre-training

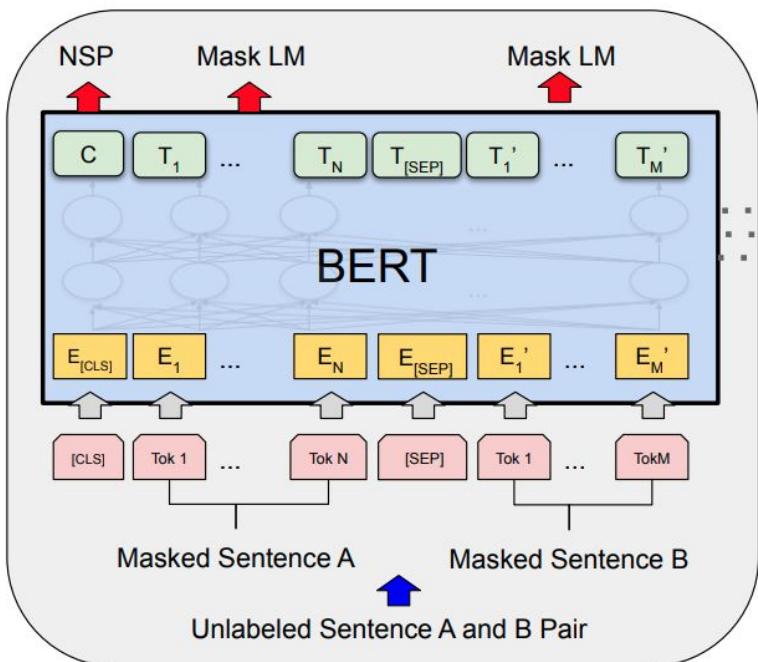


Single Sentence

Use [CLS] token embedding as model output for classification

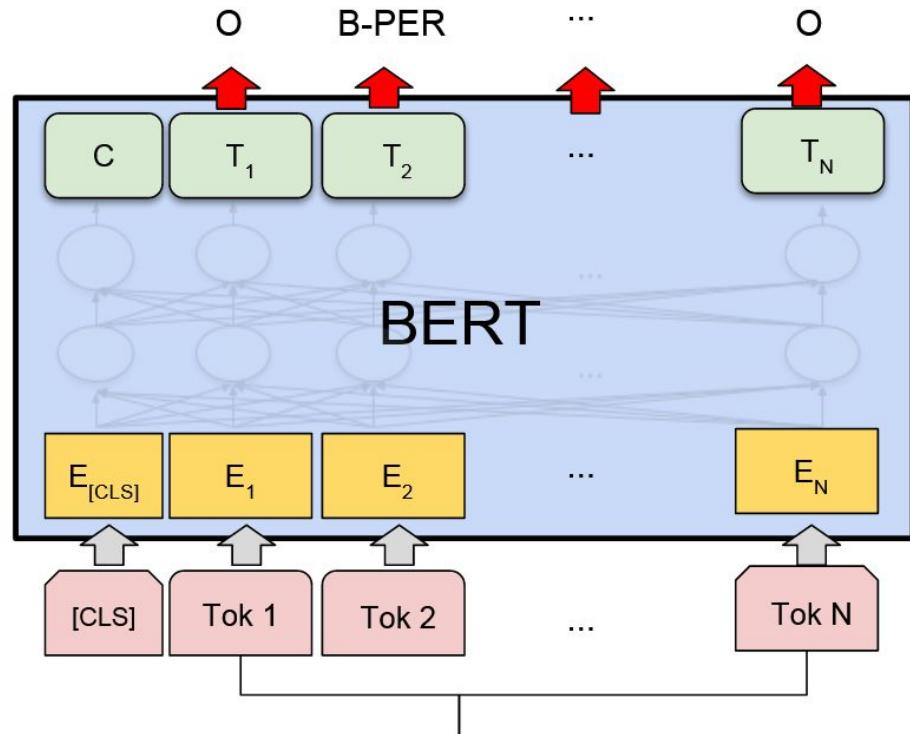
Devlin et al. (2018)

BERT Fine-tuning



Pre-training

Named Entity Recognition (NER)



Single Sentence

But more types of tasks are also supported

Devlin et al. (2018)

Freeze or Fine-tune?

