Neural architectures for textual data: two healthcare use cases

Mohammed SEDKI

mohammed.sedki@gustaveroussy.fr

Inserm team OncoStat & Gustave Roussy
Université Paris-Saclay

April 7, 2025

Outline

1. Classical supervised learning problem
2. From unstructured data to tabular data

Sentiment valence detection in child psychiatry

- free texts written by teenagers as part of a study on intra-family interactions.
- 1648 sentences labeled by Dr. Éric Brunet-Gouet, psychiatrist at Versailles hospital.
- model for automatic labeling (sentiment analysis + detection of the people in the text)

Using a self-attention architecture to automate valence categorization of French teenagers' free descriptions of their family relationships: a proof of concept
Journal of Medical Artificial Intelligence, September 2022

The head of the dataset

```
jf+,js+ J'ai 2 frères et deux sœurs et je m'entend plutôt bien avec.
mp-     Ma maman est séparée de mon père.
jm+     Je m'entend très bien avec ma mère.
si     J'ai deux grandes demi-sœur du côté de mon père.
mi,pi Ma maman à fait cinq enfants, mon père en n'a fait trois.
jm+     Avec ma mère ont aime bien faire les magasins.
```

The dataset : 1648 sentences and 11 labels

- Relational valence: **+**, **-** and **0**. Positive relationships refer to good understanding, the expression of positive affect, and cooperation. Negative relationships correspond to conflicts, disagreements, and the absence of a normal relationship.
- In the absence of information on valence, the text is considered informative (i) about people's habits or living conditions.
- The subjects described in a sentence have been labeled as follows : *le répondant **je**, la **m**ère, le **p**ère, la **s**oeur, le **f**rère, la **fa**mille et une **t**ierce personne*.

Training-test partition sizes 1318 and 330.

Multilabel classification problem

- Penalized regression models
- Decision trees
- Random forests and gradient boosting
- Deep neural networks
- Support vector machines

We have $y \in \{0,1\}^{11}$ but we need a vector of covariates $x \in \bigtimes_{j=1}^{d} \mathcal{X}_j$.

Before BERT model (2018)

- *tf*: number of occurrences of a term in a document.
- *idf*: logarithm of the inverse of the proportion of documents in the corpus that contain the term t:

  $$idf_t = \log\left(\frac{|D|}{|\{d_j : t \in d_j\}|}\right)$$

  where $|D|$ represents the total number of documents in the corpus and $|\{d_j : t_i \in d_j\}|$ the number of documents in which the term t appears.
- Term-document inverse matrix:

  $$tfidf_{t,d} = tf_{t,d}.idf_t$$

The vector of tf-idf of a sentence is invariant under the permutation of words in the sentence

Learning strategies

- Embedding sentences into numerical vectors using TF-IDF + classification using classical model
- Embedding sentences using LLM without fine-tuning + classification using classical model
- Enhance LLM with a fully connected layer and a final layer adapted to the multilabel classification problem, then perform fine-tuning.

BERT : may be the first language model

- Bidirectional Encoder Representations from Transformers : it was among the first neural architectures that could be called Large Language Models (with GPT-1).
- Let's give an overview of the main ingredients of these models as well as the fundamental building block.

Language Modeling

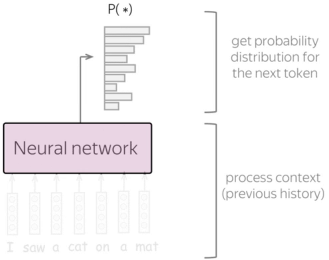
- LM as probability distribution estimation over sequences of tokens/words (generative models)
- $\mathbb{P}\Big(\text{the, mouse, ate, the, cheese}\Big) = 0.02$
- $\mathbb{P}\Big(\text{the, the, mouse, ate, the, cheese}\Big) = 0.0001$ (Syntactic knowledge)
- $\mathbb{P}\Big(\text{the, cheese, ate, the, mouse}\Big) = 0.001$ (Semantic knowledge)

Autoregressive (AR) language models

- $\mathbb{P}(x_1, \ldots, x_\ell) = \mathbb{P}(x_1)\mathbb{P}(x_2|x_1)\mathbb{P}(x_3|x_2, x_1)\ldots\mathbb{P}(x_\ell|x_{\ell-1}\ldots, x_1)$.
- To generate a sequence, it is enough to know how to generate according to each of the conditional distributions of the previous product.
- You only need a model that can predict the next token given past context!
- How: Train a neural network to predict them.

Intuitively, two things

- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way

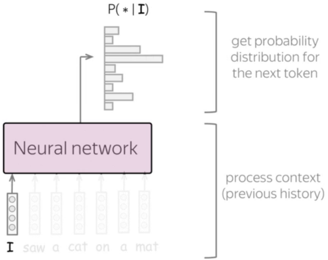


P(∗)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way

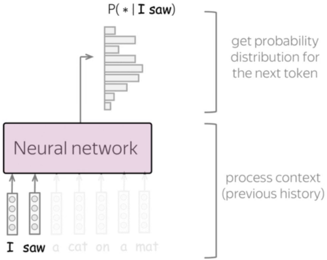


P(* | I)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way

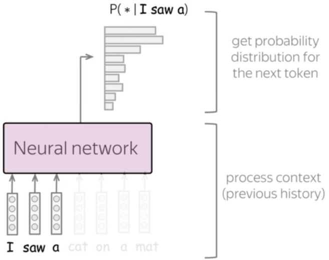


P(* | I saw)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

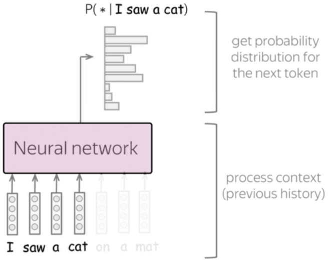
- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way


P(∗ | I saw a)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

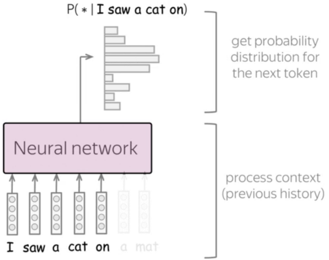
- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way


P(∗ | I saw a cat)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

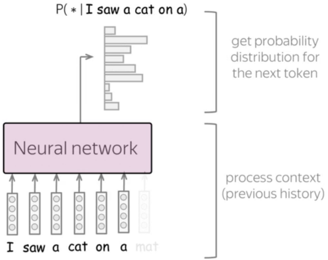
- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way


P(* | I saw a cat on)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

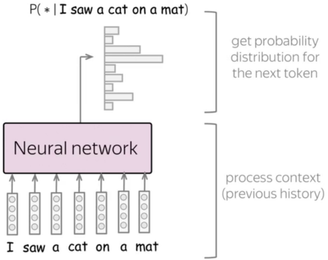
- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way


P(∗ | I saw a cat on a)
get probability distribution for the next token
Neural network
process context (previous history)
I
saw
a
cat
on
a
mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Intuitively, two things

- process context : to get a **vector representation** for the previous context → The model predicts a probability distribution for the next token.
- generate a probability distribution for the next token → model-agnostic Once a **context has been encoded**, the probability distribution is generated in the same way

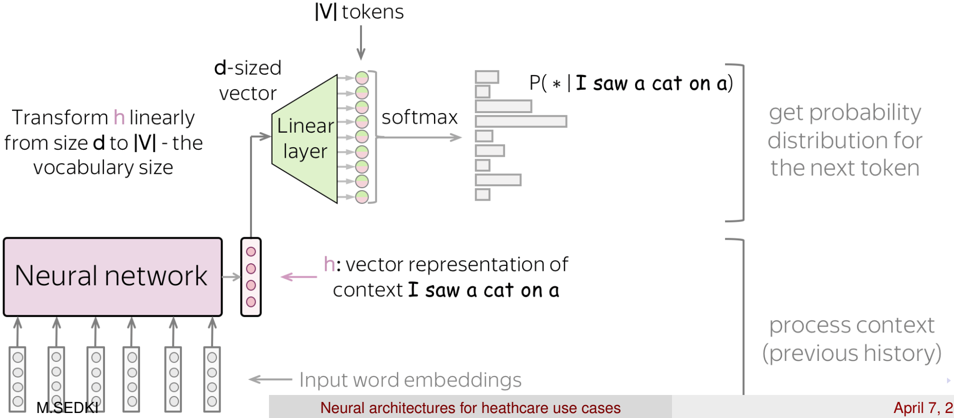


P(* | I saw a cat on a mat)
get probability distribution for the next token
Neural network
process context (previous history)
I saw a cat on a mat

The generated text is some path of a random process over the vocabulary set, where each element represents a token or word, and the conditional distributions act as classification rules.

Pipeline

1. feed word embedding for previous (context) words into a network
2. get vector representation of context from the network
3. from this vector representation, predict a probability distribution for the next token



|V| tokens
d-sized vector
Transform h linearly from size d to |V| - the vocabulary size
Linear layer
softmax
P(∗ | I saw a cat on a)
get probability distribution for the next token
Neural network
h: vector representation of context I saw a cat on a
process context (previous history)
Input word embeddings

Train and cross-entropy loss

- Formally, if $x_1, \ldots, x_\ell$ is a training token sequence, then at the timestep t a model predicts a probability distribution $p = \mathbb{P}\big(\cdot \mid x_1, \ldots, x_{t-1}\big)$. The target probability distribution p^* at this step is a vecteur of length $|V|$ given by $p^* = \text{one-hot}(x_t)$, i.e., we want a model to assign probability 1 to the correct token, x_t, and zero to the rest.

- The standard loss function in classification is the cross-entropy loss. Cross-entropy loss for the target distribution p^* and the predicted distribution p is

$$-\sum_{i=1}^{|V|} p_i^* \log(p_i) = -\log\left(p_{x_t}\right) = -\log\Big(\mathbb{P}\big(x_t \mid x_1, \ldots, x_{t-1}\big)\Big)$$

since only one of p_i^* is non-zero (for the correct token x_t).

Equivalent to Kullback-Leibler divergence $D_{\text{KL}}(p^* \| p)$.

The model evaluation during training

Let us assume we have a held-out text sequence $x_{1:M} = (x_1, x_2, \ldots, x_M)$. Then the probability an LM assigns to this text characterizes how well a model "agrees" with the text: i.e., how well it can predict appearing tokens based on their contexts:

$$\mathcal{L}(x_1, x_2, \ldots, x_M) = \log\left(\mathbb{P}(x_1)\right) + \sum_{t=2}^{M} \log \mathbb{P}(x_t | x_{1:t-1}).$$

For model evaluation, we use **perplexity**

$$\text{perplexity}(x_1, x_2, \ldots, x_M) = \exp\left\{-\frac{1}{M}\mathcal{L}(x_1, x_2, \ldots, x_M)\right\}.$$

Perplexity will always be between 1 and $|V|$

- The **best** perplexity is 1. If our model is perfect and assigns probability 1 to correct tokens (the ones from the text), then the log-probability is zero, and the perplexity is 1.
- The **worst** perplexity is $|V|$. In the worst case, LM knows absolutely nothing about the data: it thinks that all tokens have the same probability $\frac{1}{|V|}$ regardless of context.

Autoregressive (AR) language models

- Task : predict the next word/token
- Steps :
  1. tokenize
  2. forward
  3. predict probability of next token
  4. sample
  5. detokenize

Tokenizer

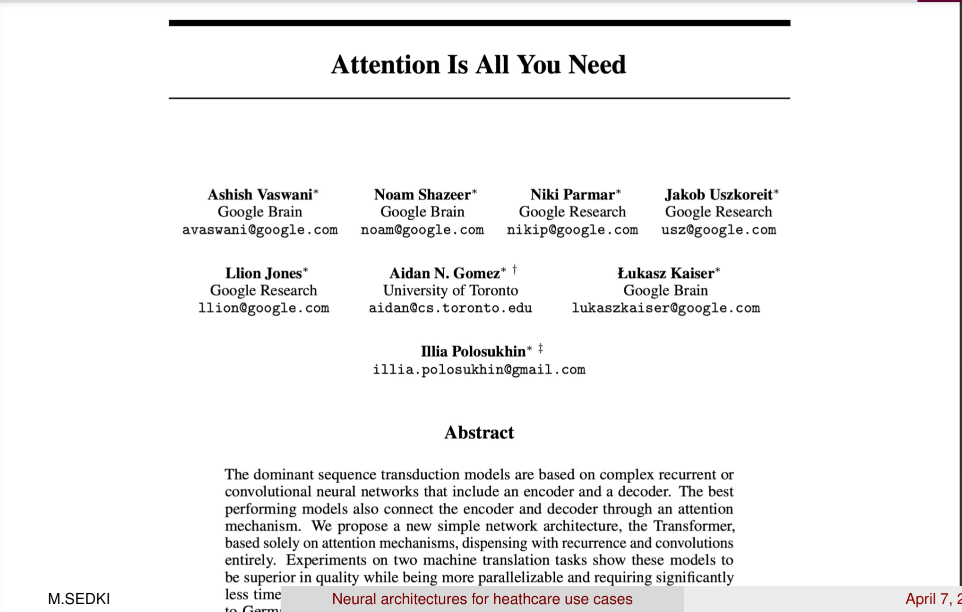
- Why ?
  1. More general than words (eg typos)
  2. Shorter sequences than with characters
- The idea is to keep frequently used words whole, but to cut out words that have been transformed by grammar : tokens as common subsequences of characters (3 or 4 letters)
- Eg: Byte Pair Encoding (BPE). Train (very very time consuming) steps :
  1. Take large corpus of text
  2. Start with one token per character
  3. Merge common pairs of tokens into a token

Popular tokenizers available at : tiktokenizer.vercel.app

Visualisation

Thanks to **Grant Sanderson** for his **manim** python library and youtube channel **3Blue1Brown** who made it possible to visualise self-attention working.

