Deep Learning

GPT and Attention

What is Natural Language Processing (NLP)?

- Subfield of AI that focuses on reading, deciphering and producing human language
- Combines computational linguistics (e.g. rule-based modelling of language) with statistical, ML, and deep learning approaches
- Through NLP, machines can understand, analyze and generate language in ways that are meaningful and contextually appropriate
- While LLMs have driven an explosion in interest, there are many older technologies which paved the way

Common Applications

- **Text Classification:** e.g. spam detection, news classification
- **Question Answering:** processing input and finding relevant information
- **Machine Translation**
- **Sentiment Analysis**
- **Code Generation**

Challenges in NLP

- **Ambiguity:** Words can mean different things depending on context
- **Nuances:** Languages are full of idioms, slang, cultural references, sarcasm…
- **Syntax vs Semantics:** A grammatically correct sentence might not make sense, or a grammatically incorrect one might be easy to understand

- I saw a man on the hill with the telescope
- That’s a cool cat
- Colourless green ideas sleep furiously
- Me went store

Challenges in NLP

Quick history of NLP

- **1950s:**
  - Alan Turing publishes “Computing Machinery and Intelligence”, in which he proposes the Turing test
  - Noam Chomsky publishes “Syntactic Structures”, an attempt to construct a formal theory of linguistic structure
- **1960s:**
  - Georgetown University develops a machine translation system, which automatically translates 60 Russian sentences into English using an extremely complex flowchart and a limited vocabulary
  - The authors claim machine translation could be a solved problem in five years

Quick history of NLP

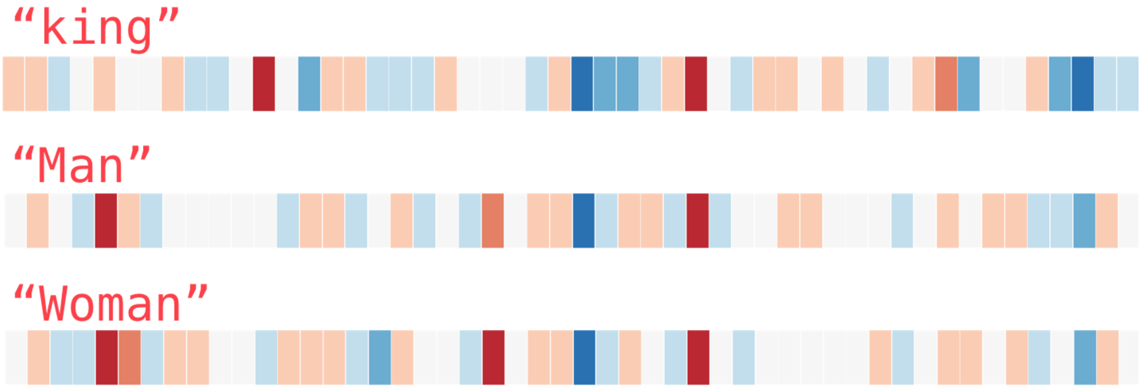
- **1980s – 1990s:**
  - Transition from rule-based to statistical approaches
  - Idea of using existing text to train a model begins to appear (e.g. bilingual documents from the Canadian parliament)
- **2000s – 2010s:**
  - Deep Learning takes over
  - Tools such as convolutional networks, and later RNNs, transform the field
- **2020s: The era of the Large Language Model**

Foundations of LLMs

- Fundamental goal of language modelling: next word prediction
- $P(cat \mid the\ dog\ and\ the)$
- To generate, pick the word with highest likelihood
- Early models could handle one, two words of context
- Locally coherent, but longer texts quickly lose meaning
- More context requires more complexity!

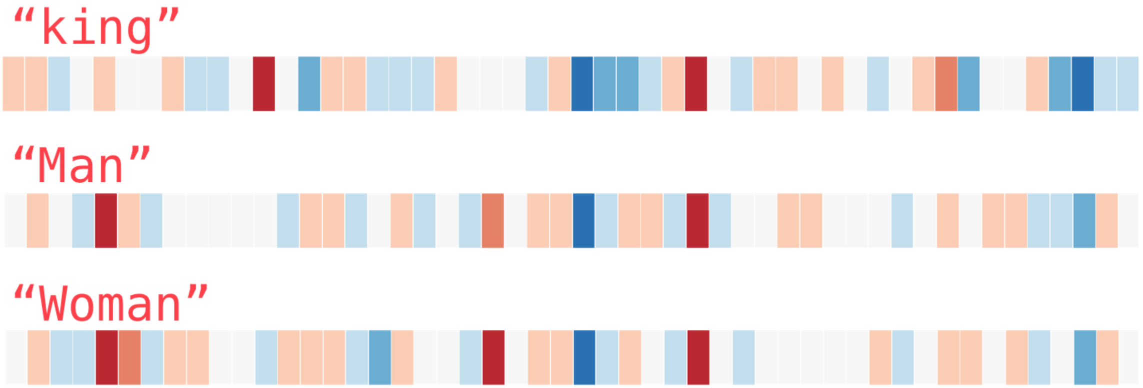
How does an LM “understand” word meaning?

- In order to predict the likelihood of a word, we must have some sense of its meaning
- Some words have similar meanings, and can easily fit in the same place
- In the same way CNNs convert an image into a set of feature maps, we can convert a word into a set of abstract linguistic features
- Word2Vec: 300 features
- GPT-3: 12,888

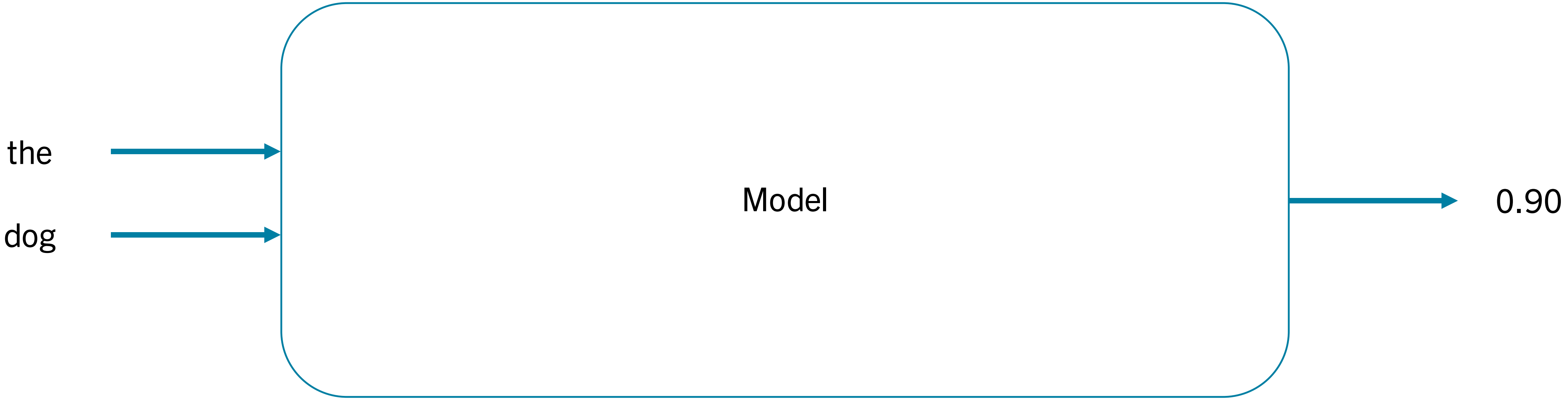


How does an LM “understand” word meaning?

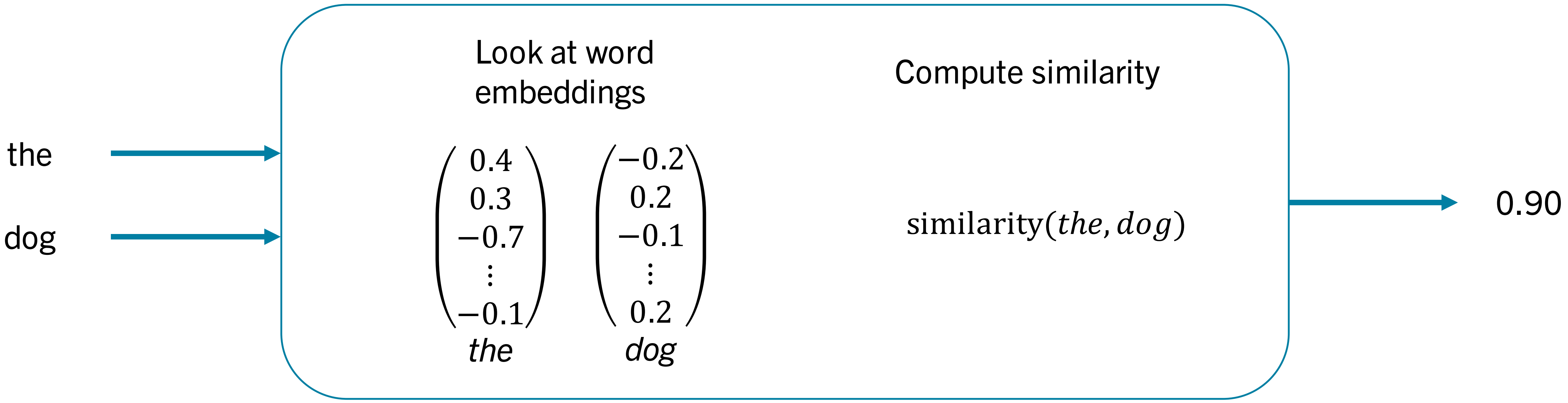
- Key concept of word embeddings: similar words should have similar vectors
- How do we accomplish this?



Building Word Embeddings



Building Word Embeddings



the

dog

Look at word embeddings

$$\begin{pmatrix} 0.4 \\ 0.3 \\ -0.7 \\ \vdots \\ -0.1 \end{pmatrix} \quad \begin{pmatrix} -0.2 \\ 0.2 \\ -0.1 \\ \vdots \\ 0.2 \end{pmatrix}$$

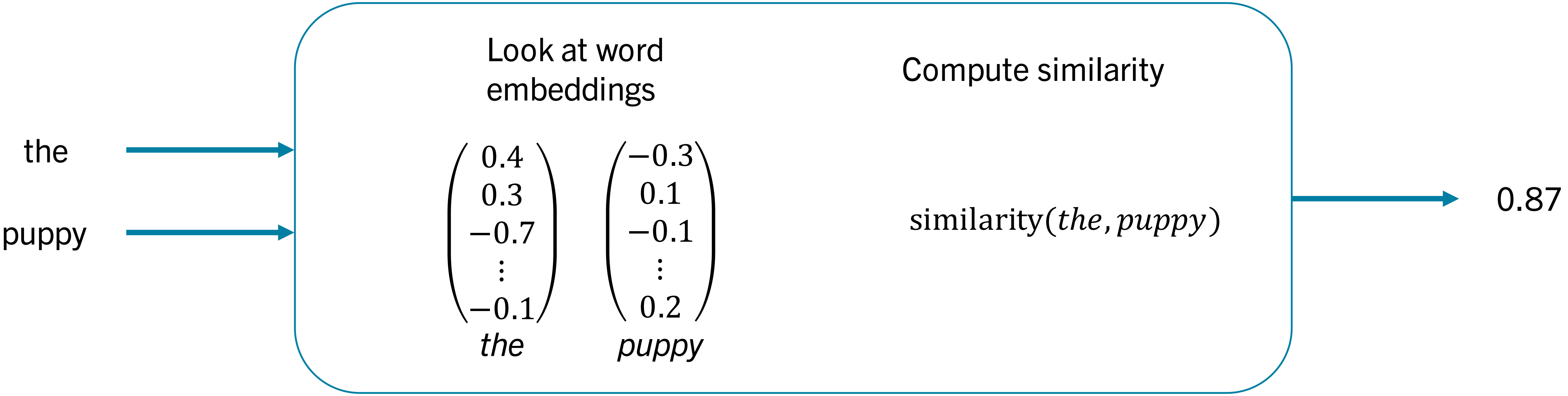
the *dog*

Compute similarity

$\text{similarity}(the, dog)$

0.90

Building Word Embeddings



the

puppy

Look at word embeddings

$$\begin{pmatrix} 0.4 \\ 0.3 \\ -0.7 \\ \vdots \\ -0.1 \end{pmatrix} \quad \begin{pmatrix} -0.3 \\ 0.1 \\ -0.1 \\ \vdots \\ 0.2 \end{pmatrix}$$