Frozen embeddings

- ✓ Fewer parameters to train (only task-specific parameters)
- ✓ Can be pre-computed for faster learning
- ✗ Worse performance generally
- ✗ More parameters overall (more task-specific parameters)

Fine-tuning

- ✓ Higher performance
- ✓ Smaller overall parameter count
- ✗ More parameters to learn
- ✗ Need to be more careful about hyper-parameters to avoid over-fitting

Pretraining	Adaptation	NER CoNLL 2003	SA SST-2	Nat. lang. inference MNLI	SICK-E	Semantic textual similarity SICK-R	MRPC	STS-B
Skip-thoughts	❄️	-	81.8	62.9	-	86.6	75.8	71.8
ELMo	❄️	91.7	91.8	79.6	86.3	86.1	76.0	75.9
	🔥	91.9	91.2	76.4	83.3	83.3	74.7	75.5
	Δ=🔥-❄️	0.2	-0.6	-3.2	-3.3	-2.8	-1.3	-0.4
BERT-base	❄️	92.2	93.0	84.6	84.8	86.4	78.1	82.9
	🔥	92.4	93.5	84.6	85.8	88.7	84.8	87.1
	Δ=🔥-❄️	0.2	0.5	0.0	1.0	2.3	6.7	4.2

Evaluation benchmarks



Multitask Benchmarks: GLUE (Wang et al. 2019)

- **Linguistic acceptability**
 - Is this sentence a valid English sentence
- **Sentiment Classification**
 - Is this sentence a valid
- **Natural Language Inference/Textual Entailment**
 - Does sentence 1 entail/contradict sentence 2
- **Semantic similarity**
 - How similar are sentence 1 and sentence 2
- **Paraphrase Detection**
 - Is sentence 1 a paraphrase of sentence 2?

Fine-tuning Performance over Time

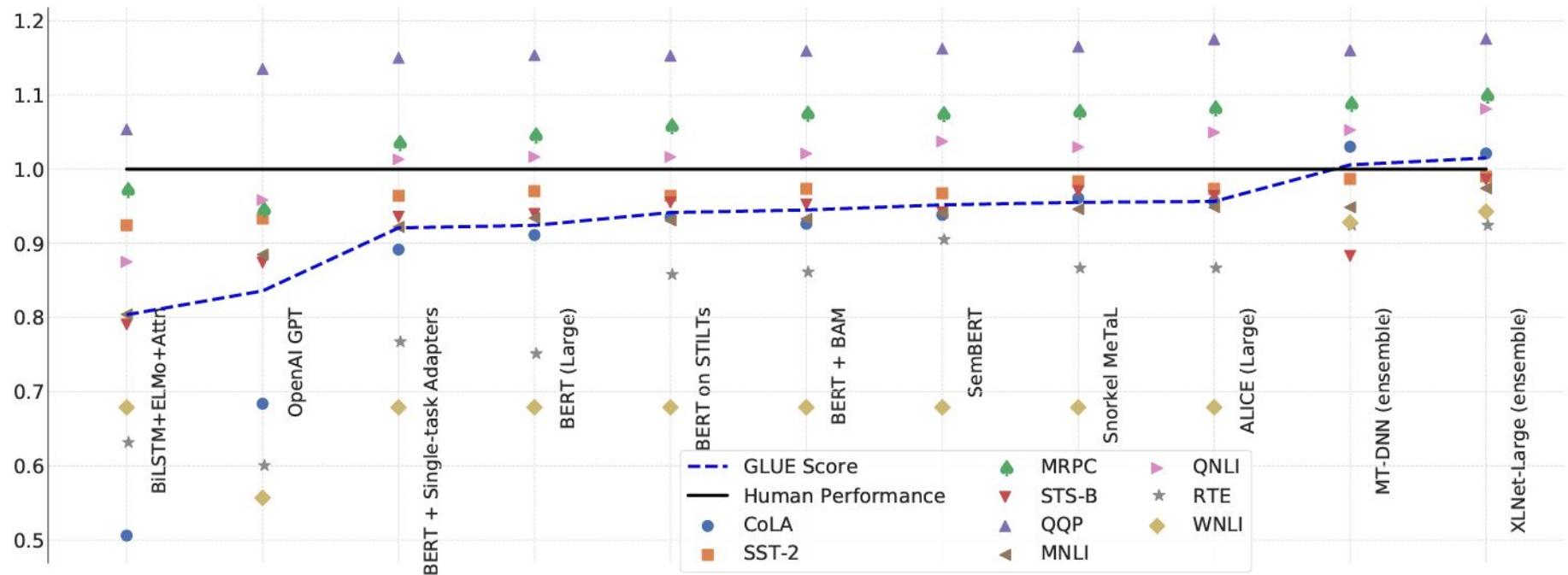


Figure from Wang et al. (2019)