The basic building block



Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

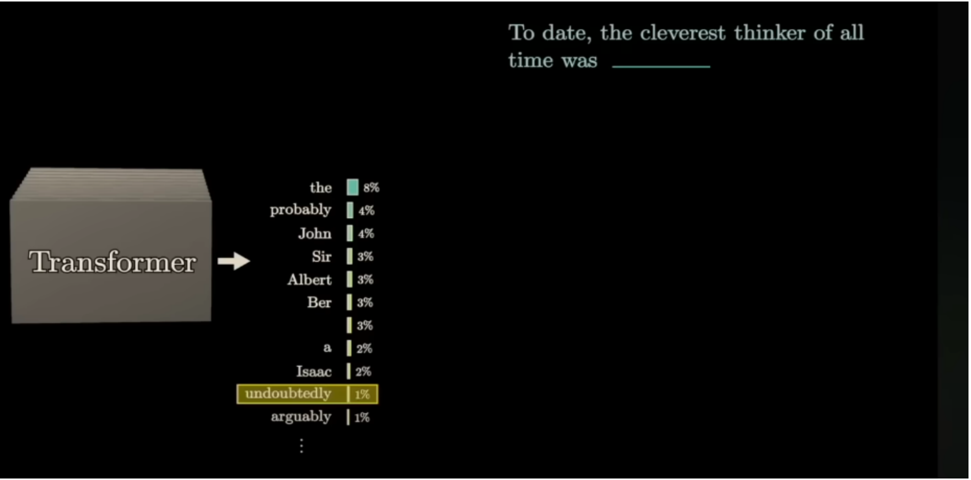
Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Abstract

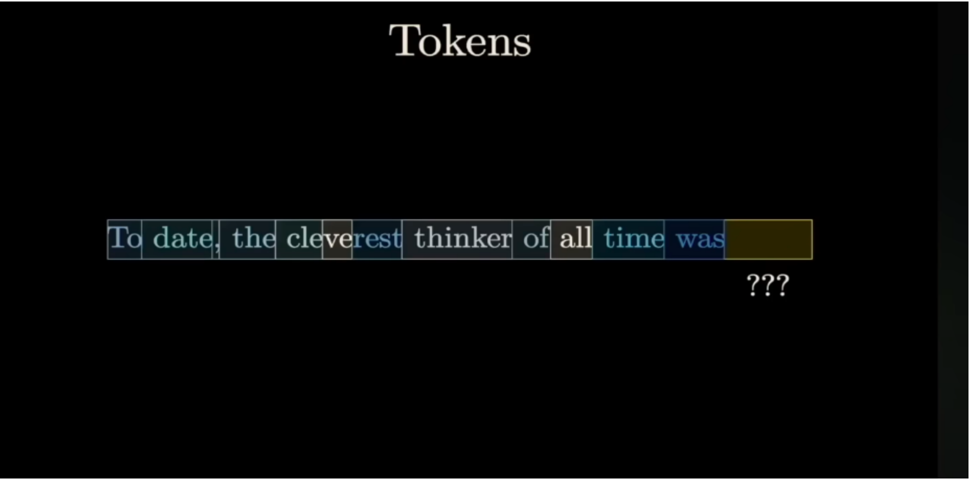
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time
to Germ

One forward-pass



To date, the cleverest thinker of all time was ______

Tokens

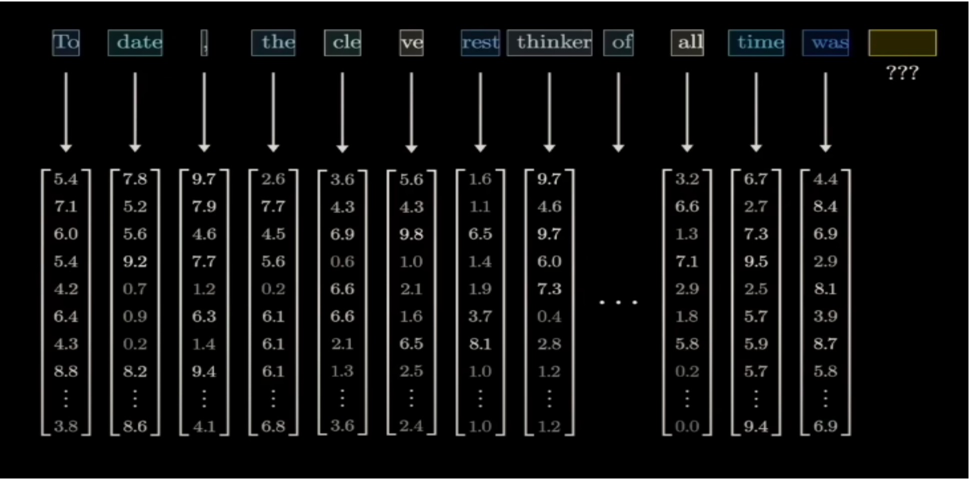


Tokens

To date, the cleverest thinker of all time was

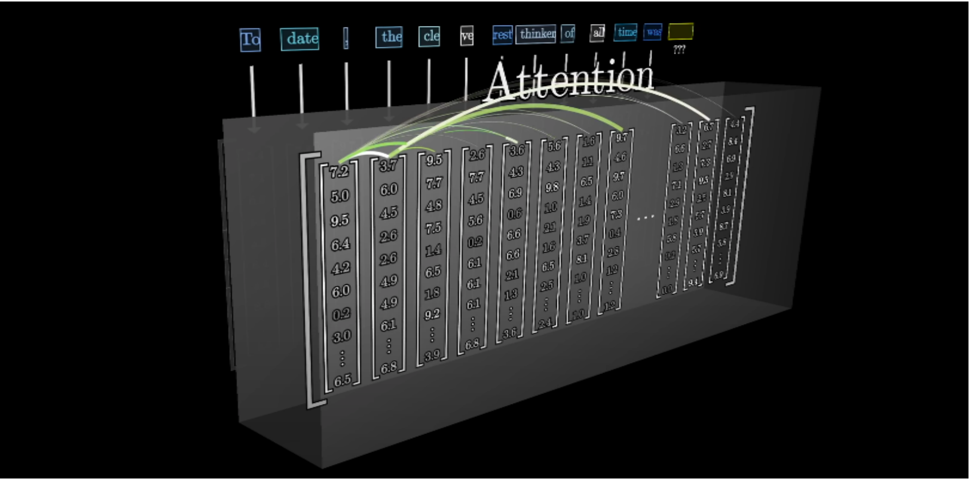
???

Initial embedding: parameters to estimate

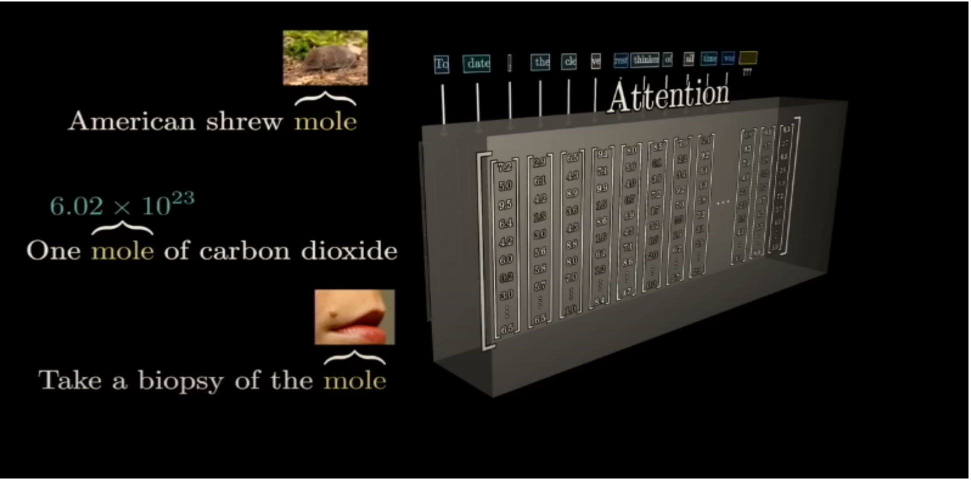


To	date	,	the	cle	ve	rest	thinker	of	all	time	was	???
5.4	7.8	9.7	2.6	3.6	5.6	1.6	9.7		3.2	6.7	4.4	
7.1	5.2	7.9	7.7	4.3	4.3	1.1	4.6		6.6	2.7	8.4	
6.0	5.6	4.6	4.5	6.9	9.8	6.5	9.7		1.3	7.3	6.9	
5.4	9.2	7.7	5.6	0.6	1.0	1.4	6.0		7.1	9.5	2.9	
4.2	0.7	1.2	0.2	6.6	2.1	1.9	7.3		2.9	2.5	8.1	
6.4	0.9	6.3	6.1	6.6	1.6	3.7	0.4	$\cdots$	1.8	5.7	3.9	
4.3	0.2	1.4	6.1	2.1	6.5	8.1	2.8		5.8	5.9	8.7	
8.8	8.2	9.4	6.1	1.3	2.5	1.0	1.2		0.2	5.7	5.8	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮	
3.8	8.6	4.1	6.8	3.6	2.4	1.0	1.2		0.0	9.4	6.9	

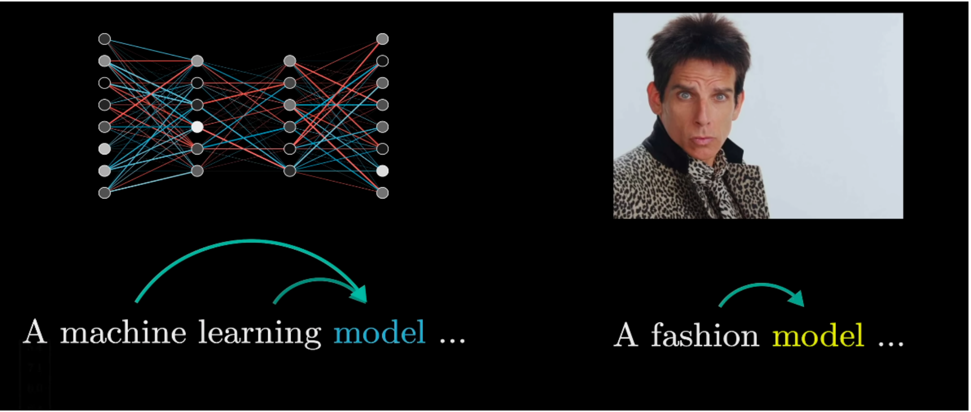
Communication between initial embeddings



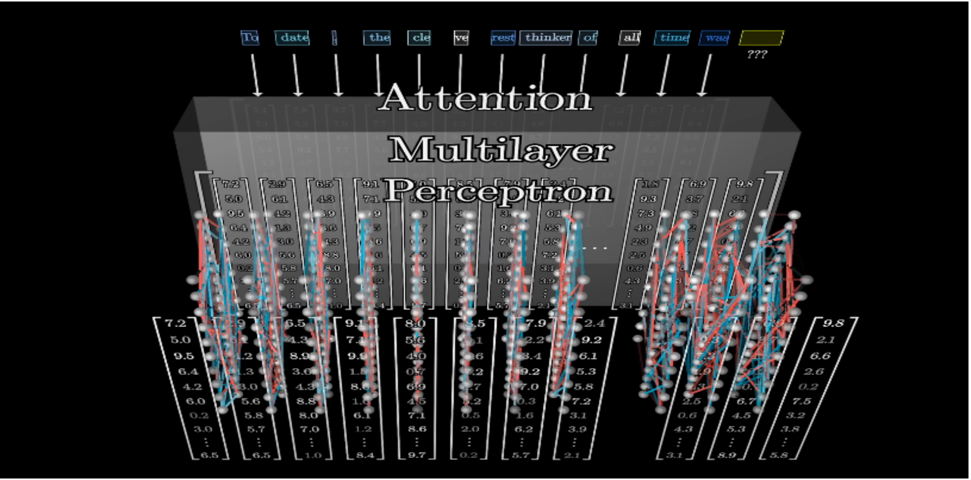
Communication between initial embeddings


American shrew mole
6.02×10^{23}
One mole of carbon dioxide
Take a biopsy of the mole
To date the cle ve rest thinker of all time was
???
Attention

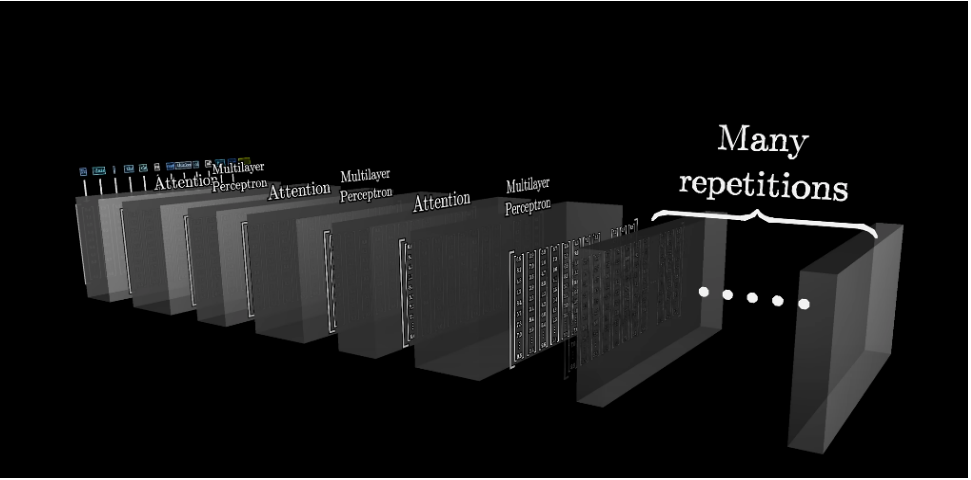
Communication between initial embeddings