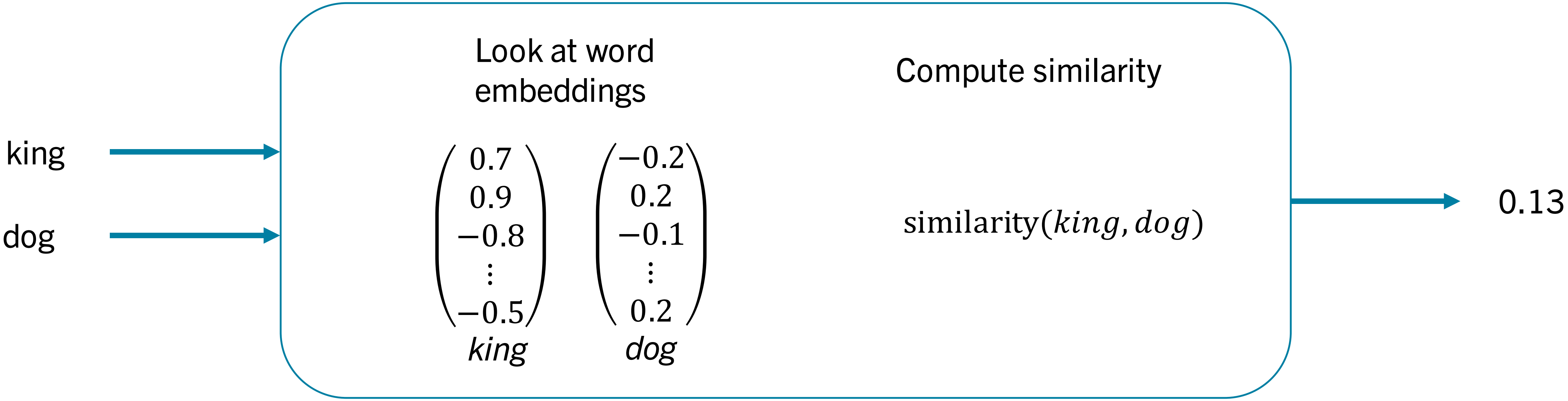
the *puppy*

Compute similarity

$$\mathrm{similarity}(\mathit{the}, \mathit{puppy})$$

0.87

Building Word Embeddings



king

dog

Look at word embeddings

$$\begin{pmatrix} 0.7 \\ 0.9 \\ -0.8 \\ \vdots \\ -0.5 \end{pmatrix} \qquad \begin{pmatrix} -0.2 \\ 0.2 \\ -0.1 \\ \vdots \\ 0.2 \end{pmatrix}$$

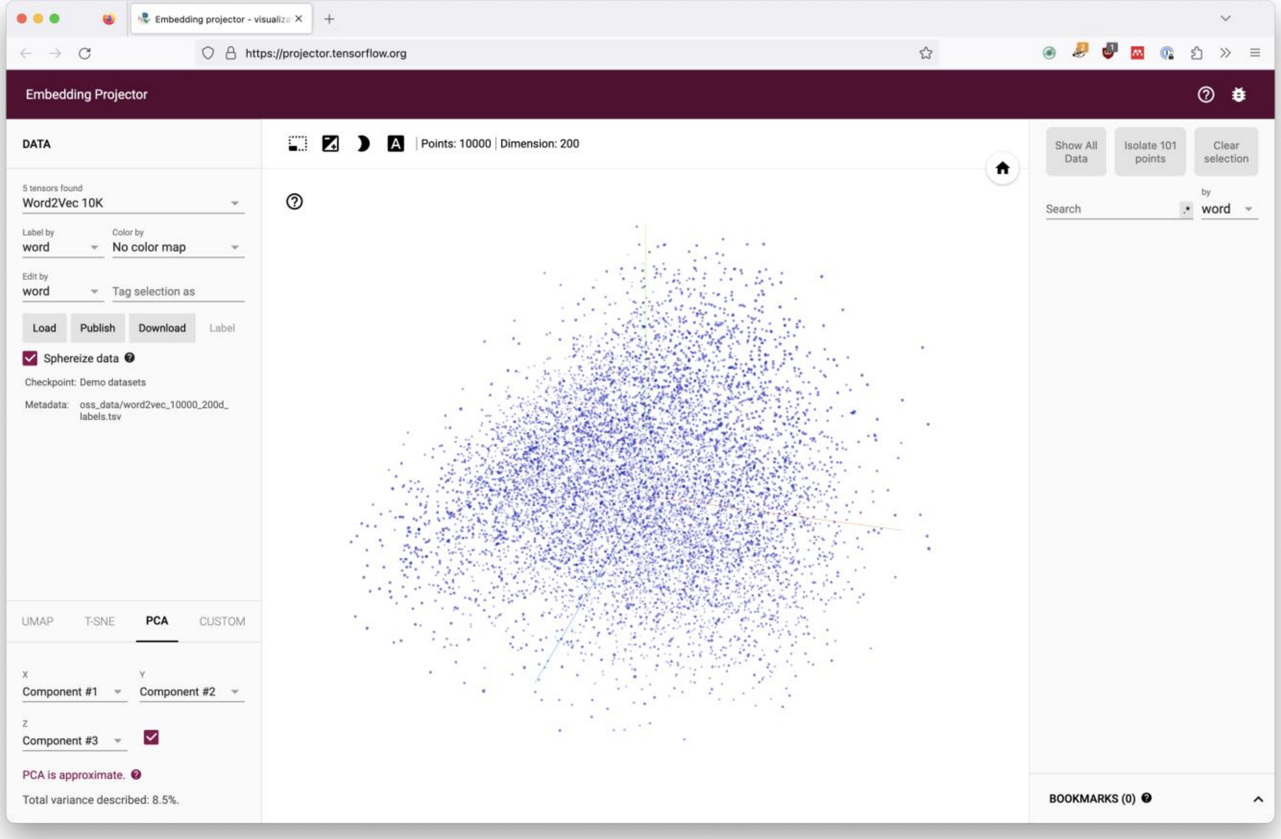
king *dog*

Compute similarity

$$\text{similarity}(king, dog)$$

0.13

Results



Points: 10000 | Dimension: 200

5 tensors found: Word2Vec 10K

Label by: word

Color by: No color map

Edit by: word

Tag selection as

☑ Sphereize data

Checkpoint: Demo datasets

Metadata: oss_data/word2vec_10000_200d_labels.tsv

X: Component #1

Y: Component #2

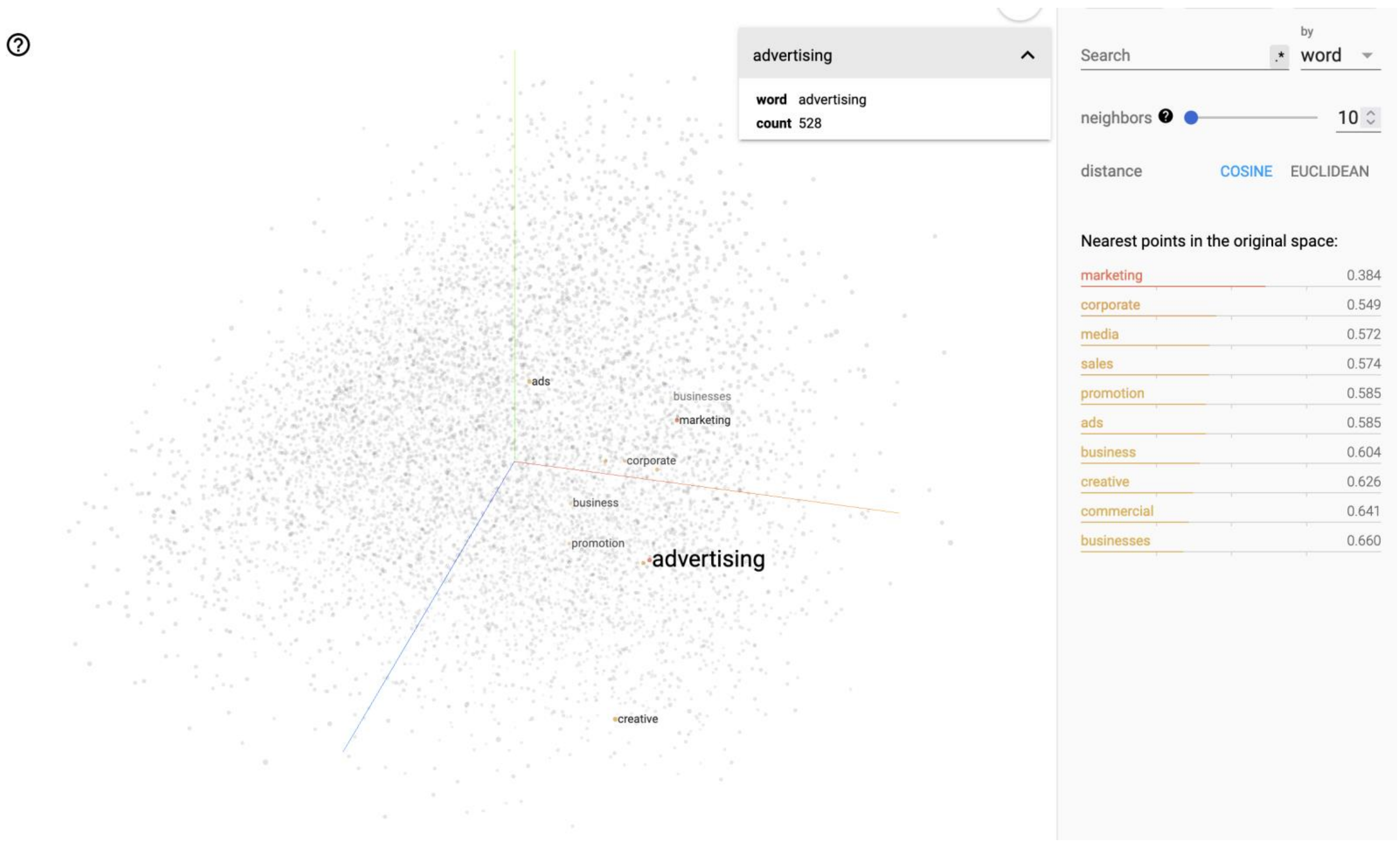
Z: Component #3 ☑

PCA is approximate.

Total variance described: 8.5%.

Search by: word

Results



advertising

word advertising
count 528

Search .* by word

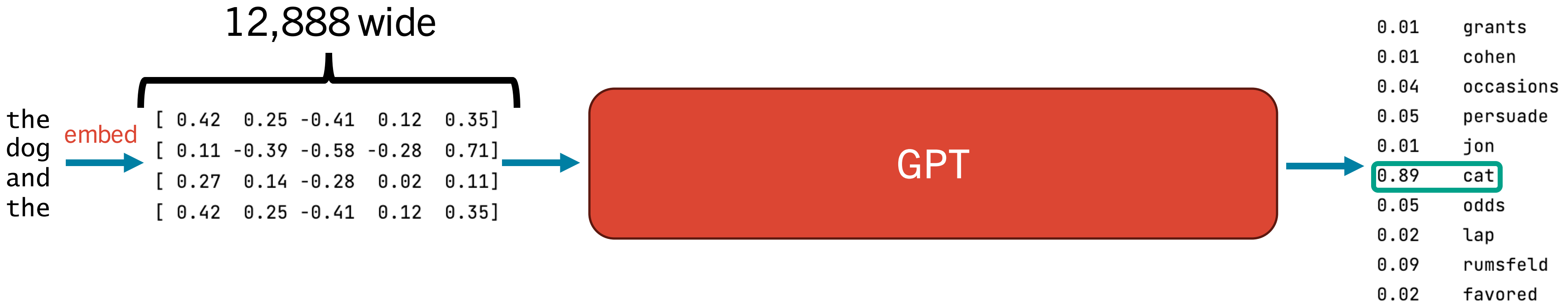
neighbors 10

distance COSINE EUCLIDEAN

Nearest points in the original space:

marketing	0.384
corporate	0.549
media	0.572
sales	0.574
promotion	0.585
ads	0.585
business	0.604
creative	0.626
commercial	0.641
businesses	0.660