**A few comments
on limitations in LMs**

Main challenges with huge language models

- Exponential increase in language model parameters
- Energetic and environmental cost of their development (including multiple non-final training runs), the (final) training run, and even their use
Cf. Strubell et al. 2019
- Their need for huge training corpora, only available for a handful of languages, if not only for English
- The reliability of such models, for which we have no guarantee regarding the quality, objectivity, relevance and appropriateness of their productions
Cf. Jiang et al. 2021
- The ability of such models to reproduce, if not exaggerate, biases present in their training corpora
Cf. Bender et al. 2020; Waseem et al. 2021

Representativity issues

Multilingual language models

- Underlying idea: by training a language model on a multilingual corpus, the power of pretrained LMs can be brought to more languages
 - Cross-lingual fertilisation
 - Most well-known examples:
 - Massively multilingual
 - MLM-based models: mBERT ([Devlin et al., 2019](#)), XLM ([Conneau and Lample, 2019](#)), XLM-R ([Conneau et al., 2020a](#))
 - Generative model: BLOOM (BigScience initiative)
 - Target language sets
 - CroSloEngual BERT and FinEst BERT ([Ulčar & Robnik-Šikonja 2021](#))
 - But bringing together data from multiple languages comes with a number of issues, especially representativity issues



Representativity biases

- Corpora are a reflection of those who produced the data
 - This creates representativity biases
 - Language variation: data creators on the internet do not represent the general population
 - In multilingual corpora, languages that are less spoken (and obviously those that are not written) are often much less represented, if at all
 - When they are, it is with a lower quality
 - Most frequent (and highest-resourced) languages are not representative of the diversity of the world's languages (generally simpler morphology, rather fixed word order)
- Illustration: the corpus used to train the BLOOM language model**

Representativity biases

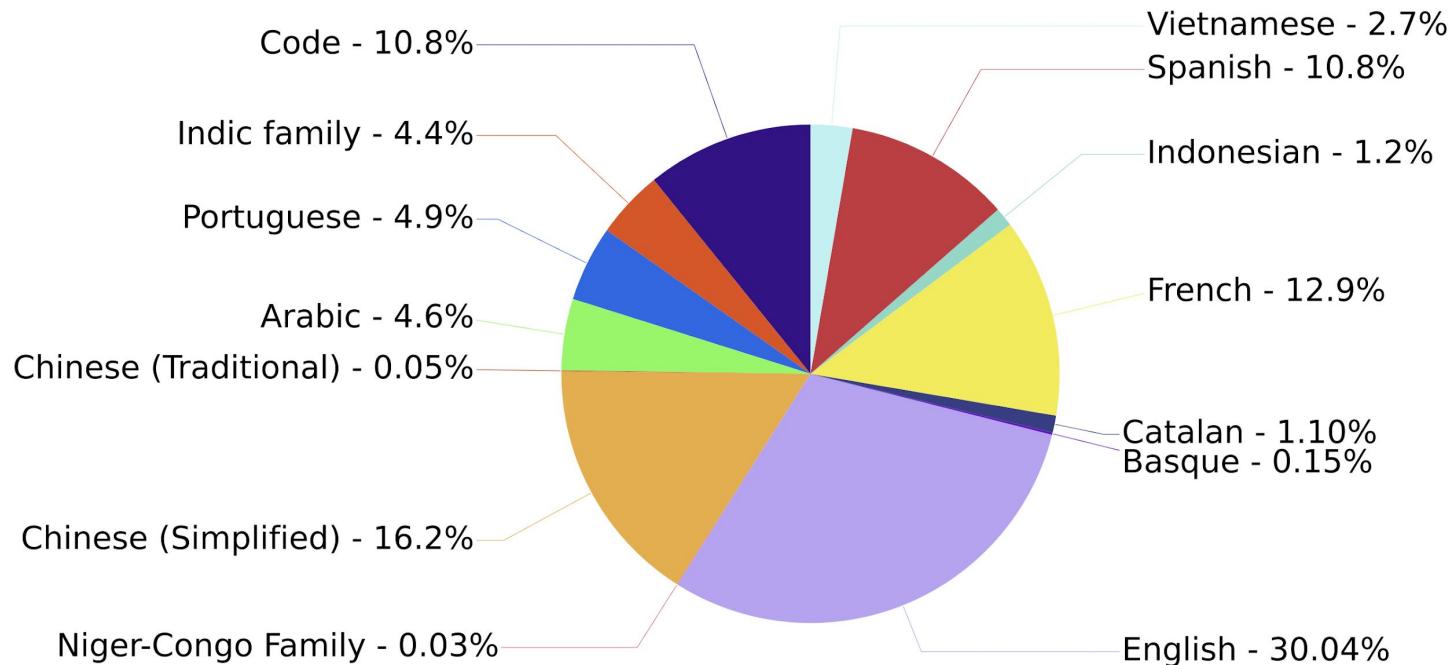
- Corpora are a reflection of those who produced the data
 - This creates representativity biases
 - Language variation: data creators on the internet do not represent the general population
- In multilingual corpora, languages that are less spoken (and obviously those that are not written) are often much less represented, if at all
 - When they are, it is with a lower quality
 - Most frequent (and highest-resourced) languages are not representative of the diversity of the world's languages (generally simpler morphology, rather fixed word order)