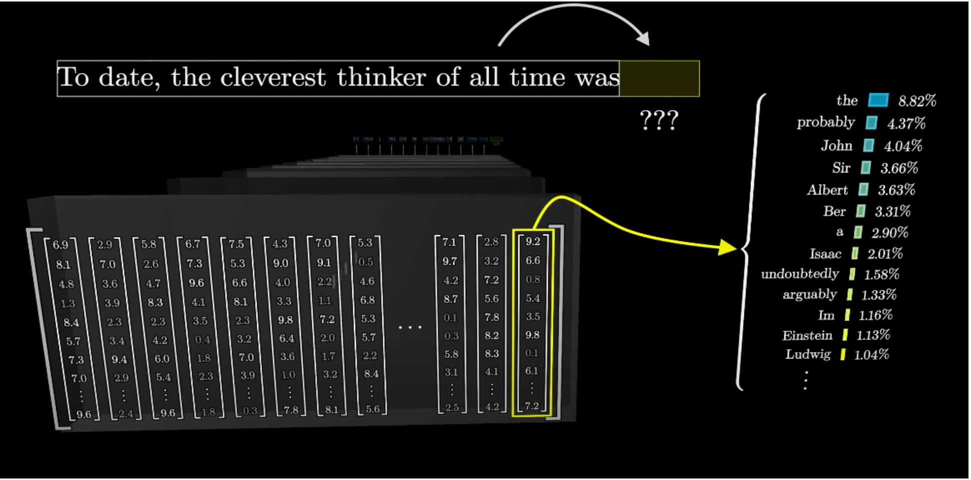
Communication + feed forward



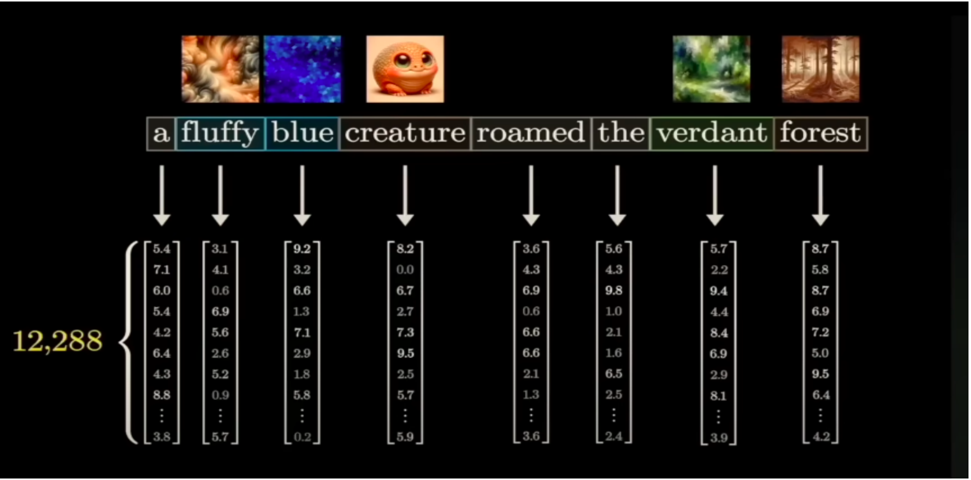
Repetitions of attention



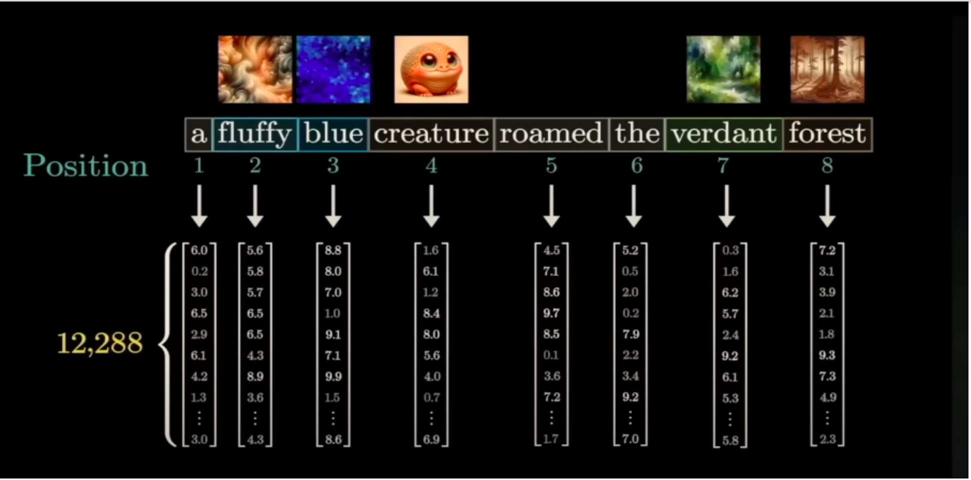
After many repetitions



Communication between embeddings



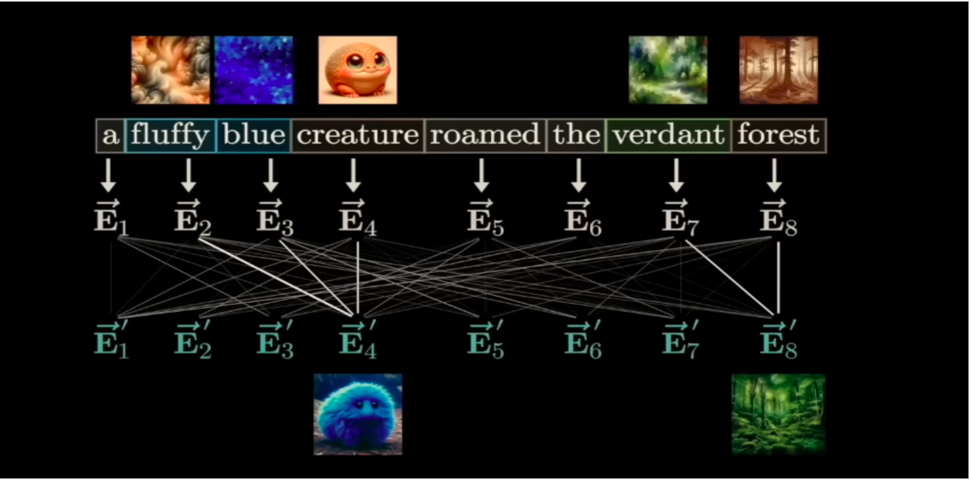
Fixed positional encodings



a	fluffy	blue	creature	roamed	the	verdant	forest
1	2	3	4	5	6	7	8
6.0	5.6	8.8	1.6	4.5	5.2	0.3	7.2
0.2	5.8	8.0	6.1	7.1	0.5	1.6	3.1
3.0	5.7	7.0	1.2	8.6	2.0	6.2	3.9
6.5	6.5	1.0	8.4	9.7	0.2	5.7	2.1
2.9	6.5	9.1	8.0	8.5	7.9	2.4	1.8
6.1	4.3	7.1	5.6	0.1	2.2	9.2	9.3
4.2	8.9	9.9	4.0	3.6	3.4	6.1	7.3
1.3	3.6	1.5	0.7	7.2	9.2	5.3	4.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3.0	4.3	8.6	6.9	1.7	7.0	5.8	2.3

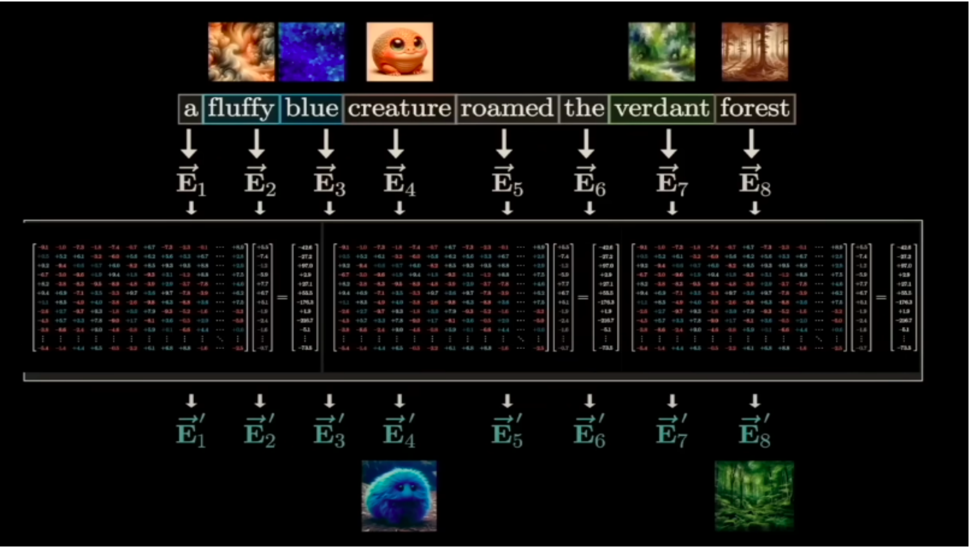
$$\mathrm{PE}_{pos,2i} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \text{ and } \mathrm{PE}_{pos,2i+1} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

We want to update embeddings

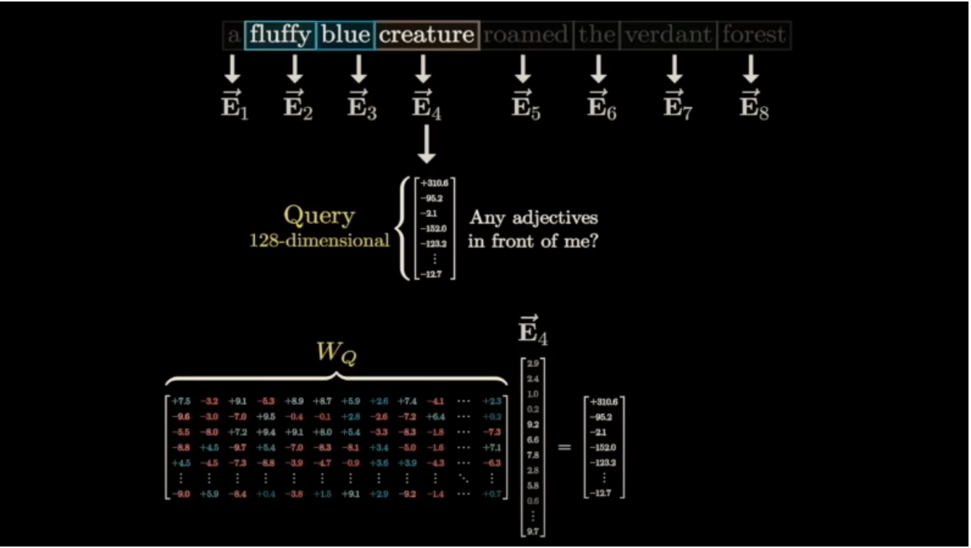


a
fluffy
blue
creature
roamed
the
verdant
forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
$\vec{E}_1'$ $\vec{E}_2'$ $\vec{E}_3'$ $\vec{E}_4'$ $\vec{E}_5'$ $\vec{E}_6'$ $\vec{E}_7'$ $\vec{E}_8'$

We need only matrix multiplication


a
fluffy
blue
creature
roamed
the
verdant
forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
$\vec{E}_1'$ $\vec{E}_2'$ $\vec{E}_3'$ $\vec{E}_4'$ $\vec{E}_5'$ $\vec{E}_6'$ $\vec{E}_7'$ $\vec{E}_8'$

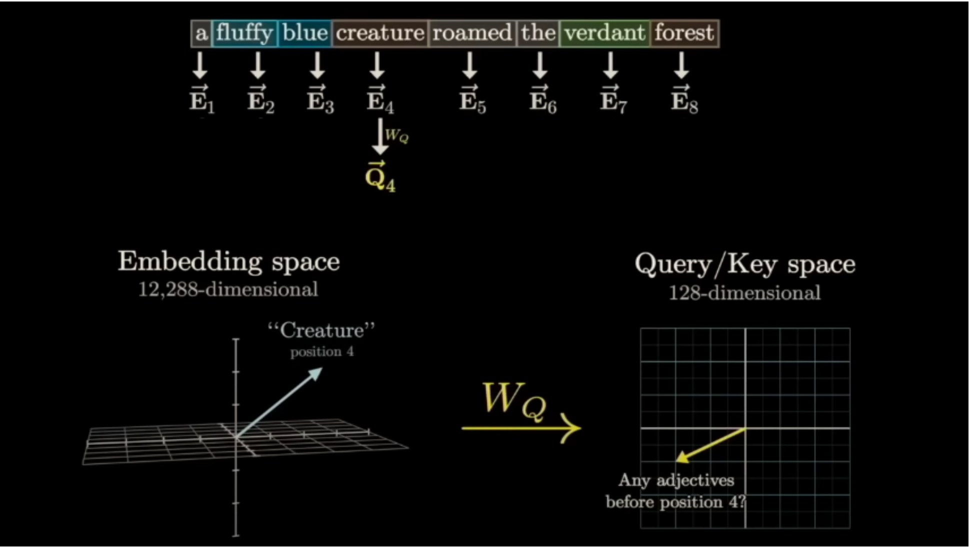
Query matrix W_Q



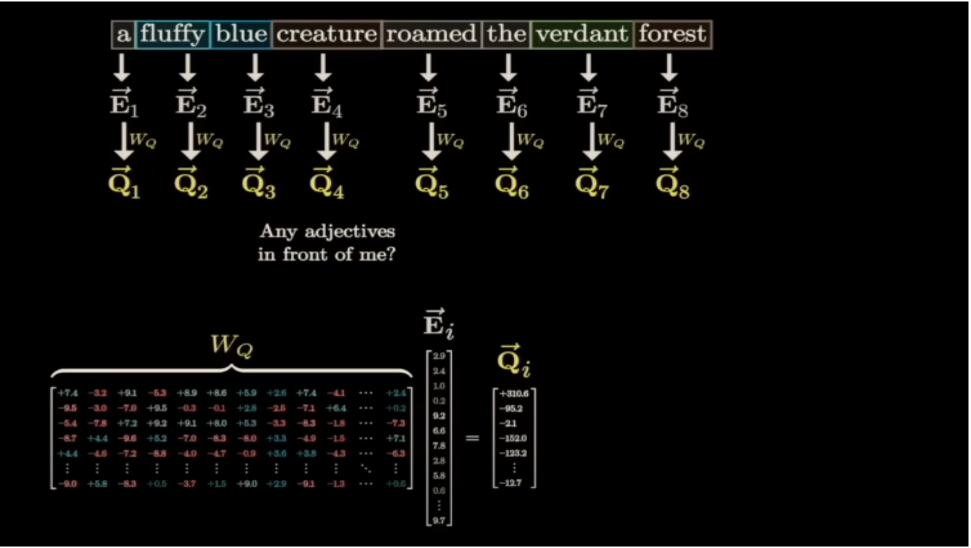
$$\overbrace{\begin{bmatrix} +7.5 & -3.2 & +9.1 & -5.3 & +8.9 & +8.7 & +5.9 & +2.6 & +7.4 & -4.1 & \cdots & +2.3 \\ -9.6 & -3.0 & -7.0 & +9.5 & -0.4 & -0.1 & +2.8 & -2.6 & -7.2 & +6.4 & \cdots & +0.2 \\ -5.5 & -8.0 & +7.2 & +9.4 & +9.1 & +8.0 & +5.4 & -3.3 & -8.3 & -1.8 & \cdots & -7.3 \\ -8.8 & +4.5 & -9.7 & +5.4 & -7.0 & -8.3 & -8.1 & +3.4 & -5.0 & -1.6 & \cdots & +7.1 \\ +4.5 & -4.5 & -7.3 & -8.8 & -3.9 & -4.7 & -0.9 & +3.6 & +3.9 & -4.3 & \cdots & -6.3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -9.0 & +5.9 & -8.4 & +0.4 & -3.8 & +1.5 & +9.1 & +2.9 & -9.2 & -1.4 & \cdots & +0.7 \end{bmatrix}}^{W_Q} \overset{\vec{E}_4}{\begin{bmatrix} 2.9 \\ 2.4 \\ 1.0 \\ 0.2 \\ 9.2 \\ 6.6 \\ 7.8 \\ 2.8 \\ 5.8 \\ 0.6 \\ \vdots \\ 9.7 \end{bmatrix}} = \begin{bmatrix} +310.6 \\ -95.2 \\ -2.1 \\ -152.0 \\ -123.2 \\ \vdots \\ -12.7 \end{bmatrix}$$

$$\text{Query} \left\{ \begin{bmatrix} +310.6 \\ -95.2 \\ -2.1 \\ -152.0 \\ -123.2 \\ \vdots \\ -12.7 \end{bmatrix} \right.$$

Query space


a fluffy blue creature roamed the verdant forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
W_Q
$\vec{Q}_4$
Embedding space
12,288-dimensional
"Creature"
position 4
W_Q
Query/Key space
128-dimensional
Any adjectives
before position 4?