Building GPT



12,888 wide

```
the
dog
and
the
```

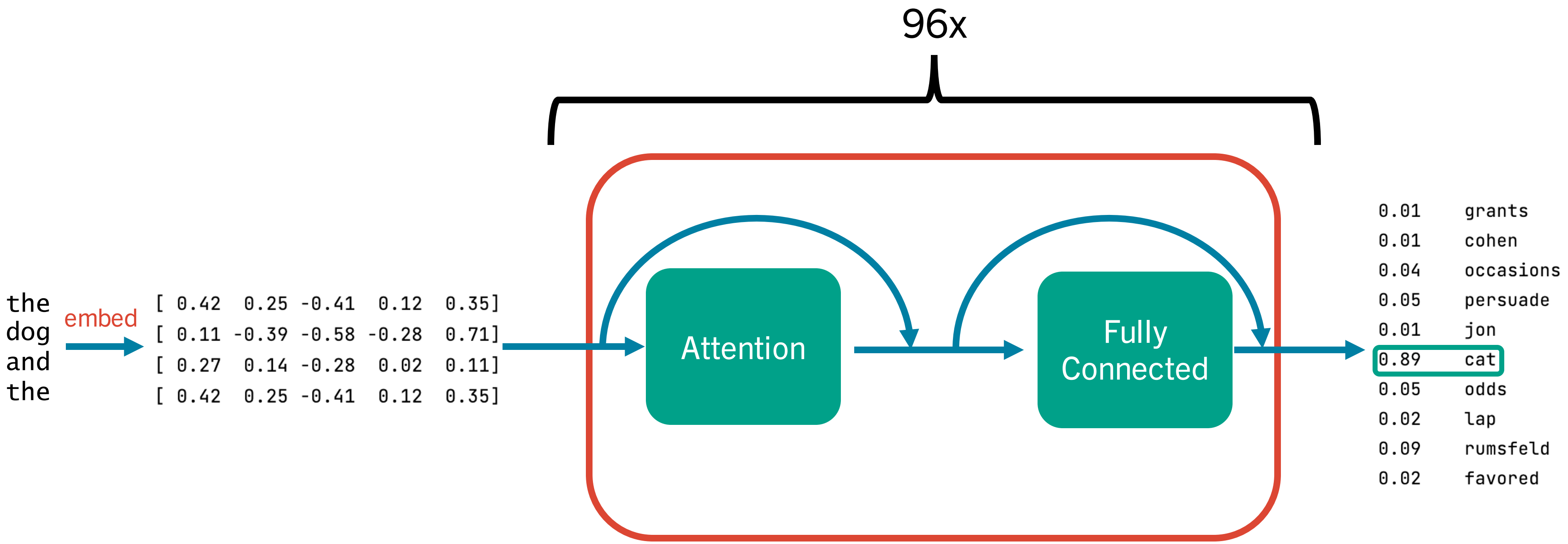
embed

```
[ 0.42  0.25 -0.41  0.12  0.35]
[ 0.11 -0.39 -0.58 -0.28  0.71]
[ 0.27  0.14 -0.28  0.02  0.11]
[ 0.42  0.25 -0.41  0.12  0.35]
```

GPT

0.01	grants
0.01	cohen
0.04	occasions
0.05	persuade
0.01	jon
0.89	cat
0.05	odds
0.02	lap
0.09	rumsfeld
0.02	favored

Building GPT: The Transfomer



96x

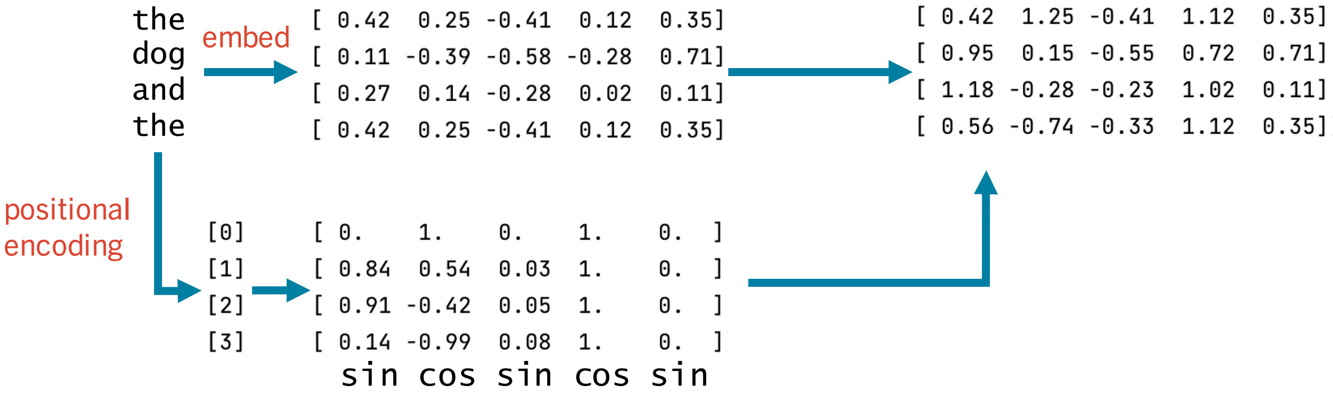
```
the                [ 0.42  0.25 -0.41  0.12  0.35]
dog    embed       [ 0.11 -0.39 -0.58 -0.28  0.71]
and                [ 0.27  0.14 -0.28  0.02  0.11]
the                [ 0.42  0.25 -0.41  0.12  0.35]
```

Attention

Fully Connected

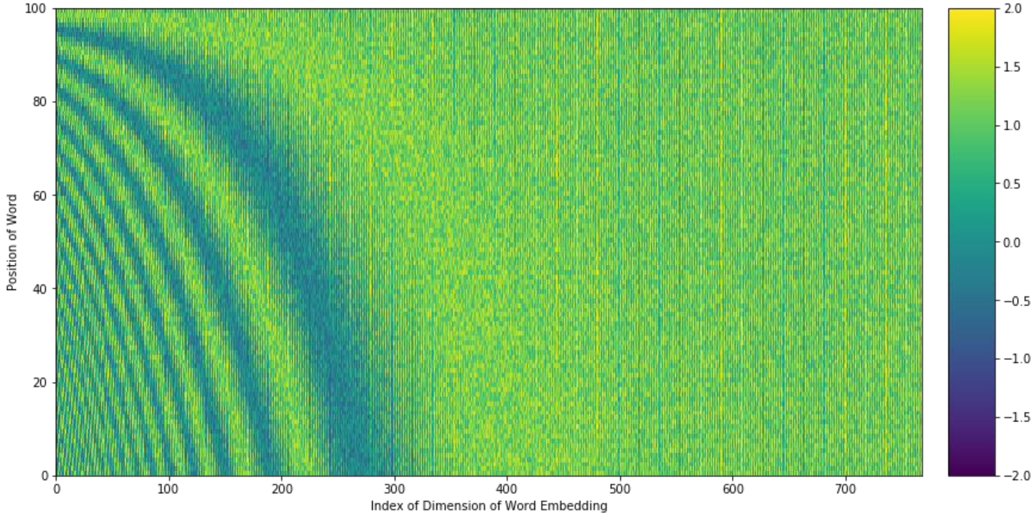
0.01	grants
0.01	cohen
0.04	occasions
0.05	persuade
0.01	jon
0.89	cat
0.05	odds
0.02	lap
0.09	rumsfeld
0.02	favored

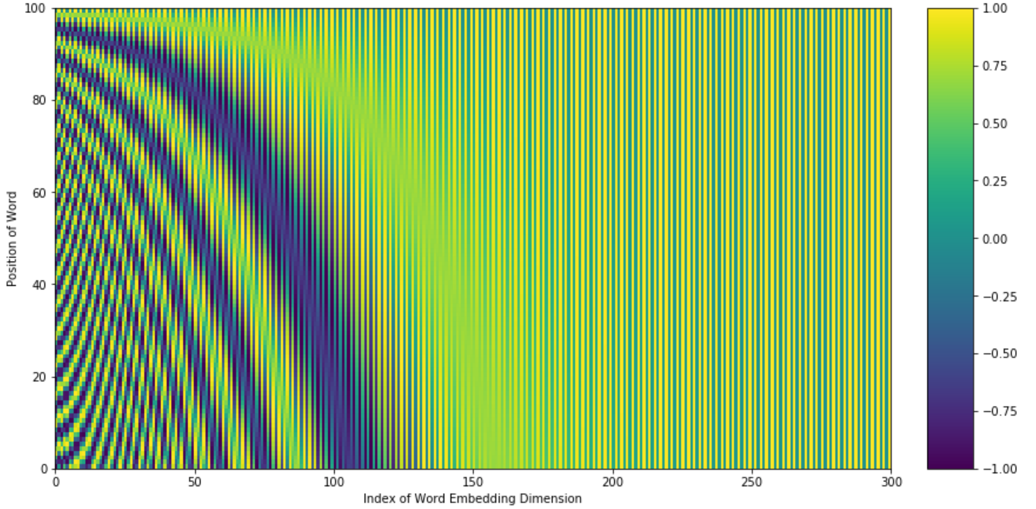
Building GPT: Positional Embedding



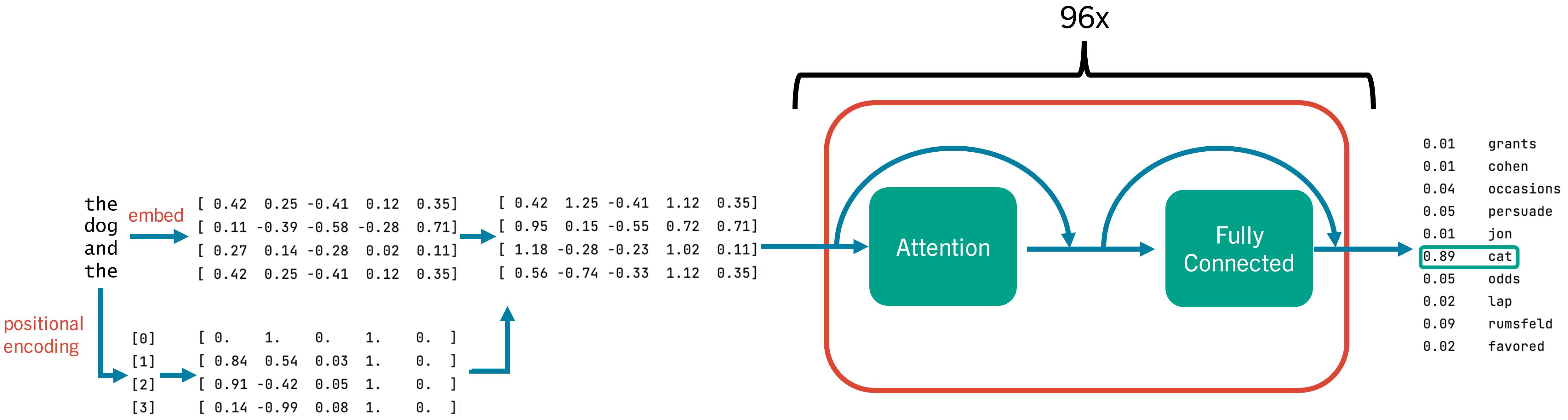
```
the          [ 0.42  0.25 -0.41  0.12  0.35]        [ 0.42  1.25 -0.41  1.12  0.35]
dog   embed  [ 0.11 -0.39 -0.58 -0.28  0.71]        [ 0.95  0.15 -0.55  0.72  0.71]
and          [ 0.27  0.14 -0.28  0.02  0.11]        [ 1.18 -0.28 -0.23  1.02  0.11]
the          [ 0.42  0.25 -0.41  0.12  0.35]        [ 0.56 -0.74 -0.33  1.12  0.35]

positional
encoding
        [0]  [ 0.    1.    0.    1.    0.  ]
        [1]  [ 0.84  0.54  0.03  1.    0.  ]
        [2]  [ 0.91 -0.42  0.05  1.    0.  ]
        [3]  [ 0.14 -0.99  0.08  1.    0.  ]
               sin  cos  sin  cos  sin
```





Building GPT



the
dog
and
the

embed

```
[ 0.42  0.25 -0.41  0.12  0.35]
[ 0.11 -0.39 -0.58 -0.28  0.71]
[ 0.27  0.14 -0.28  0.02  0.11]
[ 0.42  0.25 -0.41  0.12  0.35]
```

positional
encoding

```
[0]     [ 0.    1.    0.    1.    0.  ]
[1]     [ 0.84  0.54  0.03  1.    0.  ]
[2]     [ 0.91 -0.42  0.05  1.    0.  ]
[3]     [ 0.14 -0.99  0.08  1.    0.  ]
```

```
[ 0.42  1.25 -0.41  1.12  0.35]
[ 0.95  0.15 -0.55  0.72  0.71]
[ 1.18 -0.28 -0.23  1.02  0.11]
[ 0.56 -0.74 -0.33  1.12  0.35]
```