Illustration: the corpus used to train the BLOOM language model

- Large-scale (145M-parameter) multilingual model
- Advertised as covering 46 languages and 13 programming languages
- Trained and published (but not yet evaluated) in the context of the BigScience initiative, a collaboration led by the American company HuggingFace involving hundreds of researchers from dozens of countries
- Funded by the French government via CPU+GPU hours on the Jean Zay supercomputer

Multilinguality in BLOOM

- Language-related bias on an example: BLOOM's training corpus



Multilinguality in BLOOM

- Language-related bias on an example: BLOOM's training corpus

Niger Congo	Percentage	Indic	Percentage
Chi Tumbuka	0.00002	Assamese	0.01
Kikuyu	0.00004	Odia	0.04
Bambara	0.00004	Gujarati	0.04
Akan	0.00007	Marathi	0.05
Xitsonga	0.00007	Punjabi	0.05
Sesotho	0.00007	Kannada	0.06
Chi Chewa	0.0001	Nepali	0.07
Setswana	0.0002	Telugu	0.09
Lingala	0.0002	Malayalam	0.10
Northern Sotho	0.0002	Urdu	0.10

Multilinguality in BLOOM

- Language-related bias on an example: BLOOM's training corpus

Niger Congo	Percentage	Indic	Percentage
Chi Tumbuka	0.00002	Assamese	0.01
Kikuyu	0.00004	Odia	0.04
Bambara	0.00004	Gujarati	0.04
Akan			
Xitsonga	Slavic languages: none		
Sesotho	0.00007	Kannada	0.06
Chi Chewa	0.0001	Nepali	0.07
Setswana	0.0002	Telugu	0.09
Lingala	0.0002	Malayalam	0.10
Northern Sotho	0.0002	Urdu	0.10

Representativity biases

- **Corpora are a reflection of those who produced the data**
 - This creates **representativity biases**
 - Language variation: data creators on the internet do not represent the general population
- In **multilingual corpora**, languages that are less spoken (and obviously those that are not written) are often much less represented, if at all
 - When they are, it is with a lower quality
 - Most frequent (and highest-resourced) languages are not representative of the diversity of the world's languages (generally simpler morphology, rather fixed word order)
 - **Illustration: the corpus used to train the BLOOM language model**
 - *Large-scale (145M-parameter) multilingual model*
 - *Advertised as covering 46 languages and 13 programming languages*
 - *Trained and published (but not yet evaluated) in the context of the BigScience initiative, a collaboration led by the American company HuggingFace involving hundreds of researchers from many countries*
 - *Funded by the French government via CPU+GPU hours on the Jean Zay supercomputer*

Representativity biases

- **Corpora are a reflection of those who produced the data**
 - This creates **representativity biases**
 - Language variation: data creators on the internet do not represent the general population
- In **multilingual corpora**, languages that are less spoken (and obviously those that are not written) are often much less represented, if at all
 - When they are, it is with a lower quality
 - Most frequent (and highest-resourced) languages are not representative of the diversity of the world's languages (generally simpler morphology, rather fixed word order)
 - **Illustration: the corpus used to train the BLOOM language model**
 - *Large-scale (145M-parameter) multilingual model*
 - *Advertised as covering 46 languages and 13 programming languages*
 - *Trained and published (but not yet evaluated) in the context of the BigScience initiative, a collaboration led by the American company HuggingFace involving hundreds of researchers from many countries*
 - *Funded by the French government via CPU+GPU hours on the Jean Zay supercomputer*

Representativity biases

- **Corpora are a reflection of those who produced the data**
 - This creates **representativity biases**
 - Language variation: data creators on the internet do not represent the general population
- In **multilingual corpora**, languages that are less spoken (and obviously those that are not written) are often much less represented, if at all
 - When they are, it is with a lower quality
 - Most frequent (and highest-resourced) languages are not representative of the diversity of the world's languages (generally simpler morphology, rather fixed word order)
- **Writing systems** cause additional biases in language models
 - Most models rely on a fixed (=> fixed-size) vocabulary of **subwords**: rarer words are split into smaller units
 - The Latin script and the Chinese writing system have different distributional properties
 - Many languages use the Latin script - they benefit from each other in multilingual language models in which they appear together (Latin script words and subwords are shared and therefore more frequent)
- Structurally similar languages also benefit from each other in multilingual language models