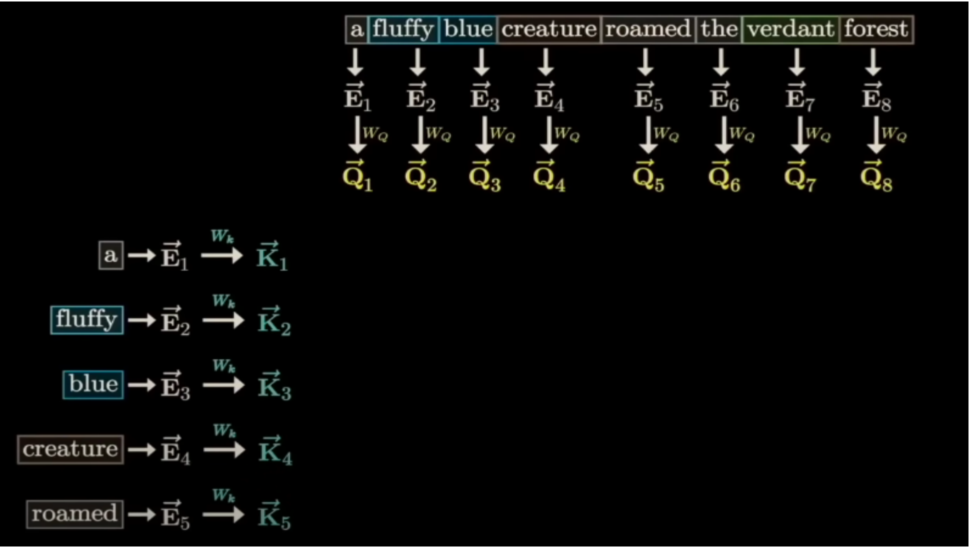
Q_i for each word/token


a
fluffy
blue
creature
roamed
the
verdant
forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
W_Q
$\vec{Q}_1$ $\vec{Q}_2$ $\vec{Q}_3$ $\vec{Q}_4$ $\vec{Q}_5$ $\vec{Q}_6$ $\vec{Q}_7$ $\vec{Q}_8$

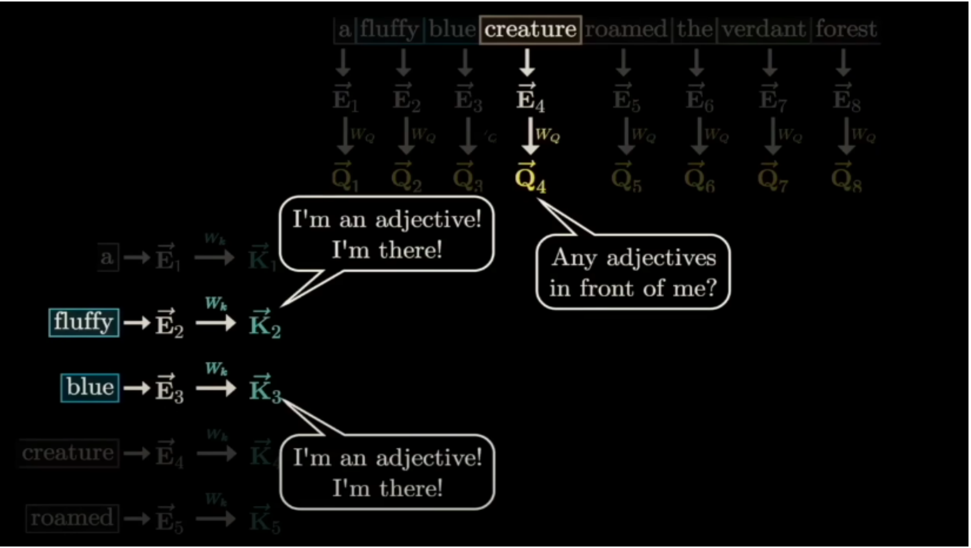
Any adjectives
in front of me?

$$\overbrace{\begin{bmatrix} +7.4 & -3.2 & +9.1 & -5.3 & +8.9 & +8.6 & +5.9 & +2.6 & +7.4 & -4.1 & \cdots & +2.4 \\ -9.5 & -3.0 & -7.0 & +9.5 & -0.3 & -0.1 & +2.8 & -2.5 & -7.1 & +6.4 & \cdots & +0.2 \\ -5.4 & -7.8 & +7.2 & +9.2 & +9.1 & +8.0 & +5.3 & -3.3 & -8.3 & -1.8 & \cdots & -7.3 \\ -8.7 & +4.4 & -9.6 & +5.2 & -7.0 & -8.3 & -8.0 & +3.3 & -4.9 & -1.5 & \cdots & +7.1 \\ +4.4 & -4.6 & -7.2 & -8.8 & -4.0 & -4.7 & -0.9 & +3.6 & +3.8 & -4.3 & \cdots & -6.3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -9.0 & +5.8 & -8.3 & +0.5 & -3.7 & +1.5 & +9.0 & +2.9 & -9.1 & -1.3 & \cdots & +0.6 \end{bmatrix}}^{W_Q} \overset{\vec{E}_i}{\begin{bmatrix} 2.9 \\ 2.4 \\ 1.0 \\ 0.2 \\ 9.2 \\ 6.6 \\ 7.8 \\ 2.8 \\ 5.8 \\ 0.6 \\ \vdots \\ 9.7 \end{bmatrix}} = \overset{\vec{Q}_i}{\begin{bmatrix} +310.6 \\ -95.2 \\ -2.1 \\ -152.0 \\ -123.2 \\ \vdots \\ -12.7 \end{bmatrix}}$$

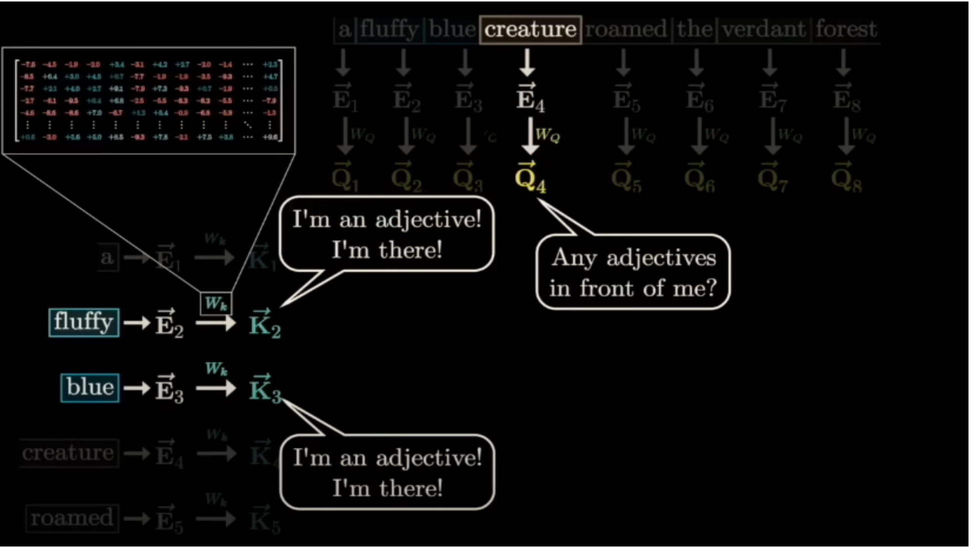
Key matrix W_K


a fluffy blue creature roamed the verdant forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
W_Q
$\vec{Q}_1$ $\vec{Q}_2$ $\vec{Q}_3$ $\vec{Q}_4$ $\vec{Q}_5$ $\vec{Q}_6$ $\vec{Q}_7$ $\vec{Q}_8$
a → $\vec{E}_1 \xrightarrow{W_k} \vec{K}_1$
fluffy → $\vec{E}_2 \xrightarrow{W_k} \vec{K}_2$
blue → $\vec{E}_3 \xrightarrow{W_k} \vec{K}_3$
creature → $\vec{E}_4 \xrightarrow{W_k} \vec{K}_4$
roamed → $\vec{E}_5 \xrightarrow{W_k} \vec{K}_5$

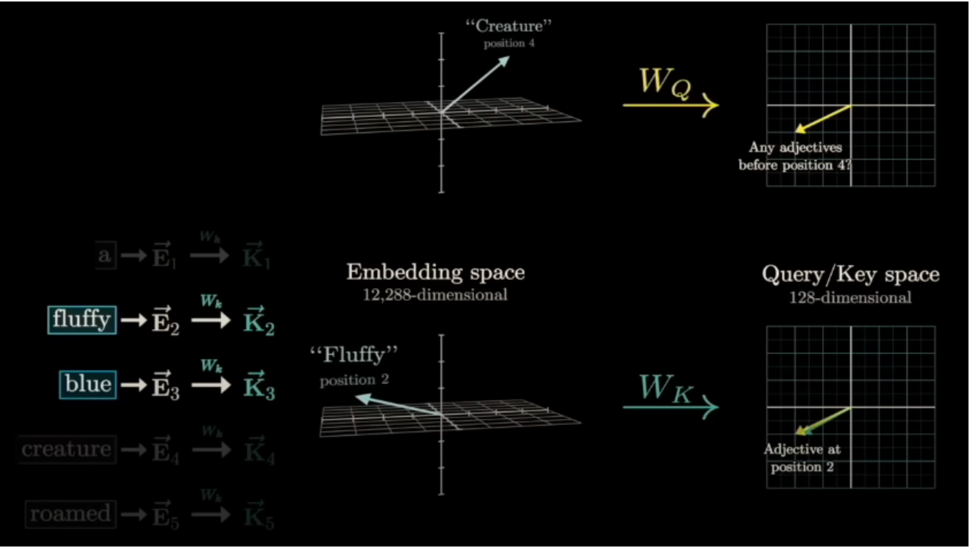
Key as an answer


a | fluffy | blue | creature | roamed | the | verdant | forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
W_Q
$\vec{Q}_1$ $\vec{Q}_2$ $\vec{Q}_3$ $\vec{Q}_4$ $\vec{Q}_5$ $\vec{Q}_6$ $\vec{Q}_7$ $\vec{Q}_8$
Any adjectives in front of me?
I'm an adjective! I'm there!
a → $\vec{E}_1$ $\xrightarrow{W_k}$ $\vec{K}_1$
fluffy → $\vec{E}_2$ $\xrightarrow{W_k}$ $\vec{K}_2$
blue → $\vec{E}_3$ $\xrightarrow{W_k}$ $\vec{K}_3$
creature → $\vec{E}_4$ $\xrightarrow{W_k}$ $\vec{K}_4$
roamed → $\vec{E}_5$ $\xrightarrow{W_k}$ $\vec{K}_5$
I'm an adjective! I'm there!

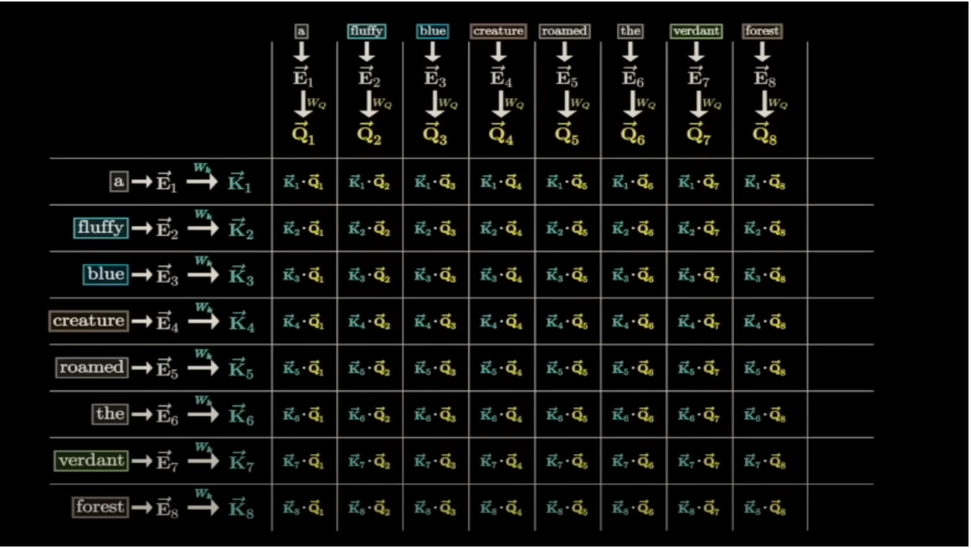
W_K is to be estimated


a fluffy blue creature roamed the verdant forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
W_Q
$\vec{Q}_1$ $\vec{Q}_2$ $\vec{Q}_3$ $\vec{Q}_4$ $\vec{Q}_5$ $\vec{Q}_6$ $\vec{Q}_7$ $\vec{Q}_8$
Any adjectives in front of me?
a → $\vec{E}_1$ $\xrightarrow{W_k}$ $\vec{K}_1$
fluffy → $\vec{E}_2$ $\xrightarrow{W_k}$ $\vec{K}_2$
I'm an adjective! I'm there!
blue → $\vec{E}_3$ $\xrightarrow{W_k}$ $\vec{K}_3$
I'm an adjective! I'm there!
creature → $\vec{E}_4$ $\xrightarrow{W_k}$ $\vec{K}_4$
roamed → $\vec{E}_5$ $\xrightarrow{W_k}$ $\vec{K}_5$