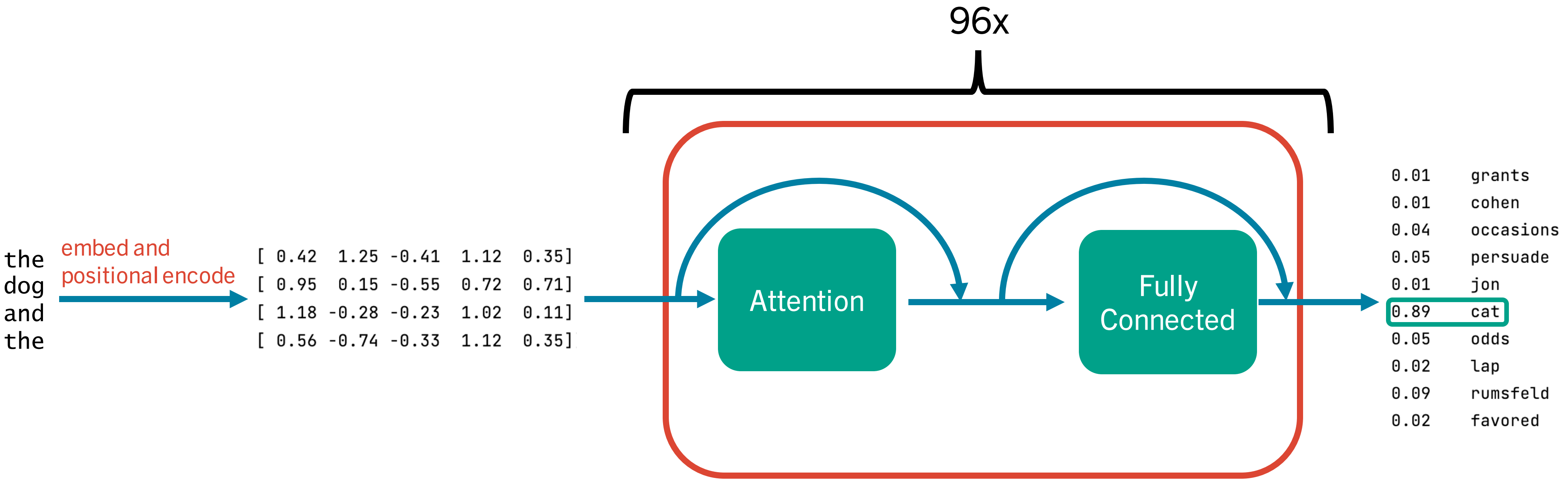
96x

Attention

Fully Connected

0.01	grants
0.01	cohen
0.04	occasions
0.05	persuade
0.01	jon
0.89	cat
0.05	odds
0.02	lap
0.09	rumsfeld
0.02	favored

Building GPT



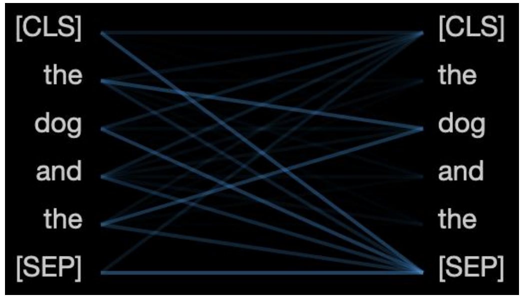
the
dog
and
the

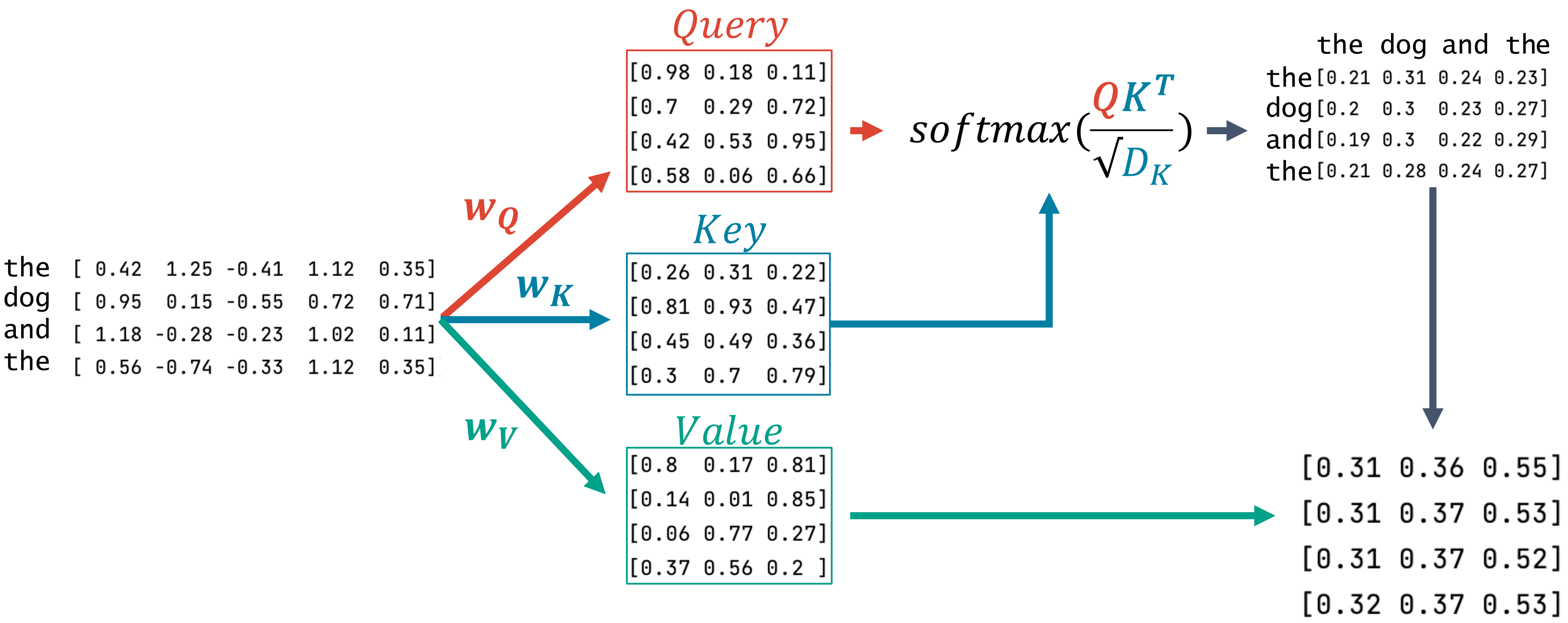
embed and positional encode

```
[ 0.42  1.25 -0.41  1.12  0.35]
[ 0.95  0.15 -0.55  0.72  0.71]
[ 1.18 -0.28 -0.23  1.02  0.11]
[ 0.56 -0.74 -0.33  1.12  0.35]
```

0.01	grants
0.01	cohen
0.04	occasions
0.05	persuade
0.01	jon
0.89	cat
0.05	odds
0.02	lap
0.09	rumsfeld
0.02	favored

Building GPT: Attention





```
the [ 0.42  1.25 -0.41  1.12  0.35]
dog [ 0.95  0.15 -0.55  0.72  0.71]
and [ 1.18 -0.28 -0.23  1.02  0.11]
the [ 0.56 -0.74 -0.33  1.12  0.35]
```

w_Q w_K w_V

Query

```
[0.98 0.18 0.11]
[0.7  0.29 0.72]
[0.42 0.53 0.95]
[0.58 0.06 0.66]
```

Key

```
[0.26 0.31 0.22]
[0.81 0.93 0.47]
[0.45 0.49 0.36]
[0.3  0.7  0.79]
```

Value

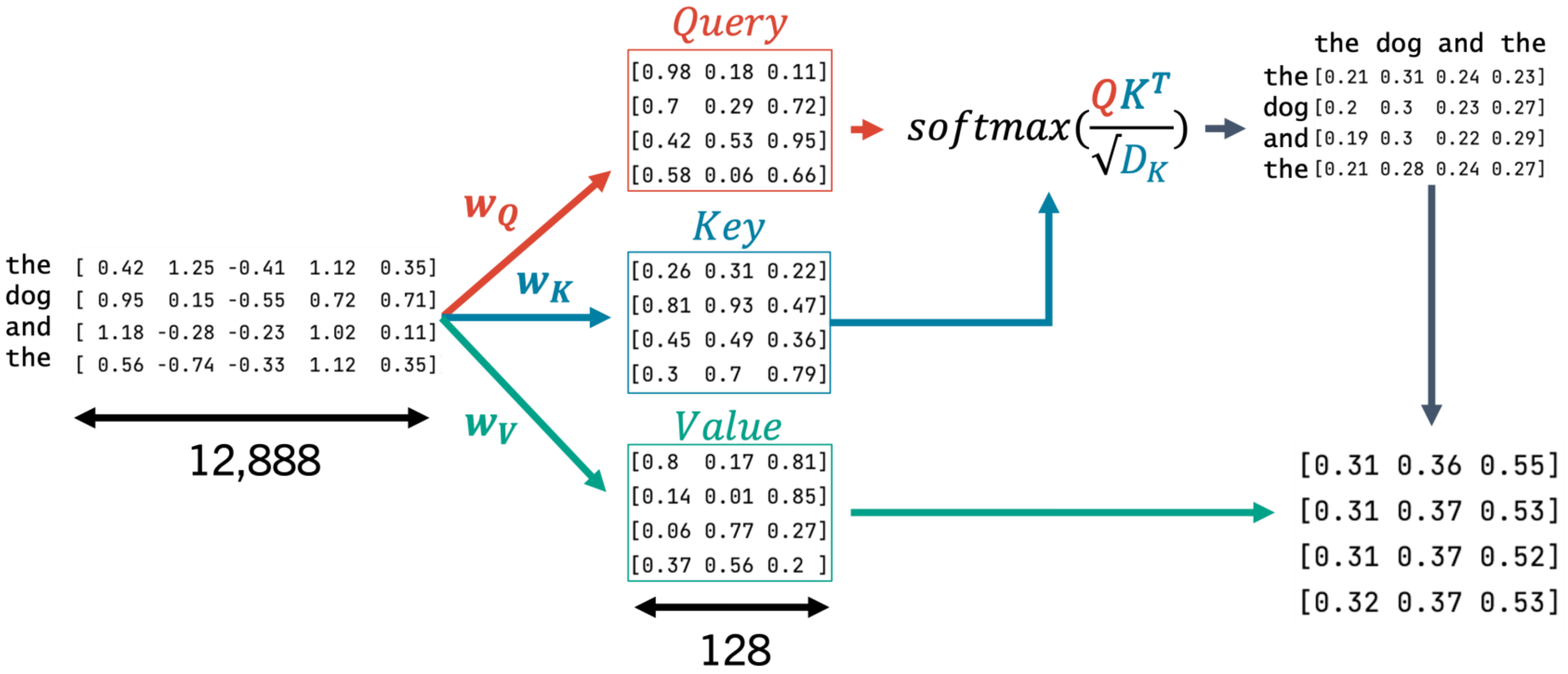
```
[0.8  0.17 0.81]
[0.14 0.01 0.85]
[0.06 0.77 0.27]
[0.37 0.56 0.2 ]
```

$$softmax(\frac{QK^T}{\sqrt{D_K}})$$

```
    the  dog  and  the
the[0.21 0.31 0.24 0.23]
dog[0.2  0.3  0.23 0.27]
and[0.19 0.3  0.22 0.29]
the[0.21 0.28 0.24 0.27]
```

```
[0.31 0.36 0.55]
[0.31 0.37 0.53]
[0.31 0.37 0.52]
[0.32 0.37 0.53]
```

Building GPT: Attention



```
the [ 0.42  1.25 -0.41  1.12  0.35]
dog [ 0.95  0.15 -0.55  0.72  0.71]
and [ 1.18 -0.28 -0.23  1.02  0.11]
the [ 0.56 -0.74 -0.33  1.12  0.35]
```

12,888

w_Q w_K w_V

Query

```
[0.98 0.18 0.11]
[0.7  0.29 0.72]
[0.42 0.53 0.95]
[0.58 0.06 0.66]
```

Key

```
[0.26 0.31 0.22]
[0.81 0.93 0.47]
[0.45 0.49 0.36]
[0.3  0.7  0.79]
```

Value

```
[0.8  0.17 0.81]
[0.14 0.01 0.85]
[0.06 0.77 0.27]
[0.37 0.56 0.2 ]
```