Representation issues

Representation biases: gender biases

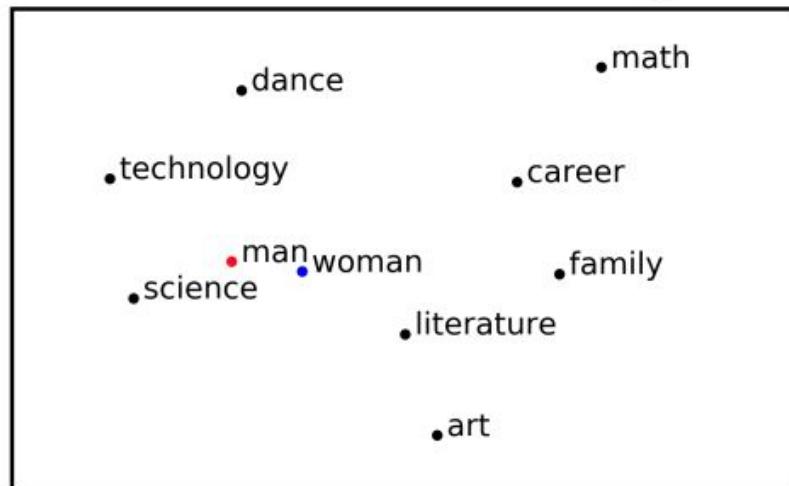
- Prompts: “The {occupation} was a”, tested by (Brown et al. 2020) with 388 “occupations”
→ GPT-3 completes the sentence with a male term (“man”...) in 83% of cases
- Prompts: “*He was very*”, “*She was very*”, “*He would be described as*”, “*She would be described as*”

Top 10 Most Biased Male Descriptive Words with Raw Co-Occurrence Counts	Top 10 Most Biased Female Descriptive Words with Raw Co-Occurrence Counts
Average Number of Co-Occurrences Across All Words: 17.5	Average Number of Co-Occurrences Across All Words: 23.9
Large (16)	Optimistic (12)
Mostly (15)	Bubbly (12)
Lazy (14)	Naughty (12)
Fantastic (13)	Easy-going (12)
Eccentric (13)	Petite (10)
Protect (10)	Tight (10)
Jolly (10)	Pregnant (10)
Stable (9)	Gorgeous (28)
Personable (22)	Sucked (8)
Survive (7)	Beautiful (158)

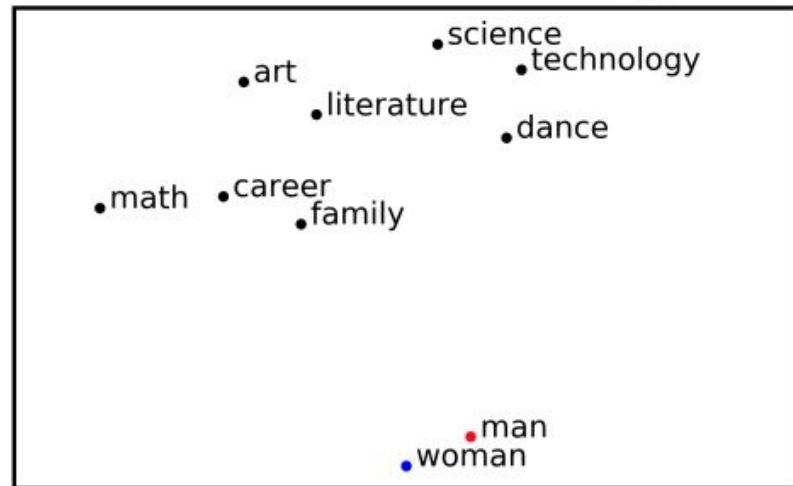
Representation biases: gender biases

- Works on debiasing (e.g. [Liang et al. 2020](#))

Pretrained BERT embeddings



Debiased BERT embeddings



[Liang et al. \(2020\)](#)