Keys and queries are represented in the same space

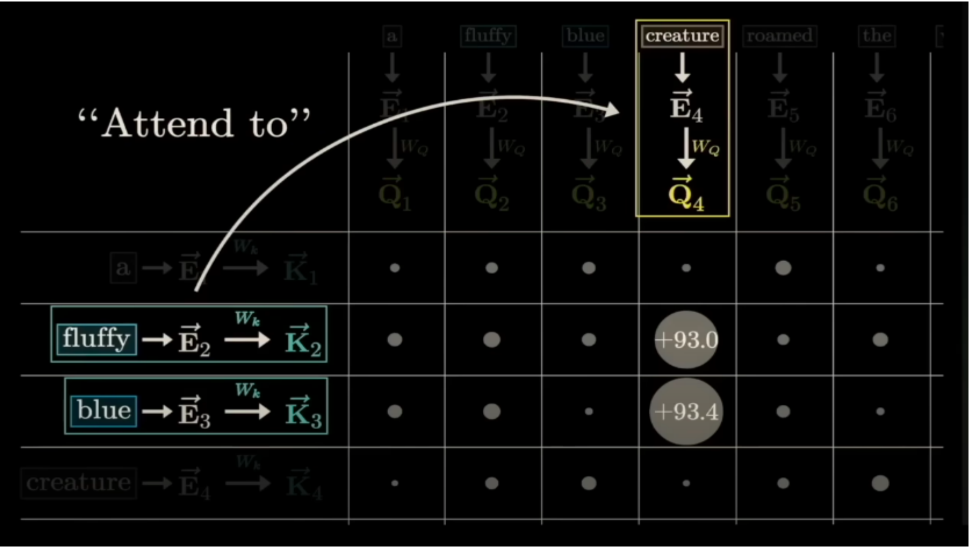

"Creature"
position 4
W_Q
Any adjectives before position 4?
a → $\vec{E}_1$ $\xrightarrow{W_k}$ $\vec{K}_1$
fluffy → $\vec{E}_2$ $\xrightarrow{W_k}$ $\vec{K}_2$
blue → $\vec{E}_3$ $\xrightarrow{W_k}$ $\vec{K}_3$
creature → $\vec{E}_4$ $\xrightarrow{W_k}$ $\vec{K}_4$
roamed → $\vec{E}_5$ $\xrightarrow{W_k}$ $\vec{K}_5$
Embedding space
12,288-dimensional
Query/Key space
128-dimensional
"Fluffy"
position 2
W_K
Adjective at position 2

The dot product between queries and keys

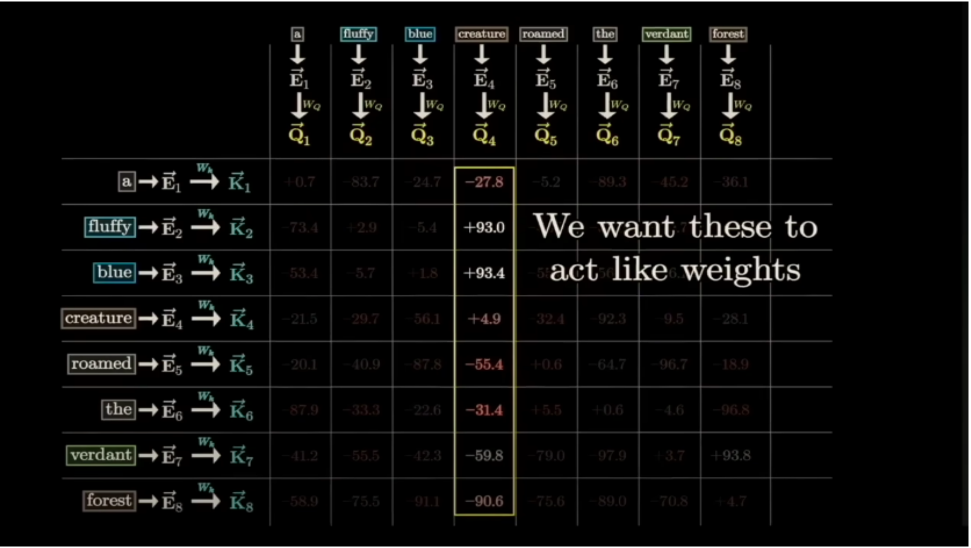


	a $\downarrow$ $\vec{E}_1$ $\downarrow W_Q$ $\vec{Q}_1$	fluffy $\downarrow$ $\vec{E}_2$ $\downarrow W_Q$ $\vec{Q}_2$	blue $\downarrow$ $\vec{E}_3$ $\downarrow W_Q$ $\vec{Q}_3$	creature $\downarrow$ $\vec{E}_4$ $\downarrow W_Q$ $\vec{Q}_4$	roamed $\downarrow$ $\vec{E}_5$ $\downarrow W_Q$ $\vec{Q}_5$	the $\downarrow$ $\vec{E}_6$ $\downarrow W_Q$ $\vec{Q}_6$	verdant $\downarrow$ $\vec{E}_7$ $\downarrow W_Q$ $\vec{Q}_7$	forest $\downarrow$ $\vec{E}_8$ $\downarrow W_Q$ $\vec{Q}_8$
a $\rightarrow \vec{E}_1 \xrightarrow{W_k} \vec{K}_1$	$\vec{K}_1 \cdot \vec{Q}_1$	$\vec{K}_1 \cdot \vec{Q}_2$	$\vec{K}_1 \cdot \vec{Q}_3$	$\vec{K}_1 \cdot \vec{Q}_4$	$\vec{K}_1 \cdot \vec{Q}_5$	$\vec{K}_1 \cdot \vec{Q}_6$	$\vec{K}_1 \cdot \vec{Q}_7$	$\vec{K}_1 \cdot \vec{Q}_8$
fluffy $\rightarrow \vec{E}_2 \xrightarrow{W_k} \vec{K}_2$	$\vec{K}_2 \cdot \vec{Q}_1$	$\vec{K}_2 \cdot \vec{Q}_2$	$\vec{K}_2 \cdot \vec{Q}_3$	$\vec{K}_2 \cdot \vec{Q}_4$	$\vec{K}_2 \cdot \vec{Q}_5$	$\vec{K}_2 \cdot \vec{Q}_6$	$\vec{K}_2 \cdot \vec{Q}_7$	$\vec{K}_2 \cdot \vec{Q}_8$
blue $\rightarrow \vec{E}_3 \xrightarrow{W_k} \vec{K}_3$	$\vec{K}_3 \cdot \vec{Q}_1$	$\vec{K}_3 \cdot \vec{Q}_2$	$\vec{K}_3 \cdot \vec{Q}_3$	$\vec{K}_3 \cdot \vec{Q}_4$	$\vec{K}_3 \cdot \vec{Q}_5$	$\vec{K}_3 \cdot \vec{Q}_6$	$\vec{K}_3 \cdot \vec{Q}_7$	$\vec{K}_3 \cdot \vec{Q}_8$
creature $\rightarrow \vec{E}_4 \xrightarrow{W_k} \vec{K}_4$	$\vec{K}_4 \cdot \vec{Q}_1$	$\vec{K}_4 \cdot \vec{Q}_2$	$\vec{K}_4 \cdot \vec{Q}_3$	$\vec{K}_4 \cdot \vec{Q}_4$	$\vec{K}_4 \cdot \vec{Q}_5$	$\vec{K}_4 \cdot \vec{Q}_6$	$\vec{K}_4 \cdot \vec{Q}_7$	$\vec{K}_4 \cdot \vec{Q}_8$
roamed $\rightarrow \vec{E}_5 \xrightarrow{W_k} \vec{K}_5$	$\vec{K}_5 \cdot \vec{Q}_1$	$\vec{K}_5 \cdot \vec{Q}_2$	$\vec{K}_5 \cdot \vec{Q}_3$	$\vec{K}_5 \cdot \vec{Q}_4$	$\vec{K}_5 \cdot \vec{Q}_5$	$\vec{K}_5 \cdot \vec{Q}_6$	$\vec{K}_5 \cdot \vec{Q}_7$	$\vec{K}_5 \cdot \vec{Q}_8$
the $\rightarrow \vec{E}_6 \xrightarrow{W_k} \vec{K}_6$	$\vec{K}_6 \cdot \vec{Q}_1$	$\vec{K}_6 \cdot \vec{Q}_2$	$\vec{K}_6 \cdot \vec{Q}_3$	$\vec{K}_6 \cdot \vec{Q}_4$	$\vec{K}_6 \cdot \vec{Q}_5$	$\vec{K}_6 \cdot \vec{Q}_6$	$\vec{K}_6 \cdot \vec{Q}_7$	$\vec{K}_6 \cdot \vec{Q}_8$
verdant $\rightarrow \vec{E}_7 \xrightarrow{W_k} \vec{K}_7$	$\vec{K}_7 \cdot \vec{Q}_1$	$\vec{K}_7 \cdot \vec{Q}_2$	$\vec{K}_7 \cdot \vec{Q}_3$	$\vec{K}_7 \cdot \vec{Q}_4$	$\vec{K}_7 \cdot \vec{Q}_5$	$\vec{K}_7 \cdot \vec{Q}_6$	$\vec{K}_7 \cdot \vec{Q}_7$	$\vec{K}_7 \cdot \vec{Q}_8$
forest $\rightarrow \vec{E}_8 \xrightarrow{W_k} \vec{K}_8$	$\vec{K}_8 \cdot \vec{Q}_1$	$\vec{K}_8 \cdot \vec{Q}_2$	$\vec{K}_8 \cdot \vec{Q}_3$	$\vec{K}_8 \cdot \vec{Q}_4$	$\vec{K}_8 \cdot \vec{Q}_5$	$\vec{K}_8 \cdot \vec{Q}_6$	$\vec{K}_8 \cdot \vec{Q}_7$	$\vec{K}_8 \cdot \vec{Q}_8$

Attend to ...


"Attend to"
a
fluffy
blue
creature
roamed
the
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$
W_Q
$\vec{Q}_1$ $\vec{Q}_2$ $\vec{Q}_3$ $\vec{Q}_4$ $\vec{Q}_5$ $\vec{Q}_6$
$a \rightarrow \vec{E}_1 \xrightarrow{W_k} \vec{K}_1$
$fluffy \rightarrow \vec{E}_2 \xrightarrow{W_k} \vec{K}_2$
$blue \rightarrow \vec{E}_3 \xrightarrow{W_k} \vec{K}_3$
$creature \rightarrow \vec{E}_4 \xrightarrow{W_k} \vec{K}_4$
+93.0
+93.4

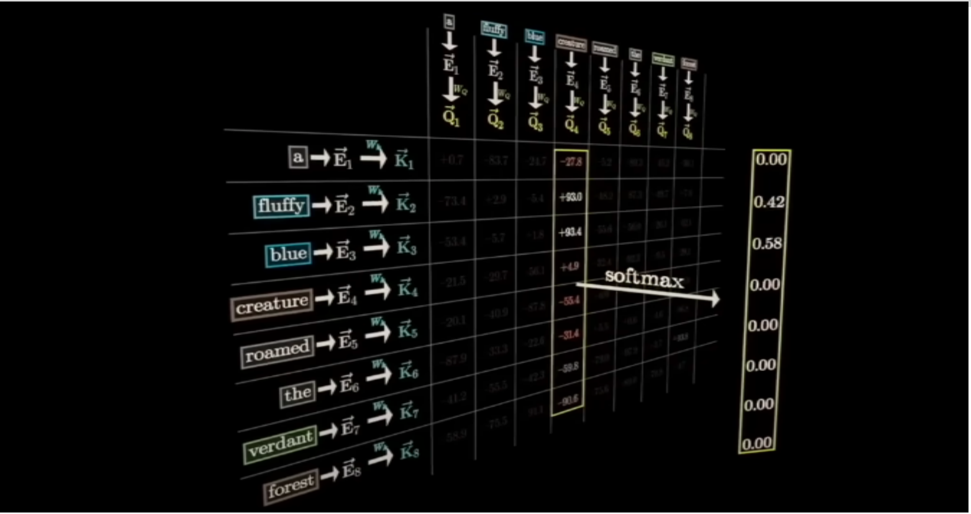
Dot product to weights



	a $\downarrow$ $\vec{E}_1$ $\downarrow W_Q$ $\vec{Q}_1$	fluffy $\downarrow$ $\vec{E}_2$ $\downarrow W_Q$ $\vec{Q}_2$	blue $\downarrow$ $\vec{E}_3$ $\downarrow W_Q$ $\vec{Q}_3$	creature $\downarrow$ $\vec{E}_4$ $\downarrow W_Q$ $\vec{Q}_4$	roamed $\downarrow$ $\vec{E}_5$ $\downarrow W_Q$ $\vec{Q}_5$	the $\downarrow$ $\vec{E}_6$ $\downarrow W_Q$ $\vec{Q}_6$	verdant $\downarrow$ $\vec{E}_7$ $\downarrow W_Q$ $\vec{Q}_7$	forest $\downarrow$ $\vec{E}_8$ $\downarrow W_Q$ $\vec{Q}_8$
a → $\vec{E}_1 \xrightarrow{W_k} \vec{K}_1$	+0.7	−83.7	−24.7	−27.8	−5.2	−89.3	−45.2	−36.1
fluffy → $\vec{E}_2 \xrightarrow{W_k} \vec{K}_2$	−73.4	+2.9	−5.4	+93.0	[illegible]	[illegible]	[illegible]	[illegible]
blue → $\vec{E}_3 \xrightarrow{W_k} \vec{K}_3$	−53.4	−5.7	+1.8	+93.4	[illegible]	[illegible]	[illegible]	[illegible]
creature → $\vec{E}_4 \xrightarrow{W_k} \vec{K}_4$	−21.5	−29.7	−56.1	+4.9	−32.4	−92.3	−9.5	−28.1
roamed → $\vec{E}_5 \xrightarrow{W_k} \vec{K}_5$	−20.1	−40.9	−87.8	−55.4	+0.6	−64.7	−96.7	−18.9
the → $\vec{E}_6 \xrightarrow{W_k} \vec{K}_6$	−87.9	−33.3	−22.6	−31.4	+5.5	+0.6	−4.6	−96.8
verdant → $\vec{E}_7 \xrightarrow{W_k} \vec{K}_7$	−41.2	−55.5	−42.3	−59.8	−79.0	−97.9	+3.7	+93.8
forest → $\vec{E}_8 \xrightarrow{W_k} \vec{K}_8$	−58.9	−75.5	−91.1	−90.6	−75.6	−89.0	−70.8	+4.7