128

$$softmax(\frac{QK^T}{\sqrt{D_K}})$$

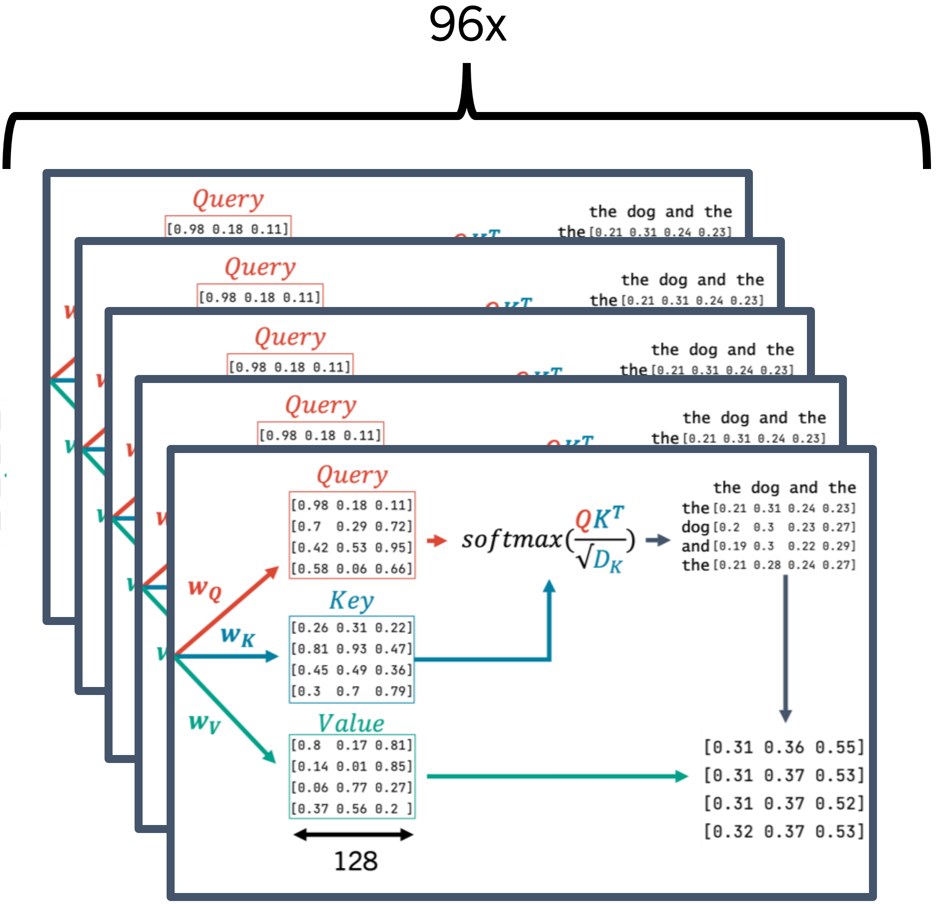
	the	dog	and	the
the	[0.21	0.31	0.24	0.23]
dog	[0.2	0.3	0.23	0.27]
and	[0.19	0.3	0.22	0.29]
the	[0.21	0.28	0.24	0.27]

```
[0.31 0.36 0.55]
[0.31 0.37 0.53]
[0.31 0.37 0.52]
[0.32 0.37 0.53]
```

Building GPT: Attention

```
the  [ 0.42  1.25 -0.41  1.12  0.35]
dog  [ 0.95  0.15 -0.55  0.72  0.71]
and  [ 1.18 -0.28 -0.23  1.02  0.11]
the  [ 0.56 -0.74 -0.33  1.12  0.35]
```

12,888



Query

```
[0.98 0.18 0.11]
[0.7  0.29 0.72]
[0.42 0.53 0.95]
[0.58 0.06 0.66]
```

$$softmax(\frac{QK^T}{\sqrt{D_K}})$$

Key

```
[0.26 0.31 0.22]
[0.81 0.93 0.47]
[0.45 0.49 0.36]
[0.3  0.7  0.79]
```

Value

```
[0.8  0.17 0.81]
[0.14 0.01 0.85]
[0.06 0.77 0.27]
[0.37 0.56 0.2 ]
```

```
     the  dog  and  the
the [0.21 0.31 0.24 0.23]
dog [0.2  0.3  0.23 0.27]
and [0.19 0.3  0.22 0.29]
the [0.21 0.28 0.24 0.27]
```

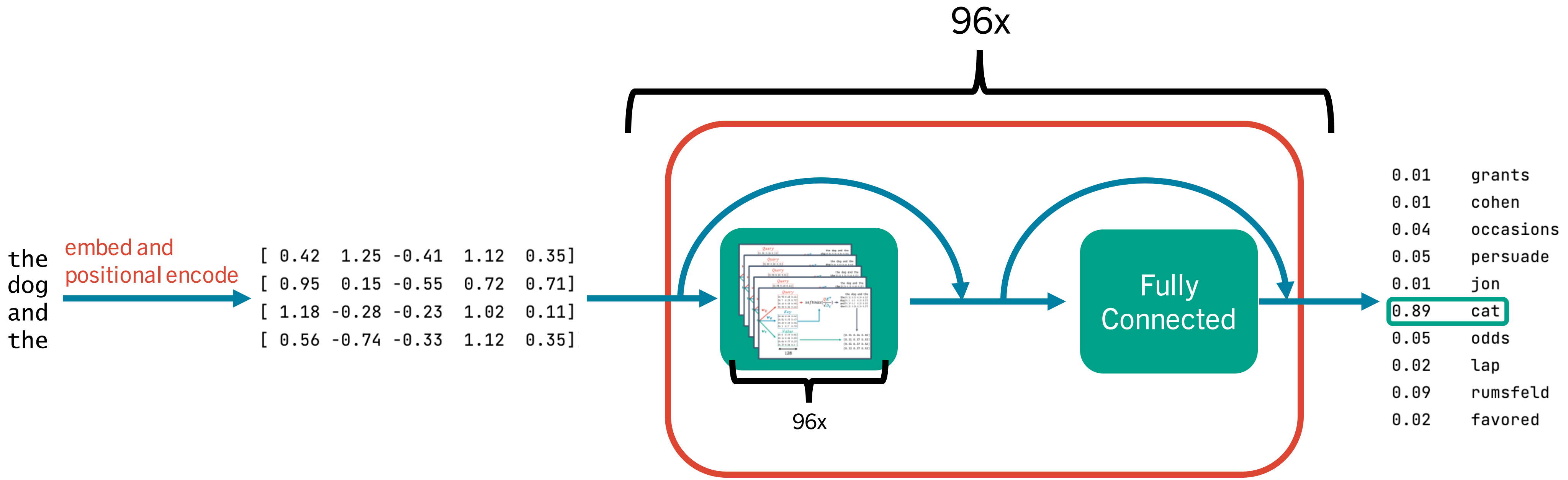
```
[0.31 0.36 0.55]
[0.31 0.37 0.53]
[0.31 0.37 0.52]
[0.32 0.37 0.53]
```

96x128

```
[0.23 0.15 0.61 0.97 0.  ]
[0.43 0.09 0.91 0.94 0.88]
[0.02 0.93 0.68 0.08 0.99]
[0.15 0.86 0.1  0.93 0.72]
```

12,888

Building GPT



the
dog
and
the

embed and positional encode

```
[ 0.42  1.25 -0.41  1.12  0.35]
[ 0.95  0.15 -0.55  0.72  0.71]
[ 1.18 -0.28 -0.23  1.02  0.11]
[ 0.56 -0.74 -0.33  1.12  0.35]
```

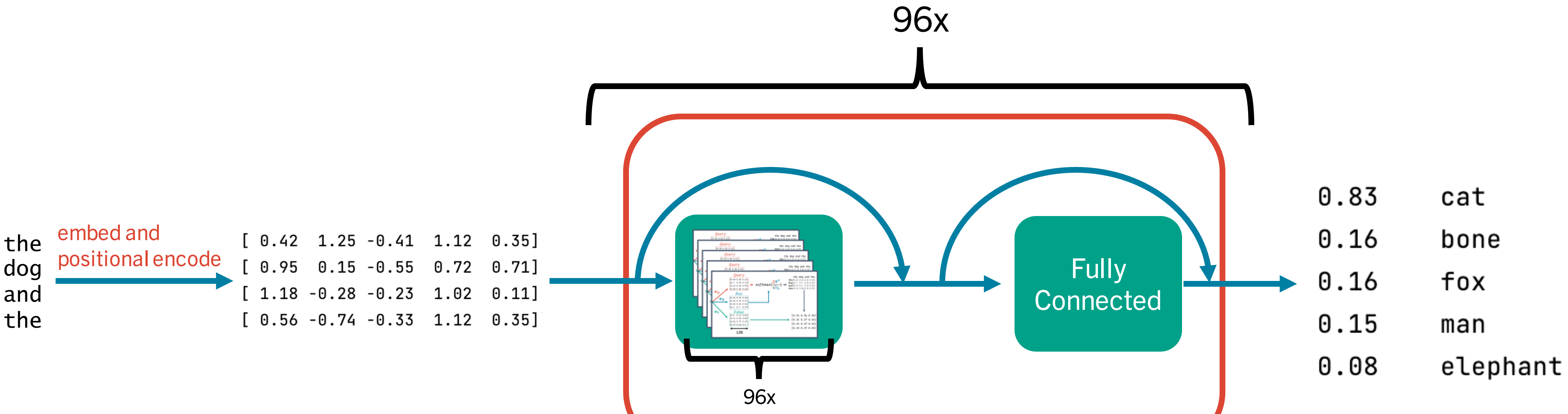
96x

96x

Fully Connected

0.01	grants
0.01	cohen
0.04	occasions
0.05	persuade
0.01	jon
0.89	cat
0.05	odds
0.02	lap
0.09	rumsfeld
0.02	favored

Building GPT: Top-P

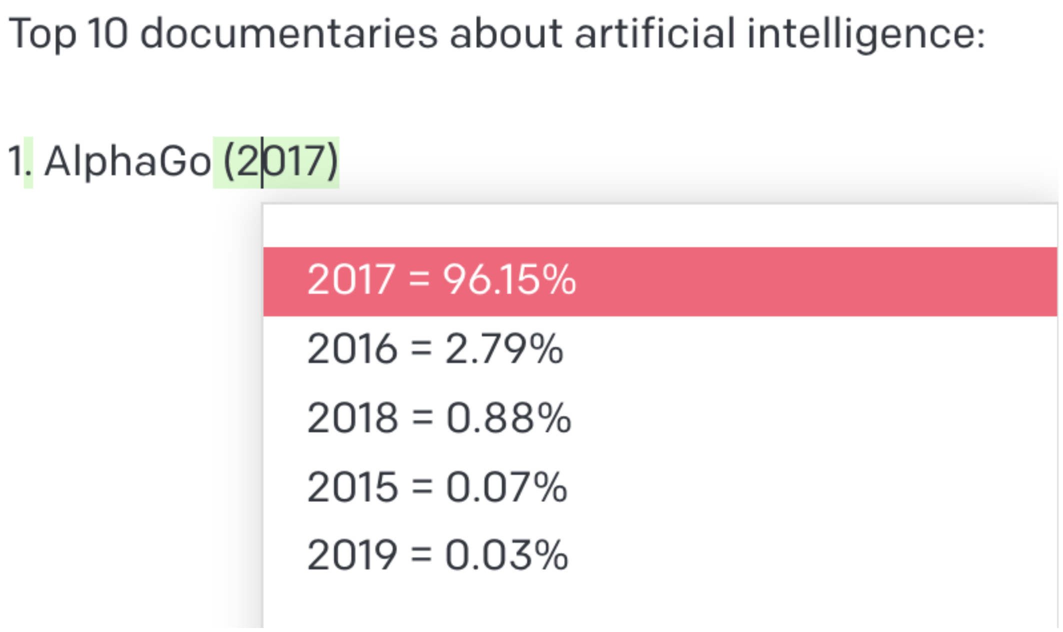


the
dog
and
the

```
[ 0.42  1.25 -0.41  1.12  0.35]
[ 0.95  0.15 -0.55  0.72  0.71]
[ 1.18 -0.28 -0.23  1.02  0.11]
[ 0.56 -0.74 -0.33  1.12  0.35]
```

0.83	cat
0.16	bone
0.16	fox
0.15	man
0.08	elephant

Building GPT: Top-P

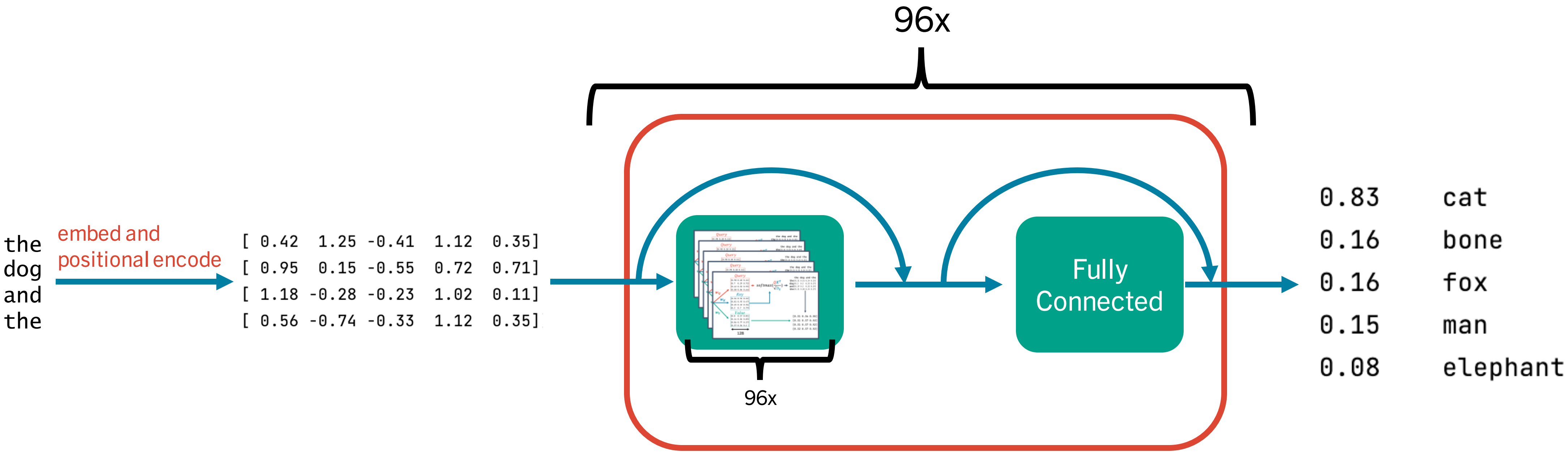


Top 10 documentaries about artificial intelligence:

1. AlphaGo (2017)

2017 = 96.15%
2016 = 2.79%
2018 = 0.88%
2015 = 0.07%
2019 = 0.03%

Building GPT



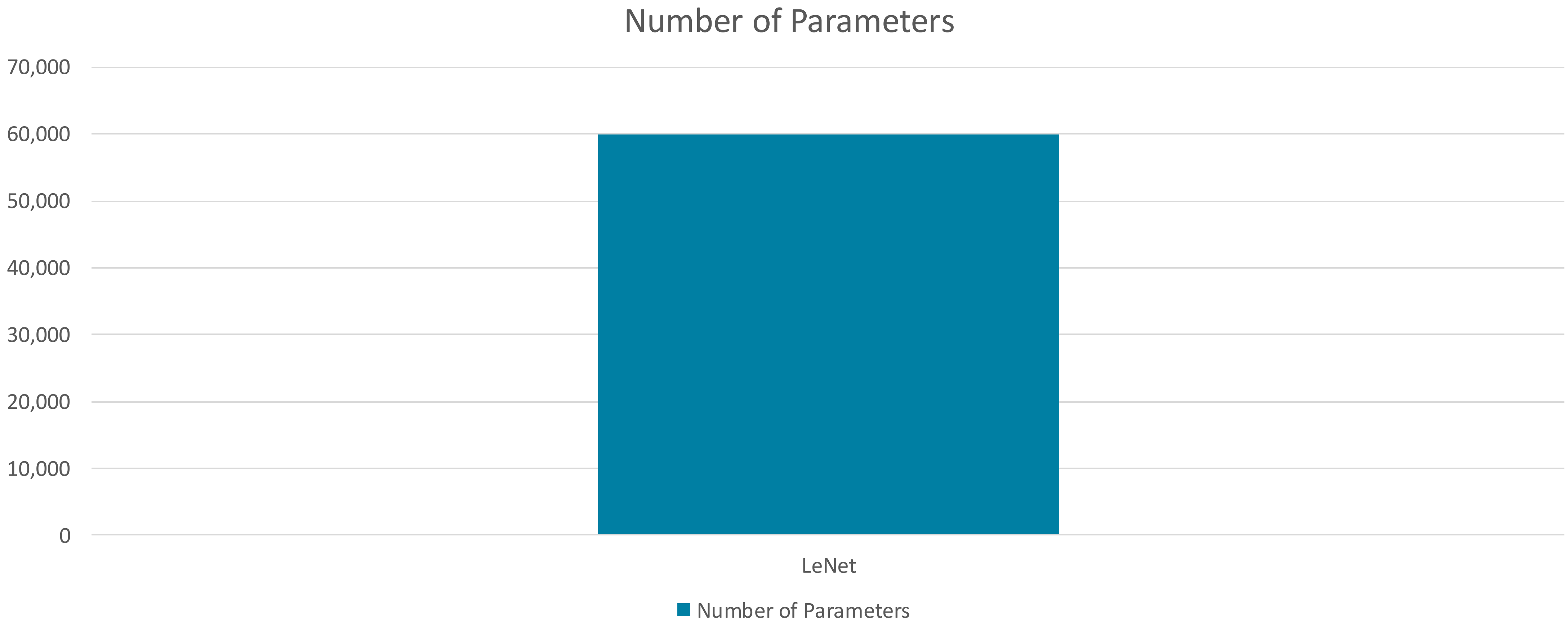
the
dog
and
the

embed and positional encode

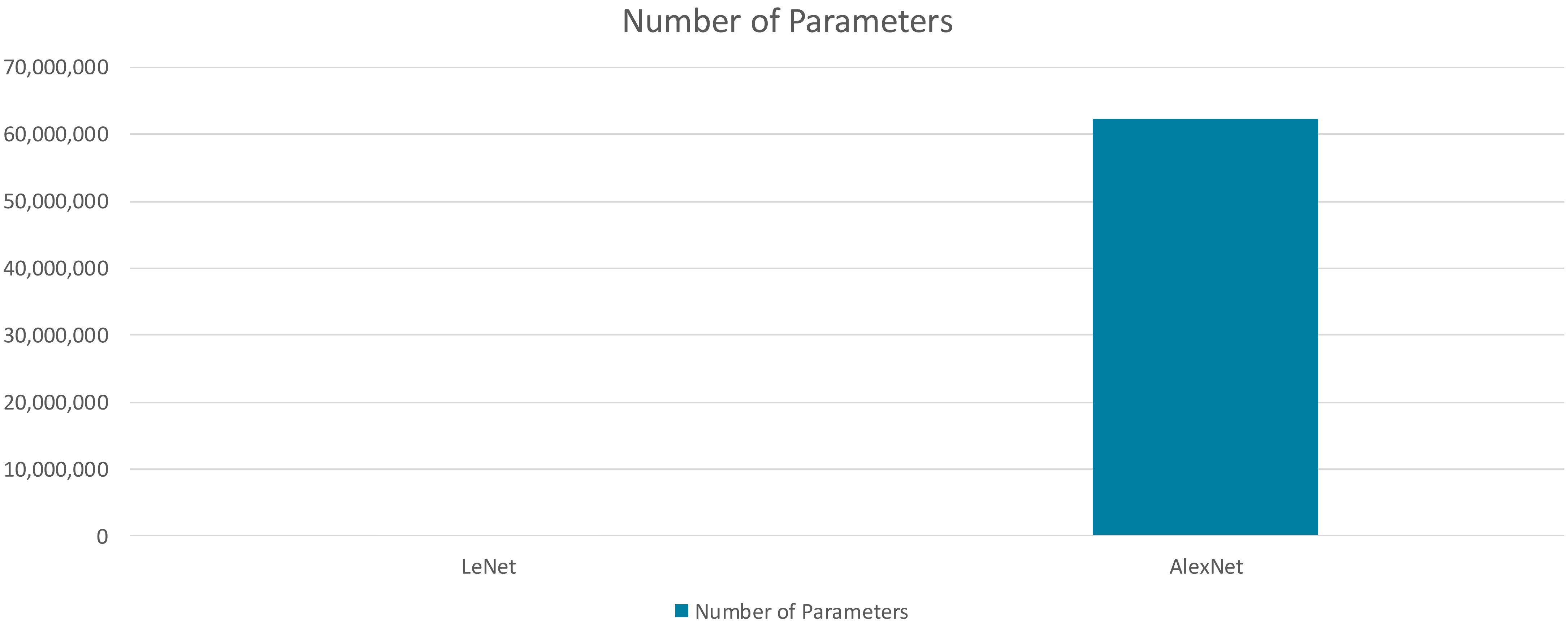
```
[ 0.42  1.25 -0.41  1.12  0.35]
[ 0.95  0.15 -0.55  0.72  0.71]
[ 1.18 -0.28 -0.23  1.02  0.11]
[ 0.56 -0.74 -0.33  1.12  0.35]
```

0.83	cat
0.16	bone
0.16	fox
0.15	man
0.08	elephant

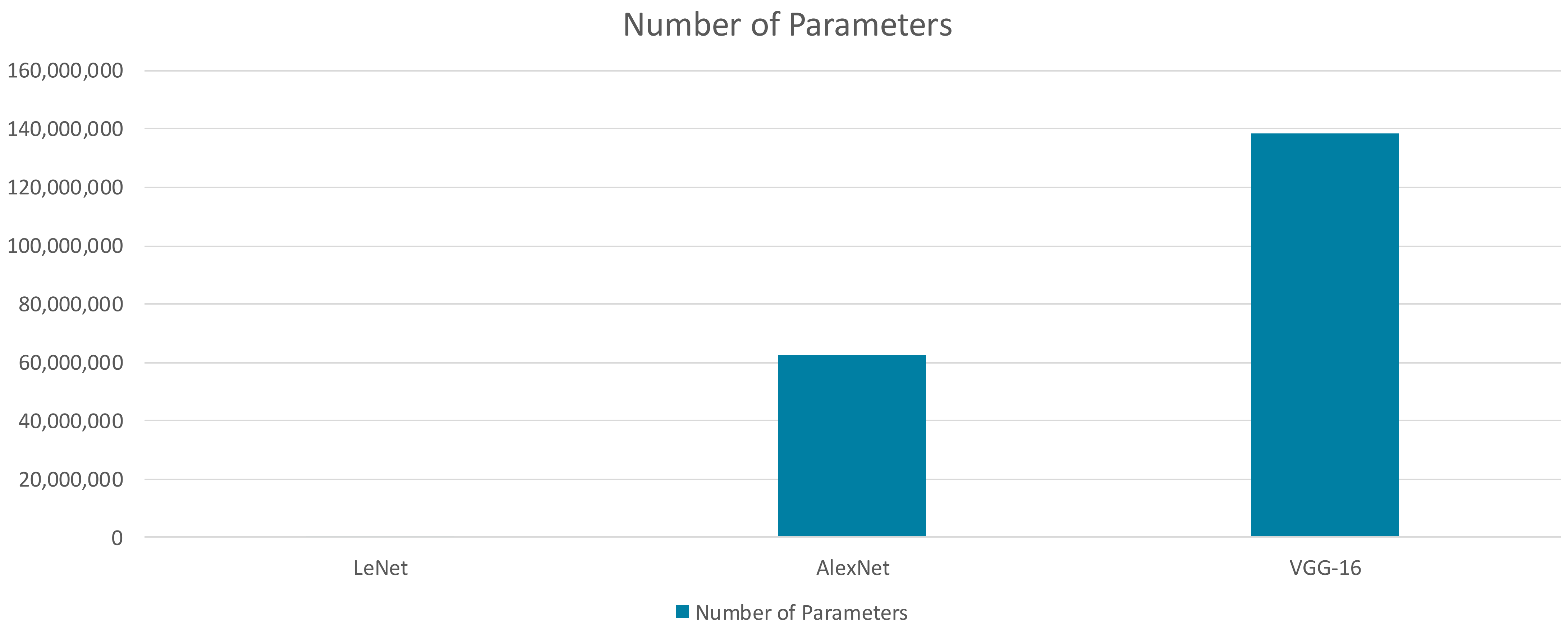
Scale of GPT


Number of Parameters
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
LeNet
■ Number of Parameters

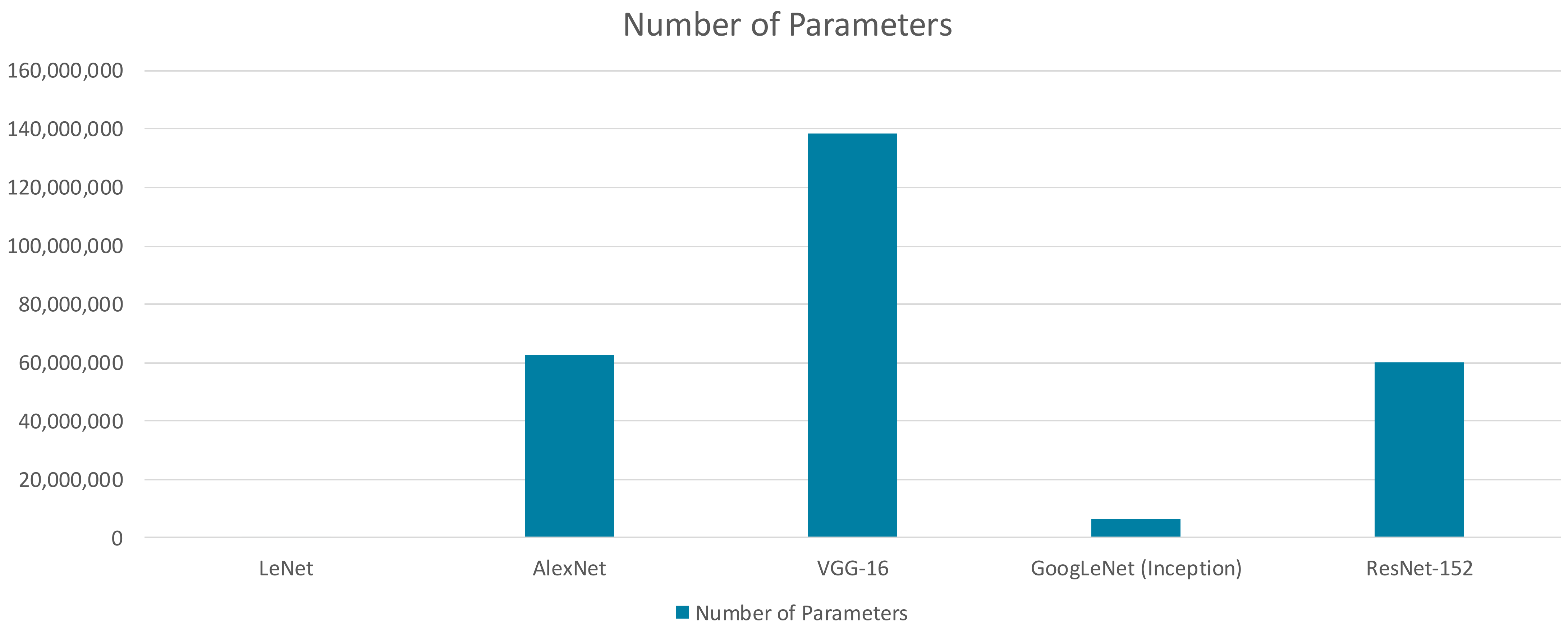
Scale of GPT


Number of Parameters
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
LeNet
AlexNet
■ Number of Parameters