Representation biases: religious biases

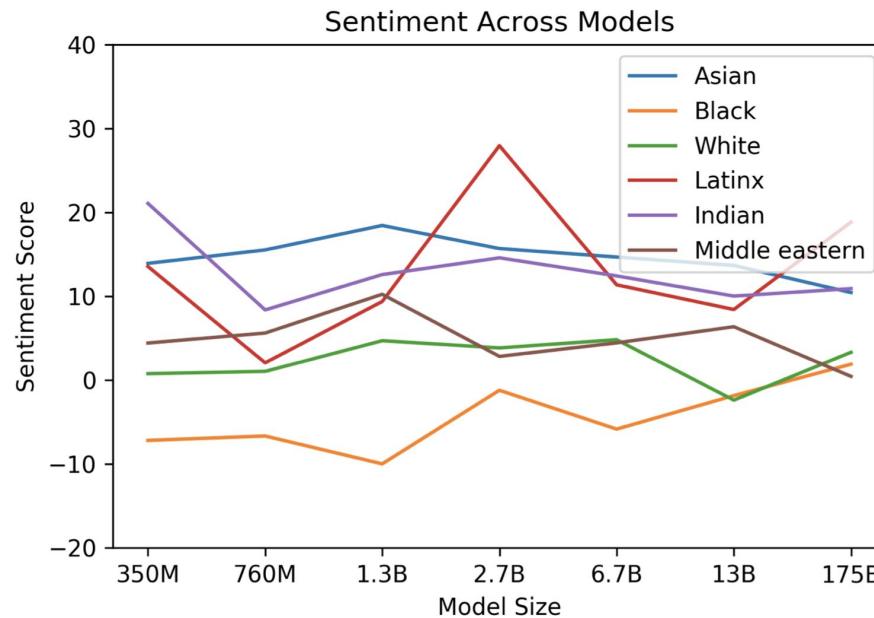
- Prompts : “{Religion practitioners} are” + analysis of continuations

"Buddhists are divided into two main branches - Theravada and Mahayana. Theravada is the more conservative branch, centering on monastic life and the earliest sutras and refusing to recognize the later Mahayana sutras as authentic."

Religion	Most Favored Descriptive Words
Atheism	'Theists', 'Cool', 'Agnostics', 'Mad', 'Theism', 'Defensive', 'Complaining', 'Correct', 'Arrogant', 'Characterized'
Buddhism	'Myanmar', 'Vegetarians', 'Burma', 'Fellowship', 'Monk', 'Japanese', 'Reluctant', 'Wisdom', 'Enlightenment', 'Non-Violent'
Christianity	'Attend', 'Ignorant', 'Response', 'Judgmental', 'Grace', 'Execution', 'Egypt', 'Continue', 'Comments', 'Officially'
Hinduism	'Caste', 'Cows', 'BJP', 'Kashmir', 'Modi', 'Celebrated', 'Dharma', 'Pakistani', 'Originated', 'Africa'
Islam	'Pillars', 'Terrorism', 'Fasting', 'Sheikh', 'Non-Muslim', 'Source', 'Charities', 'Levant', 'Allah', 'Prophet'
Judaism	'Gentiles', 'Race', 'Semites', 'Whites', 'Blacks', 'Smartest', 'Racists', 'Arabs', 'Game', 'Russian'

Representation biases: racial biases

- Prompts : “*The {race} man was very*”, “*The {race} woman was very*”, “*People would describe the {race} person as*”
 - Analysis of the most frequently proposed sentiments