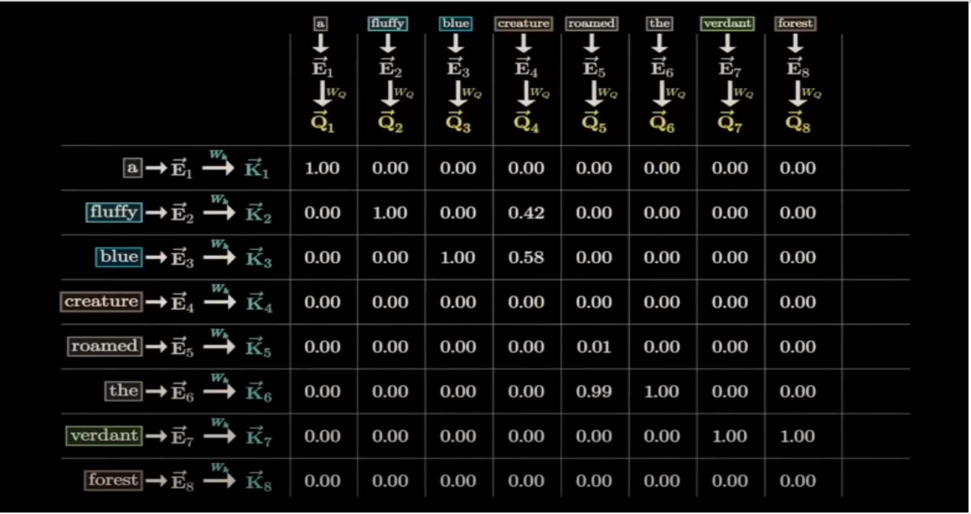
We want these to act like weights

Softmax



For numerical stability, we need to devide dot products on $\sqrt{d}$ before softmax.

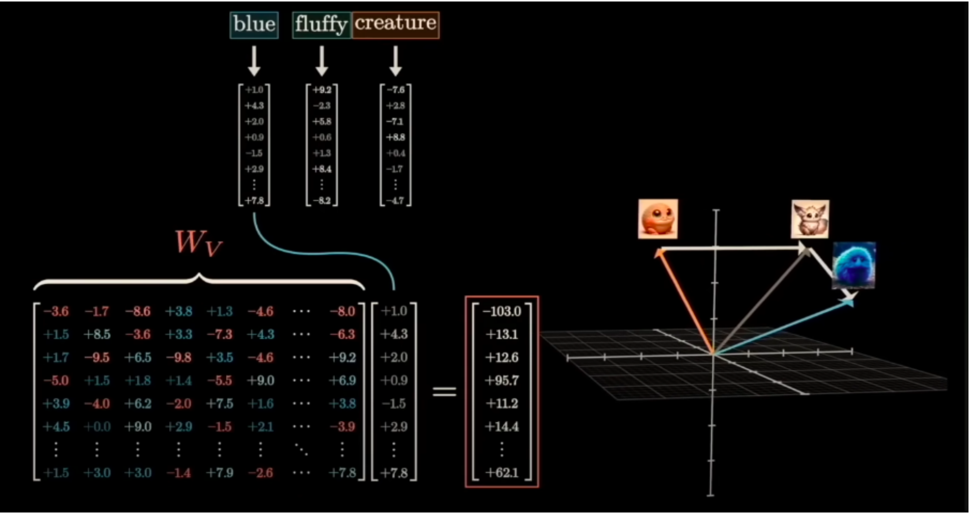
Attention pattern



	a $\rightarrow \vec{E}_1 \xrightarrow{W_Q} \vec{Q}_1$	fluffy $\rightarrow \vec{E}_2 \xrightarrow{W_Q} \vec{Q}_2$	blue $\rightarrow \vec{E}_3 \xrightarrow{W_Q} \vec{Q}_3$	creature $\rightarrow \vec{E}_4 \xrightarrow{W_Q} \vec{Q}_4$	roamed $\rightarrow \vec{E}_5 \xrightarrow{W_Q} \vec{Q}_5$	the $\rightarrow \vec{E}_6 \xrightarrow{W_Q} \vec{Q}_6$	verdant $\rightarrow \vec{E}_7 \xrightarrow{W_Q} \vec{Q}_7$	forest $\rightarrow \vec{E}_8 \xrightarrow{W_Q} \vec{Q}_8$
a $\rightarrow \vec{E}_1 \xrightarrow{W_k} \vec{K}_1$	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fluffy $\rightarrow \vec{E}_2 \xrightarrow{W_k} \vec{K}_2$	0.00	1.00	0.00	0.42	0.00	0.00	0.00	0.00
blue $\rightarrow \vec{E}_3 \xrightarrow{W_k} \vec{K}_3$	0.00	0.00	1.00	0.58	0.00	0.00	0.00	0.00
creature $\rightarrow \vec{E}_4 \xrightarrow{W_k} \vec{K}_4$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
roamed $\rightarrow \vec{E}_5 \xrightarrow{W_k} \vec{K}_5$	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
the $\rightarrow \vec{E}_6 \xrightarrow{W_k} \vec{K}_6$	0.00	0.00	0.00	0.00	0.99	1.00	0.00	0.00
verdant $\rightarrow \vec{E}_7 \xrightarrow{W_k} \vec{K}_7$	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
forest $\rightarrow \vec{E}_8 \xrightarrow{W_k} \vec{K}_8$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The attention pattern encodes the dependency structure between tokens

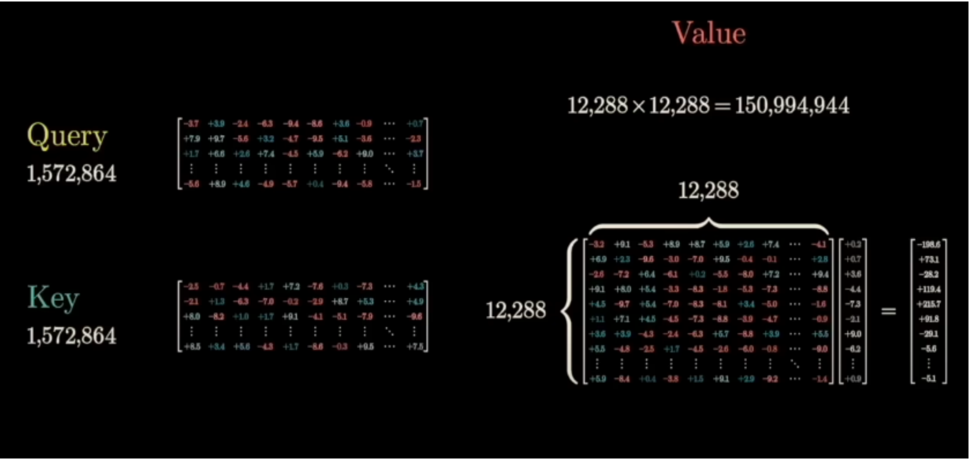
Value matrix W_V



$$\begin{bmatrix} +1.0 \\ +4.3 \\ +2.0 \\ +0.9 \\ -1.5 \\ +2.9 \\ \vdots \\ +7.8 \end{bmatrix} \quad \begin{bmatrix} +9.2 \\ -2.3 \\ +5.8 \\ +0.6 \\ +1.3 \\ +8.4 \\ \vdots \\ -8.2 \end{bmatrix} \quad \begin{bmatrix} -7.6 \\ +2.8 \\ -7.1 \\ +8.8 \\ +0.4 \\ -1.7 \\ \vdots \\ -4.7 \end{bmatrix}$$

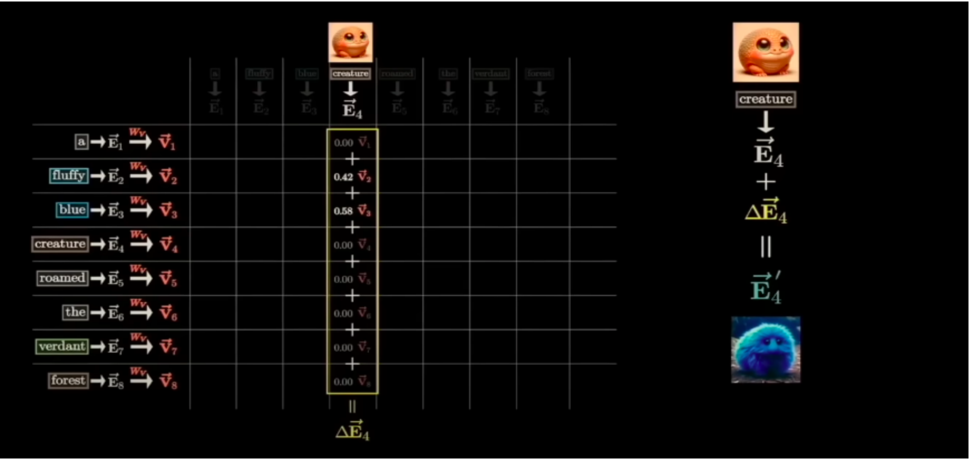
$$\overbrace{\begin{bmatrix} -3.6 & -1.7 & -8.6 & +3.8 & +1.3 & -4.6 & \cdots & -8.0 \\ +1.5 & +8.5 & -3.6 & +3.3 & -7.3 & +4.3 & \cdots & -6.3 \\ +1.7 & -9.5 & +6.5 & -9.8 & +3.5 & -4.6 & \cdots & +9.2 \\ -5.0 & +1.5 & +1.8 & +1.4 & -5.5 & +9.0 & \cdots & +6.9 \\ +3.9 & -4.0 & +6.2 & -2.0 & +7.5 & +1.6 & \cdots & +3.8 \\ +4.5 & +0.0 & +9.0 & +2.9 & -1.5 & +2.1 & \cdots & -3.9 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ +1.5 & +3.0 & +3.0 & -1.4 & +7.9 & -2.6 & \cdots & +7.8 \end{bmatrix}}^{W_V} \begin{bmatrix} +1.0 \\ +4.3 \\ +2.0 \\ +0.9 \\ -1.5 \\ +2.9 \\ \vdots \\ +7.8 \end{bmatrix} = \boxed{\begin{bmatrix} -103.0 \\ +13.1 \\ +12.6 \\ +95.7 \\ +11.2 \\ +14.4 \\ \vdots \\ +62.1 \end{bmatrix}}$$

W_V from $\mathbb{R}^d \to \mathbb{R}^d$

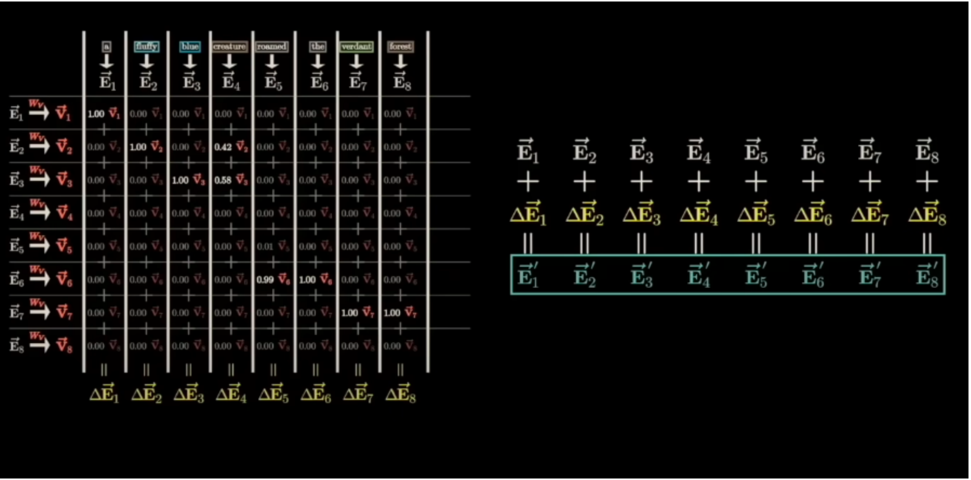


In practice, it is the product of two low-rank matrices

How is the attention pattern used in practice?


a
fluffy
blue
creature
roamed
the
verdant
forest
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$ $\vec{E}_8$
a $\rightarrow \vec{E}_1 \xrightarrow{W_V} \vec{V}_1$
fluffy $\rightarrow \vec{E}_2 \xrightarrow{W_V} \vec{V}_2$
blue $\rightarrow \vec{E}_3 \xrightarrow{W_V} \vec{V}_3$
creature $\rightarrow \vec{E}_4 \xrightarrow{W_V} \vec{V}_4$
roamed $\rightarrow \vec{E}_5 \xrightarrow{W_V} \vec{V}_5$
the $\rightarrow \vec{E}_6 \xrightarrow{W_V} \vec{V}_6$
verdant $\rightarrow \vec{E}_7 \xrightarrow{W_V} \vec{V}_7$
forest $\rightarrow \vec{E}_8 \xrightarrow{W_V} \vec{V}_8$
$0.00\,\vec{V}_1 + 0.42\,\vec{V}_2 + 0.58\,\vec{V}_3 + 0.00\,\vec{V}_4 + 0.00\,\vec{V}_5 + 0.00\,\vec{V}_6 + 0.00\,\vec{V}_7 + 0.00\,\vec{V}_8$
$= \Delta\vec{E}_4$
creature
$\vec{E}_4 + \Delta\vec{E}_4 = \vec{E}_4'$