Scale of GPT

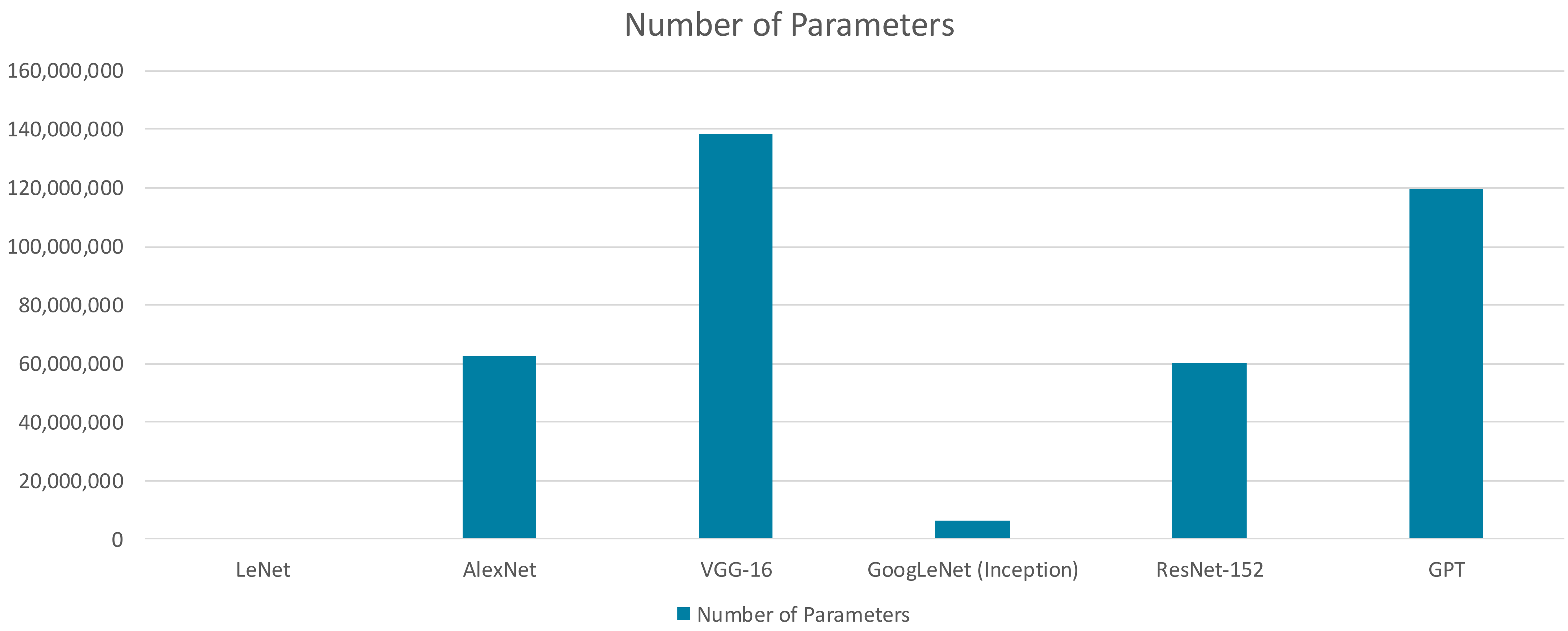

Number of Parameters
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
LeNet
AlexNet
VGG-16
■ Number of Parameters

Scale of GPT

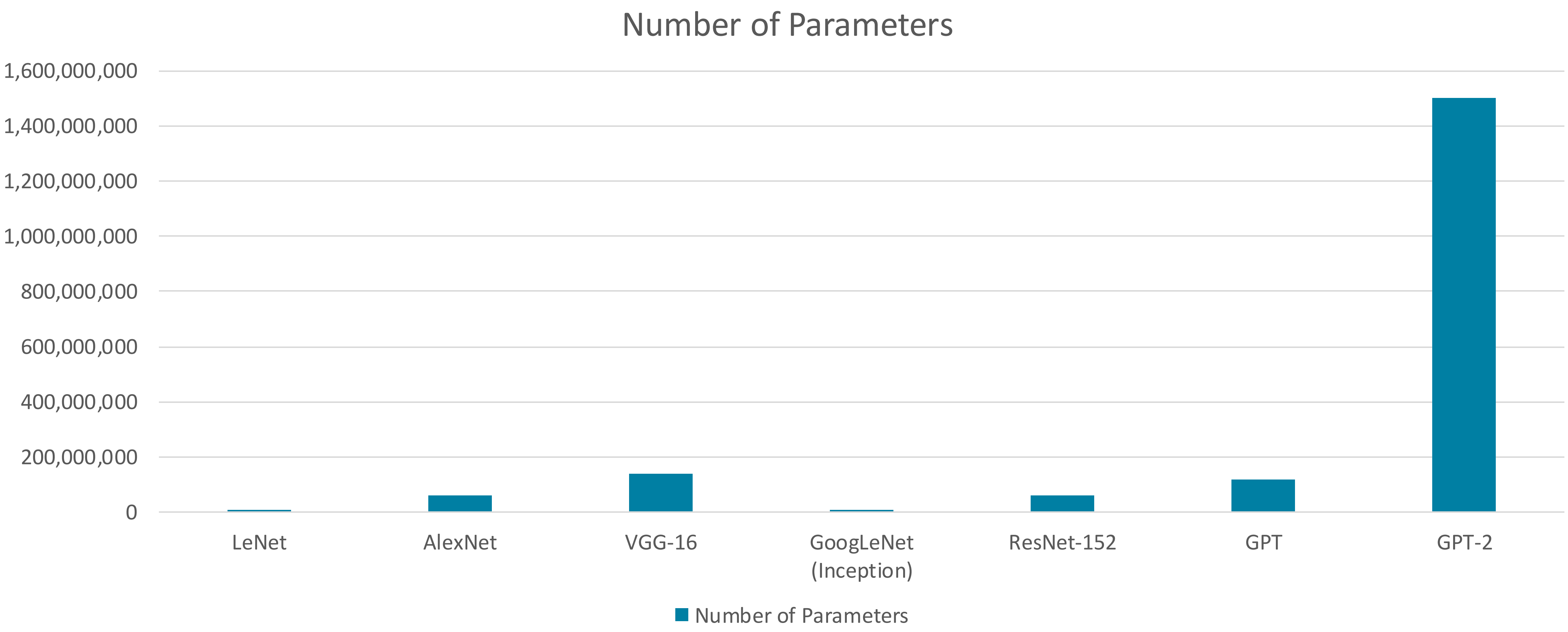


Number of Parameters
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
LeNet
AlexNet
VGG-16
GoogLeNet (Inception)
ResNet-152
Number of Parameters

Scale of GPT

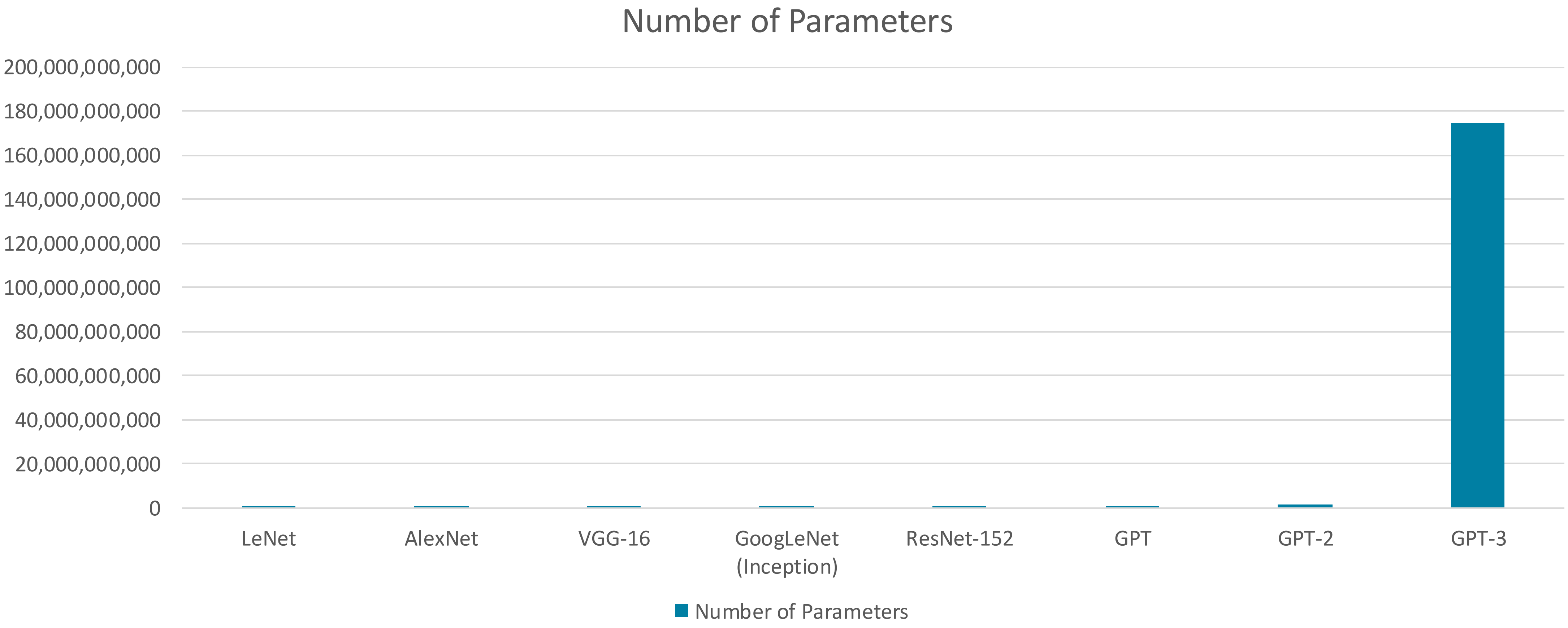

Number of Parameters
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
LeNet
AlexNet
VGG-16
GoogLeNet (Inception)
ResNet-152
GPT
Number of Parameters

Scale of GPT

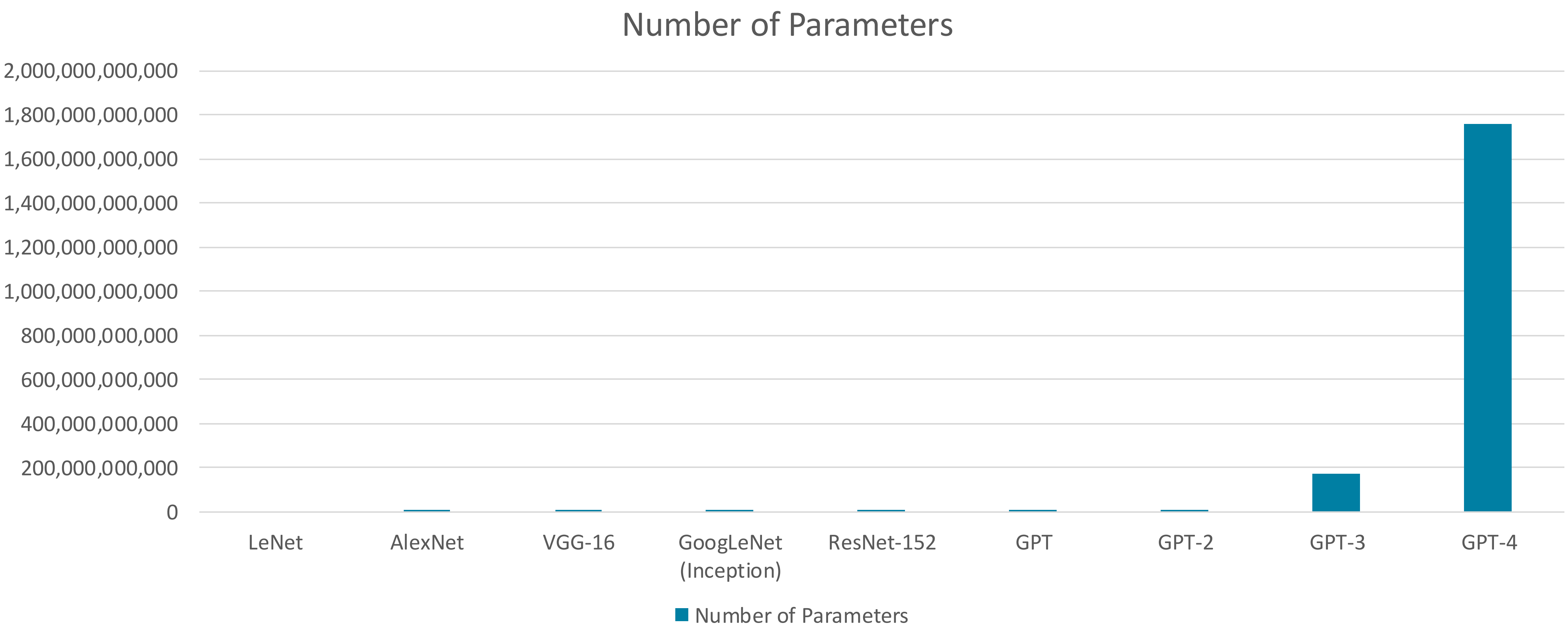


Number of Parameters
1,600,000,000
1,400,000,000
1,200,000,000
1,000,000,000
800,000,000
600,000,000
400,000,000
200,000,000
0
LeNet
AlexNet
VGG-16
GoogLeNet (Inception)
ResNet-152
GPT
GPT-2
■ Number of Parameters