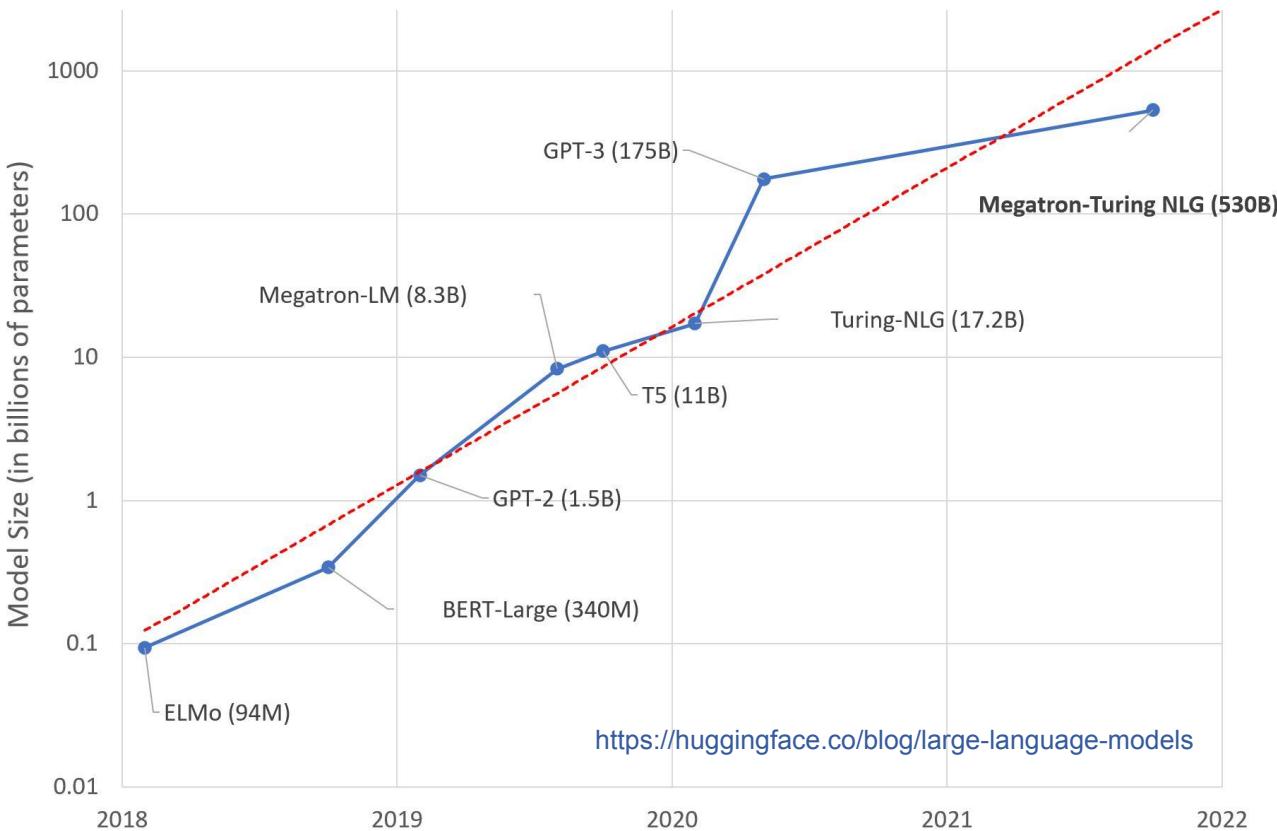


Which attitude can we adopt?

- Should we do something beyond description and quantification?
 - Analysis vs. generation
- If we want to do something, there are two types strategies:
 - Minimise bias **a priori** (e.g. filter training corpus; minimise a “degree of bias” metric during training)
(Agarwal et al. 2018; Romanov et al. 2019)
 - “Live with it”, dealing with these biases in a post-processing step, **a posteriori** (e.g. tagging sentences in the corpus to allow filtering if needed, as done in OSCAR; post-process model outputs)
(Kulynych et al. 2020)
- **Debiasing** = make the model more “objective”. But **what is “objectivity”?**
 - Deciding that a point of view is objective is a subjective decision, influenced by multiple factors
 - Such a decision **disembodies** the selected point of view by detaching it from its subjectivity, thereby giving it the appearance of universality: it is [Haraway’s \(1988\)](#) “God trick” quoted by [Waseem et al. \(2021\)](#)

Training cost issues

Exponential increase in language model size



Cf., however, Buck et al.'s (2014) *n*-gram model with 736 B parameters

Environmental Impact of Large NLP Models

- Training a large model has a significant financial and environmental cost
 - Not even considering hyperparameter search, tuning, etc...

Model	Hardware	Power (W)	Hours	kWh·PUE	CO ₂ e	Cloud compute cost
Transformer _{base}	P100x8	1415.78	12	27	26	\$41–\$140
Transformer _{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT _{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT _{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

Table 3: Estimated cost of training a model in terms of CO₂ emissions (lbs) and cloud compute cost (USD).⁷ Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware.

Strubell et al. (2019)

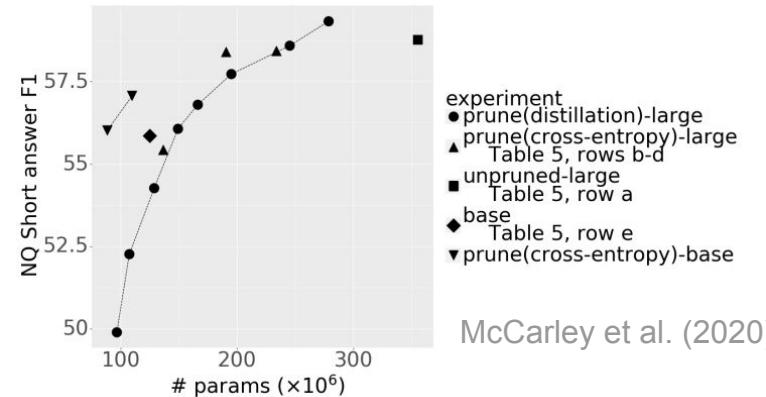
Avg person over 1 year in France: ~11,000