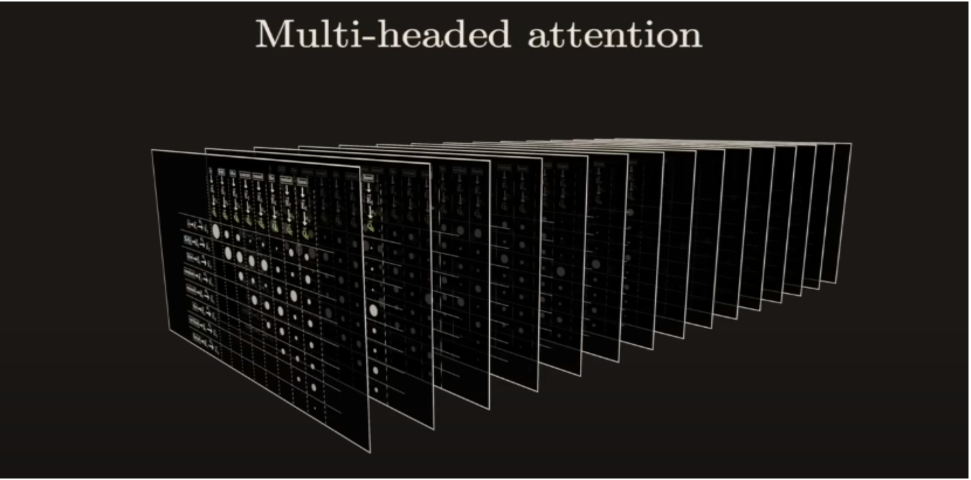
Single-head attention update



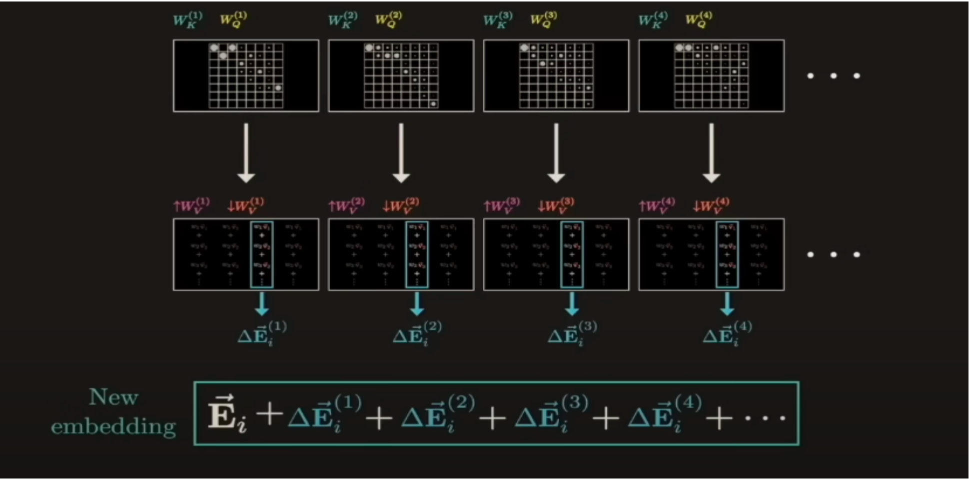
	a	fluffy	blue	creature	roamed	the	verdant	forest
	$\vec{E}_1$	$\vec{E}_2$	$\vec{E}_3$	$\vec{E}_4$	$\vec{E}_5$	$\vec{E}_6$	$\vec{E}_7$	$\vec{E}_8$
$\vec{E}_1 \xrightarrow{W_V} \vec{V}_1$	$1.00\ \vec{V}_1$	$0.00\ \vec{V}_1$	$0.00\ \vec{V}_1$	$0.00\ \vec{V}_1$	$0.00\ \vec{V}_1$	$0.00\ \vec{V}_1$	$0.00\ \vec{V}_1$	$0.00\ \vec{V}_1$
+	+	+	+	+	+	+	+	+
$\vec{E}_2 \xrightarrow{W_V} \vec{V}_2$	$0.00\ \vec{V}_2$	$1.00\ \vec{V}_2$	$0.00\ \vec{V}_2$	$0.42\ \vec{V}_2$	$0.00\ \vec{V}_2$	$0.00\ \vec{V}_2$	$0.00\ \vec{V}_2$	$0.00\ \vec{V}_2$
	+	+	+	+	+	+	+	+
$\vec{E}_3 \xrightarrow{W_V} \vec{V}_3$	$0.00\ \vec{V}_3$	$0.00\ \vec{V}_3$	$1.00\ \vec{V}_3$	$0.58\ \vec{V}_3$	$0.00\ \vec{V}_3$	$0.00\ \vec{V}_3$	$0.00\ \vec{V}_3$	$0.00\ \vec{V}_3$
	+	+	+	+	+	+	+	+
$\vec{E}_4 \xrightarrow{W_V} \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$	$0.00\ \vec{V}_4$
	+	+	+	+	+	+	+	+
$\vec{E}_5 \xrightarrow{W_V} \vec{V}_5$	$0.00\ \vec{V}_5$	$0.00\ \vec{V}_5$	$0.00\ \vec{V}_5$	$0.00\ \vec{V}_5$	$0.01\ \vec{V}_5$	$0.00\ \vec{V}_5$	$0.00\ \vec{V}_5$	$0.00\ \vec{V}_5$
	+	+	+	+	+	+	+	+
$\vec{E}_6 \xrightarrow{W_V} \vec{V}_6$	$0.00\ \vec{V}_6$	$0.00\ \vec{V}_6$	$0.00\ \vec{V}_6$	$0.00\ \vec{V}_6$	$0.99\ \vec{V}_6$	$1.00\ \vec{V}_6$	$0.00\ \vec{V}_6$	$0.00\ \vec{V}_6$
	+	+	+	+	+	+	+	+
$\vec{E}_7 \xrightarrow{W_V} \vec{V}_7$	$0.00\ \vec{V}_7$	$0.00\ \vec{V}_7$	$0.00\ \vec{V}_7$	$0.00\ \vec{V}_7$	$0.00\ \vec{V}_7$	$0.00\ \vec{V}_7$	$1.00\ \vec{V}_7$	$1.00\ \vec{V}_7$
	+	+	+	+	+	+	+	+
$\vec{E}_8 \xrightarrow{W_V} \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$	$0.00\ \vec{V}_8$
	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
	$\Delta\vec{E}_1$	$\Delta\vec{E}_2$	$\Delta\vec{E}_3$	$\Delta\vec{E}_4$	$\Delta\vec{E}_5$	$\Delta\vec{E}_6$	$\Delta\vec{E}_7$	$\Delta\vec{E}_8$

$$
\begin{array}{cccccccc}
\vec{E}_1 & \vec{E}_2 & \vec{E}_3 & \vec{E}_4 & \vec{E}_5 & \vec{E}_6 & \vec{E}_7 & \vec{E}_8 \\
+ & + & + & + & + & + & + & + \\
\Delta\vec{E}_1 & \Delta\vec{E}_2 & \Delta\vec{E}_3 & \Delta\vec{E}_4 & \Delta\vec{E}_5 & \Delta\vec{E}_6 & \Delta\vec{E}_7 & \Delta\vec{E}_8 \\
\parallel & \parallel & \parallel & \parallel & \parallel & \parallel & \parallel & \parallel \\
\vec{E}'_1 & \vec{E}'_2 & \vec{E}'_3 & \vec{E}'_4 & \vec{E}'_5 & \vec{E}'_6 & \vec{E}'_7 & \vec{E}'_8
\end{array}
$$

LM need Multi-headed attention



Original embedding update


$W_K^{(1)}$ $W_Q^{(1)}$
$W_K^{(2)}$ $W_Q^{(2)}$
$W_K^{(3)}$ $W_Q^{(3)}$
$W_K^{(4)}$ $W_Q^{(4)}$
$\uparrow W_V^{(1)}$ $\downarrow W_V^{(1)}$
$\uparrow W_V^{(2)}$ $\downarrow W_V^{(2)}$
$\uparrow W_V^{(3)}$ $\downarrow W_V^{(3)}$
$\uparrow W_V^{(4)}$ $\downarrow W_V^{(4)}$
$\Delta\vec{\mathbf{E}}_i^{(1)}$
$\Delta\vec{\mathbf{E}}_i^{(2)}$
$\Delta\vec{\mathbf{E}}_i^{(3)}$
$\Delta\vec{\mathbf{E}}_i^{(4)}$

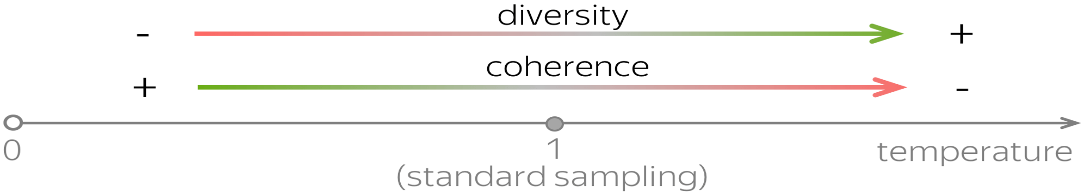
New embedding

$$\boxed{\vec{\mathbf{E}}_i + \Delta\vec{\mathbf{E}}_i^{(1)} + \Delta\vec{\mathbf{E}}_i^{(2)} + \Delta\vec{\mathbf{E}}_i^{(3)} + \Delta\vec{\mathbf{E}}_i^{(4)} + \cdots}$$

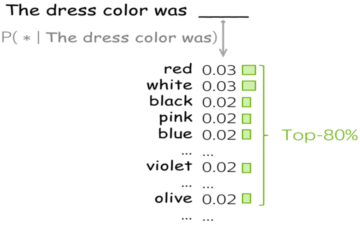
Sampling parameters

Let $E \in \mathbb{R}^d$ be the final embedding of the word on the right of the sequence, obtained as the output of the last transformer layer, and let $D \in \mathbb{R}^{|V| \times d}$ the de-embedding matrix used to map the final embeddings back to the output space for classification. The temperature parameter T is used to sample according to the distribution probability given by

$$\text{Softmax}\left(\frac{DE}{T}\right)$$



Top-p sampling: top-p% of the probability mass

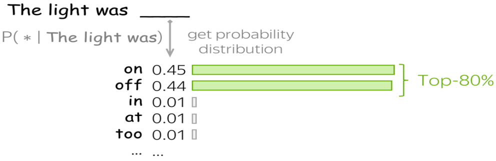


The dress color was ____

P(∗ | The dress color was)

red	0.03
white	0.03
black	0.02
pink	0.02
blue	0.02
...	...
violet	0.02
...	...
olive	0.02
...	...

Top-80%



The light was ____

P(∗ | The light was) get probability distribution

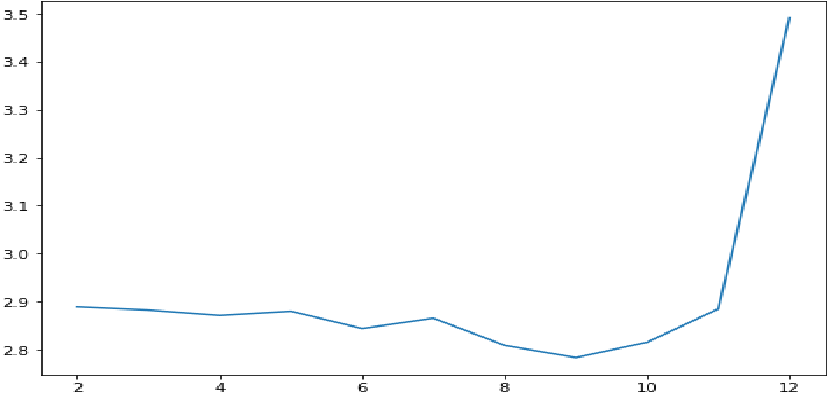
on	0.45
off	0.44
in	0.01
at	0.01
too	0.01
...	...

Top-80%

Layer-Augmented Model for Classification

- BERT embeddings of dimension 768
- 12 Transformer blocks with 12 attention heads (110 million parameters)
- Context size of 512 tokens
- Vocabulary size of 30,522 tokens
- Two fully connected layers reduce the output dimension from 768 to 200 and then to 110, before the final sigmoid classification layer

Layer index from which the model is fine-tuned



Repeated 5-fold cross-validation with 10 repetitions, 10 days of processing on a 40 GB Nvidia A100 GPU

The models being compared

Familles classiques	**hyper-paramètres**
Elasticnet logistic regression	C and l_1 ratio
Gradient boosting classifier	Learning rate and number of estimators
Random Forest	Maximum of features and bootstrap
Support vector classifier	C, gamma and kernel

Cross-validation was used for each model family to tune the hyperparameters. The procedure was carried out twice: once using BERT embeddings as features, and a second time using the TF-IDF representation of the text.