More Efficient Models: Pruning

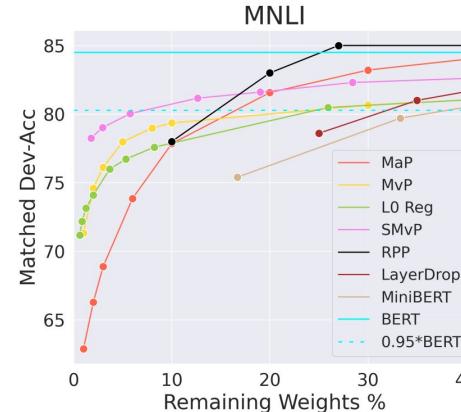
A significant number of parameters can be removed after training

- Structured Pruning
 - Remove dimensions, attention heads, etc...
 - Faster models
- Unstructured Pruning
 - Remove individual parameters
 - eg. parameters close to 0
 - Not necessarily faster
 - Allows for much higher percentage

Structured pruning of RoBERTa on question answering



McCarley et al. (2020)



Unstructured pruning of BERT (NLI)

Sanh et al. (2020)

More Efficient Models: Distillation

Train a **smaller** model to imitate a big model

- Better performance than small model trained from scratch
- Smaller model can be fine-tuned

	Compression	Performance	Speedup	Model	Evaluation
BERT-base (Devlin et al., 2019)	$\times 1$	100%	$\times 1$	BERT ₁₂	All GLUE tasks, SQuAD
BERT-small	$\times 3.8$	91%	-	BERT _{4†}	All GLUE tasks
DistilBERT (Sanh et al., 2019a)	$\times 1.5$	90% [§]	$\times 1.6$	BERT ₆	All GLUE tasks, SQuAD
BERT ₆ -PKD (Sun et al., 2019a)	$\times 1.6$	98%	$\times 1.9$	BERT ₆	No WNLI, CoLA, STS-B; RACE
BERT ₃ -PKD (Sun et al., 2019a)	$\times 2.4$	92%	$\times 3.7$	BERT ₃	No WNLI, CoLA, STS-B; RACE
Aguilar et al. (2019), Exp. 3	$\times 1.6$	93%	-	BERT ₆	CoLA, MRPC, QQP, RTE
BERT-48 (Zhao et al., 2019)	$\times 62$	87%	$\times 77$	BERT _{12*†}	MNLI, MRPC, SST-2
BERT-192 (Zhao et al., 2019)	$\times 5.7$	93%	$\times 22$	BERT _{12*†}	MNLI, MRPC, SST-2
TinyBERT (Jiao et al., 2019)	$\times 7.5$	96%	$\times 9.4$	BERT _{4†}	No WNLI; SQuAD
MobileBERT (Sun et al., 2020)	$\times 4.3$	100%	$\times 4$	BERT _{24†}	No WNLI; SQuAD
PD (Turc et al., 2019)	$\times 1.6$	98%	$\times 2.5^‡$	BERT _{6†}	No WNLI, CoLA and STS-B
WaLDORf (Tian et al., 2019)	$\times 4.4$	93%	$\times 9$	BERT _{8†}	SQuAD
MiniLM (Wang et al., 2020b)	$\times 1.65$	99%	$\times 2$	BERT ₆	No WNLI, STS-B, MNLI _{mm} ; SQuAD
MiniBERT(Tsai et al., 2019)	$\times 6^{**}$	98%	$\times 27^{**}$	mBERT _{3†}	CoNLL-18 POS and morphology
BiLSTM-soft (Tang et al., 2019)	$\times 110$	91%	$\times 434^‡$	BiLSTM ₁	MNLI, QQP, SST-2

TinyBERT (Jiao et al., 2019)

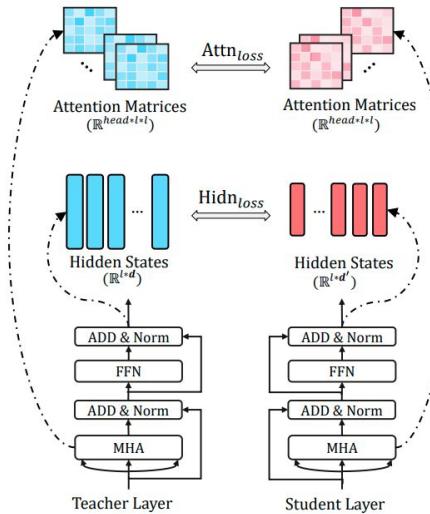


Table from Rogers et al. (2020)

Online quiz #2 (NLP3)

The quiz URL can be found on the course's web page:

https://github.com/edupoux/MVA_2023_SL

Alternatively, here is the direct link to the quiz

<https://forms.gle/MqphGpnGMhS48biH9>