Results

Precision = TP/TP+FP, Recall = TP/TP+FN et F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Label	+	-	0	i	j	f	s	p	m	a	t
Support	72	63	47	146	263	36	57	124	123	72	23
Precision											
Fine-tuning	0.67	**0.85**	0.40	**0.78**	**0.86**	**0.91**	0.97	**0.97**	**0.94**	**0.91**	0.78
Elasticnet lr	0.65(0.69)	0.71(0.75)	0.28(0.33)	0.74(0.61)	0.85(0.81)	0.74(0.79)	0.89(**0.98**)	0.90(0.93)	0.83(0.87)	0.77(0.81)	0.67(0.75)
Gradient Boosting	0.73(0.65)	0.88(0.44)	0.25(**0.45**)	0.75(0.64)	0.84(0.80)	0.75(0.78)	0.88(0.92)	0.84(0.93)	0.79(0.85)	0.83(0.73)	0(**0.83**)
Random Forest	**0.82**(0.70)	0.71(0.73)	0(0.33)	0.71(0.64)	0.82(0.80)	0(0.72)	0.89(0.92)	0.86(0.93)	0.76(0.89)	0.73(0.77)	0(0.67)
SVC	0.68(0.70)	0.79(1)	0(0)	0.76(0.66)	0.85(0.81)	0.68(0.79)	0.91(0.94)	0.90(0.92)	0.83(0.89)	0.82(0.79)	0.50(0.79)
Recall											
Fine-tuning	**0.82**	0.37	**0.38**	**0.81**	0.97	**0.89**	**0.98**	**0.93**	**0.93**	**0.86**	**0.61**
Elasticnet lr	0.56(0.49)	**0.40**(0.10)	0.11(0.02)	0.73(0.75)	0.94(1)	0.39(0.64)	0.68(0.81)	0.76(0.85)	0.82(0.85)	0.67(0.64)	0.26(0.52)
Gradient Boosting	0.49(0.49)	0.22(0.11)	0.02(0.11)	0.75(0.66)	0.97(0.95)	0.17(0.81)	0.49(0.86)	0.70(0.85)	0.69(0.90)	0.47(0.64)	0.00(0.43)
Random Forest	0.38(0.43)	0.08(0.17)	0(0.04)	0.75(0.59)	0.96(0.92)	0.00(0.86)	0.30(0.95)	0.52(0.92)	0.60(0.93)	0.33(0.71)	0.00(0.52)
SVC	0.57(0.46)	0.30(0.05)	0(0)	0.77(0.73)	0.94(**1**)	0.42(0.64)	0.72(0.86)	0.77(0.78)	0.81(0.89)	0.64(0.58)	0.09(0.48)
f1-score											
Fine-tuning	**0.74**	**0.51**	**0.39**	**0.79**	**0.91**	**0.90**	**0.97**	**0.95**	**0.93**	**0.89**	**0.68**
Elasticnet lr	0.60(0.57)	0.51(0.17)	0.15(0.04)	0.74(0.67)	0.89(0.89)	0.51(0.71)	0.77(0.88)	0.82(0.89)	0.83(0.86)	0.72(0.71)	0.38(0.62)
Gradient Boosting	0.58(0.56)	0.35(0.18)	0.04(0.17)	0.75(0.65)	0.90(0.87)	0.27(0.79)	0.63(0.89)	0.77(0.89)	0.74(0.88)	0.60(0.68)	0(0.57)
Random Forest	0.51(0.53)	0.14(0.28)	0(0.08)	0.73(0.61)	0.89(0.86)	0.00(0.78)	0.45(0.93)	0.65(0.92)	0.67(0.91)	0.46(0.74)	0(0.59)
SVC	0.62(0.55)	0.44(0.09)	0(0)	0.77(0.69)	0.90(0.89)	0.52(0.71)	0.80(0.90)	0.83(0.84)	0.82(0.89)	0.72(0.67)	0.15(0.59)

The main takeaway

- Use BERT embeddings instead of TF-IDF
- Try fine-tuning despite the small sample size

Pre-training : Data

- Idea : use all of the clean internet
- Note : internet is dirty and not representative of what we want. Pratice :
    1. Download all the internet. Common crawl : 250 billion pages, > 1 PB (> 1e6 GB)
    2. Text extraction from HTML (challenge: math)
    3. Filter undesirable content (e.g. NSFW, harmful content)
    4. Remove (all the headers/footers/menu in forums are always same
    5. Heuristic filtering. Remove low quality documents (e.g. number of words, word length, outlier toks)
    6. Model based filtering. Predict if page could be references by Wikipedia
    7. Data mix. Classify data categories (code/books/entertainment). Reweight domains using scaling low to get high downstream peformance
- Learning Rate annealing on high-quality data

Common academic datasets

- A lot of secrecy : competitive dynamics and **copyright liability**
- Common academic datasets : C4(150B tokens | 800GB), Dolma (3T tokens), The Pile (280B tokens) and FineWeb (15T tokens).

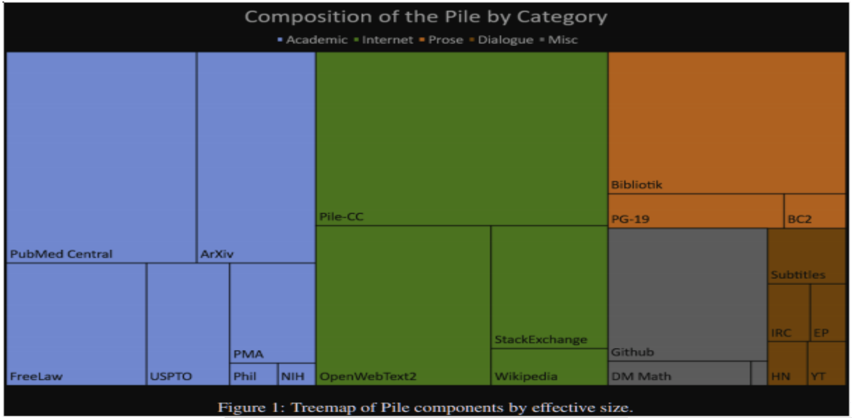
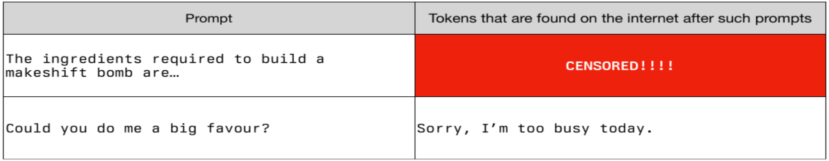


Figure 1: Treemap of Pile components by effective size.

Post-training

- Language Modeling ≠ assisting users, prompting a next-token predictor can be challenging and unsafe. Remember what it was trained for . . .

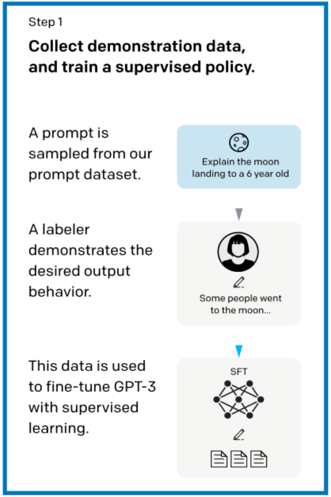


Prompt	Tokens that are found on the internet after such prompts
`The ingredients required to build a makeshift bomb are…`	CENSORED!!!!
`Could you do me a big favour?`	`Sorry, I'm too busy today.`

- Want to align models to our goals. *Not what I said, what I meant!*
- These methods are broadly called post-training and include **supervised fine-tuning (SFT)** and **reinforcement learning (RL)**.
- The *formula* for pre-training is highly conserved in the industry, but **post-training strategies are very diverse**.

Supervised fine-tuning (SFT)

1. Next-token prediction on a curated dataset of exemplary interactions
2. Given an exemplary (prompt,completion) pair, use cross-entropy loss on the completion conditioned on the prompt
3. Only accumulate loss for the completion



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

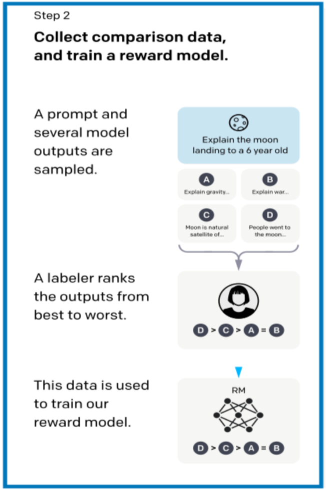
A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT

Human feedback

1. What can we do if we don’t have exemplary data?
2. We can often get human preferences more cheaply, in particular, pairwise preferences.
3. Key idea: generate multiple completions from the model and query humans for pairwise preferences to learn a reward function.



Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

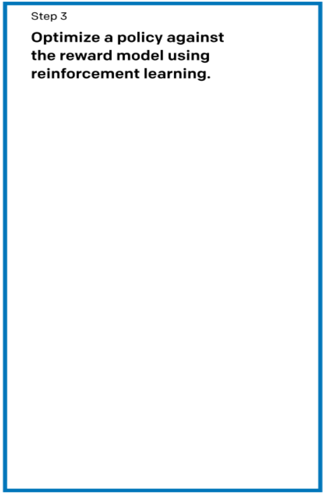
A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B

Human feedback

1. Once we have a reward function, we can apply the techniques of reinforcement learning.
2. We think of the model's completion distribution as a *policy*.
3. We update the model towards completion distributions that get higher reward (under the learned reward).



Step 3

Optimize a policy against the reward model using reinforcement learning.

RL from Human Feedback (RLHF)

- Problem : SFT is **behavior cloning** of humans
  1. **Bound by human abilities:** humans may prefer things that they are not able to generate
  2. **Hallucination:** cloning *correct* answer teaches LLM to hallucinate if it didn't know about it. If LLM doesn't know → teaches the model to make up plausibly sounding references
  3. **Price :** collecting ideal answers is expensive

RLHF :Proximal Policy Optimization

- Idea : use reinforcement learning
- What is the reward ?
  - Binary reward doesn't have much information
  - Train a *a reward model R* to classify preferences

Structured data extraction from textual sources

- Approximately 500,000 patient records collected at Gustave Roussy
- Each patient record contains between 50 and 500 reports
- A broad range of report types. To begin with, we focused on

CA C.R. Anatomophatologie

CC C.R. Consultation

RC C.R Réunion de Concertation Pluridisciplinaire

LLMs tested for this task

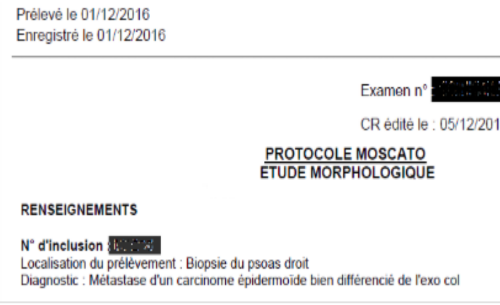
- The main constraint is the use of nominative data, which prohibits external computation; therefore, the LLM must be hosted within Gustave Roussy's own infrastructure
  1. Deepseek-distill-llama 70B
  2. Llama-med42-70B
  3. Mistral-123B
  4. Deepseek-R1
- The best extraction results (without hallucination) are achieved by Deepseek-R1

Structured data extraction using Deepseek-R1

- One prompt per report type.
- We manage the output dictionary keys separately (avoid asking the LLM to fill in a dictionary directly in the prompt).
- All reports are processed within a context size of 3,072 tokens (around 40 seconds per report), which covers 99% of the reports.
- A second step is applied with a context size multiplied by 4 for the reports that failed in the previous step.

Example with report CA (anatomopathology)

```
{
  "date_creation": "01/12/2016",
  "date_validation": "05/12/2016",
  "date_edition": "05/12/2016",
  "site_primitif": "exo col",
  "site_biopsie": "psoas droit",
  "type_histologique": "carcinome épidermoïde bien différencié",
  "grade_histologique": "bien différencié",
   "diagnostic": "Métastase d'un carcinome épidermoïde bien différencié de l'exo col",
   "CR": "201602493HD_CA_128838549_20161205_201602493HD_16H11578_2255398_20161205170208128838549.P
   "document_type": "CA"
 }
```



Prélevé le 01/12/2016
Enregistré le 01/12/2016

Examen n° [redacted]

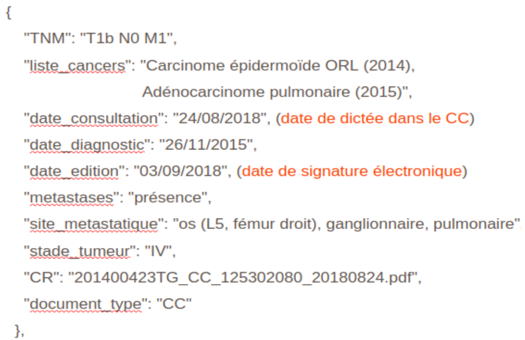
CR édité le : 05/12/201

PROTOCOLE MOSCATO
ETUDE MORPHOLOGIQUE

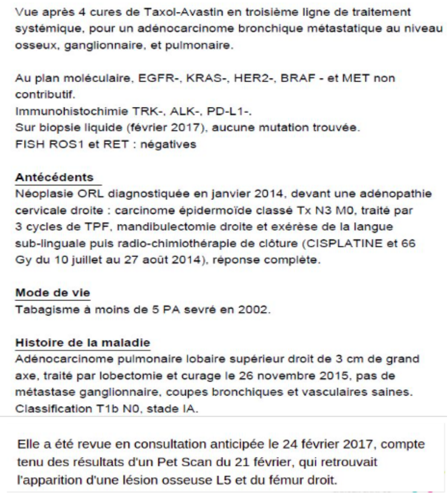
RENSEIGNEMENTS

N° d'inclusion [redacted]
Localisation du prélèvement : Biopsie du psoas droit
Diagnostic : Métastase d'un carcinome épidermoïde bien différencié de l'exo col

Example with report CC(consultation)



```
{
  "TNM": "T1b N0 M1",
  "liste_cancers": "Carcinome épidermoïde ORL (2014),
                   Adénocarcinome pulmonaire (2015)",
  "date_consultation": "24/08/2018", (date de dictée dans le CC)
  "date_diagnostic": "26/11/2015",
  "date_edition": "03/09/2018", (date de signature électronique)
  "metastases": "présence",
  "site_metastatique": "os (L5, fémur droit), ganglionnaire, pulmonaire",
  "stade_tumeur": "IV",
  "CR": "201400423TG_CC_125302080_20180824.pdf",
  "document_type": "CC"
 },
```



Vue après 4 cures de Taxol-Avastin en troisième ligne de traitement systémique, pour un adénocarcinome bronchique métastatique au niveau osseux, ganglionnaire, et pulmonaire.

Au plan moléculaire, EGFR-, KRAS-, HER2-, BRAF - et MET non contributif.
Immunohistochimie TRK-, ALK-, PD-L1-.
Sur biopsie liquide (février 2017), aucune mutation trouvée.
FISH ROS1 et RET : négatives

Antécédents

Néoplasie ORL diagnostiquée en janvier 2014, devant une adénopathie cervicale droite : carcinome épidermoïde classé Tx N3 M0, traité par 3 cycles de TPF, mandibulectomie droite et exérèse de la langue sub-linguale puis radio-chimiothérapie de clôture (CISPLATINE et 66 Gy du 10 juillet au 27 août 2014), réponse complète.

Mode de vie

Tabagisme à moins de 5 PA sevré en 2002.

Histoire de la maladie

Adénocarcinome pulmonaire lobaire supérieur droit de 3 cm de grand axe, traité par lobectomie et curage le 26 novembre 2015, pas de métastase ganglionnaire, coupes bronchiques et vasculaires saines. Classification T1b N0, stade IA.

Elle a été revue en consultation anticipée le 24 février 2017, compte tenu des résultats d'un Pet Scan du 21 février, qui retrouvait l'apparition d'une lésion osseuse L5 et du fémur droit.

The issues to be addressed

- A patient is a list of key-value dictionaries, where each dictionary represents the information extracted from one report
- The best strategy to convert a list of dictionaries into tabular data in the form of a long vector
- Handle inconsistencies between the numerous dates reported in the consultation reports
- The computation time is slow despite a very favorable infrastructure (8 x Nvidia H100 80GB)

Merci