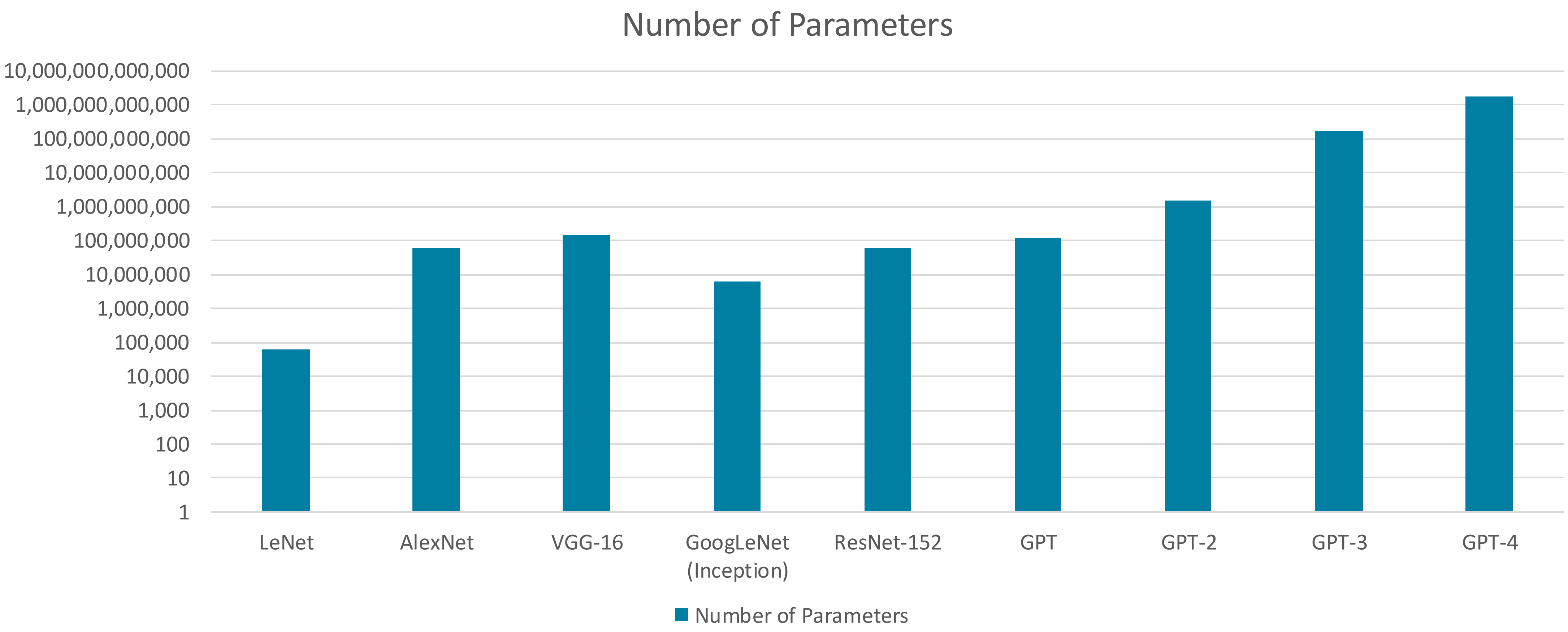
Scale of GPT


Number of Parameters
200,000,000,000
180,000,000,000
160,000,000,000
140,000,000,000
120,000,000,000
100,000,000,000
80,000,000,000
60,000,000,000
40,000,000,000
20,000,000,000
0
LeNet
AlexNet
VGG-16
GoogLeNet (Inception)
ResNet-152
GPT
GPT-2
GPT-3
■ Number of Parameters

Scale of GPT

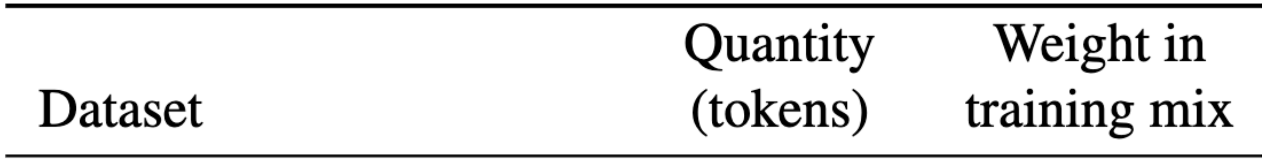

Number of Parameters
2,000,000,000,000
1,800,000,000,000
1,600,000,000,000
1,400,000,000,000
1,200,000,000,000
1,000,000,000,000
800,000,000,000
600,000,000,000
400,000,000,000
200,000,000,000
0
LeNet
AlexNet
VGG-16
GoogLeNet (Inception)
ResNet-152
GPT
GPT-2
GPT-3
GPT-4
■ Number of Parameters

Scale of GPT


Number of Parameters
10,000,000,000,000
1,000,000,000,000
100,000,000,000
10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
LeNet
AlexNet
VGG-16
GoogLeNet (Inception)
ResNet-152
GPT
GPT-2
GPT-3
GPT-4
■ Number of Parameters

GPT's Training Data

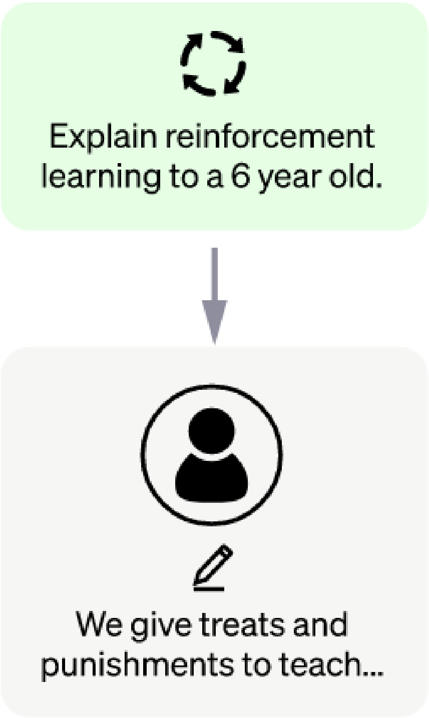
- 1 token ≈ ¾ word
- Some datasets are sampled more times than others
- Common Crawl: billions of webpages collected over 7 years
- Webtext2: Dataset of webpages that have been shared on Reddit
- Books1: Free ebooks (?)
- Books2: Secret!
- English Wikipedia



Dataset	Quantity (tokens)	Weight in training mix

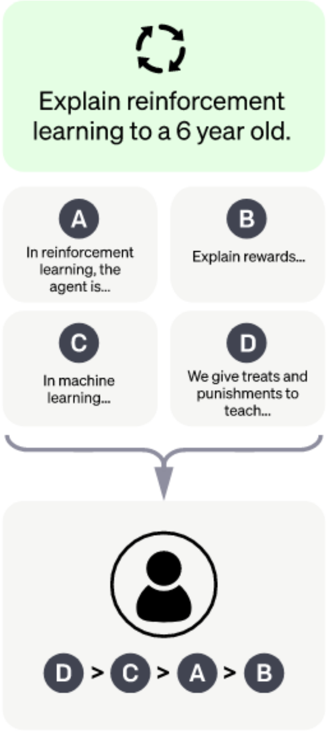
The training innovation of ChatGPT

Human annotators write answers to questions


Explain reinforcement learning to a 6 year old.
We give treats and punishments to teach...

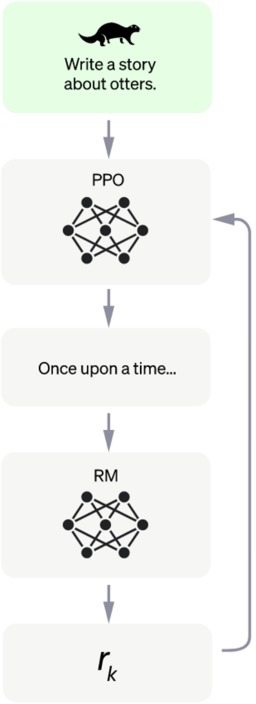
The generalist GPT model is taught from these Q&A pairs

Human annotators write more answers, and someone else ranks them


Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
D > C > A > B

A separate model learns to rate the quality of an answer

GPT writes answers to sampled questions

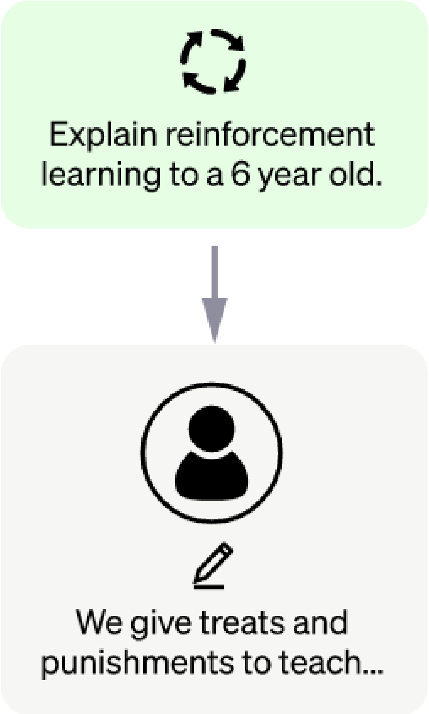


The reward model rates each answer, allowing GPT to keep learning

Figures: https://openai.com/blog/chatgpt/

The training innovation of ChatGPT

Human annotators write answers to questions

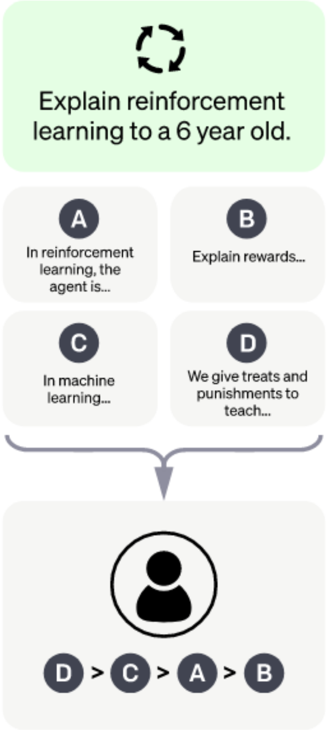


Explain reinforcement learning to a 6 year old.

We give treats and punishments to teach...

The generalist GPT model is taught from these Q&A pairs

Human annotators write more answers, and someone else ranks them



Explain reinforcement learning to a 6 year old.

A In reinforcement learning, the agent is...

B Explain rewards...

C In machine learning...

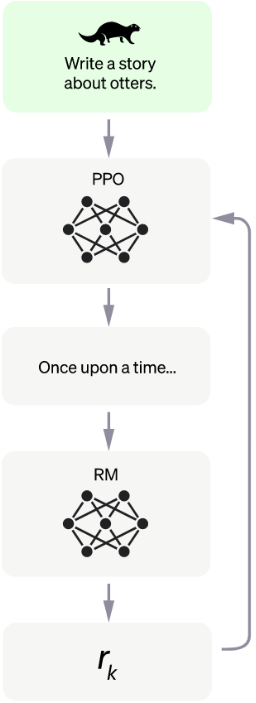
D We give treats and punishments to teach...

D > C > A > B

A separate model learns to rate the quality of an answer

No more humans involved!

GPT writes answers to sampled questions



Write a story about otters.

PPO

Once upon a time...

RM

r_k

The reward model rates each answer, allowing GPT to keep learning

Figures: https://openai.com/blog/chatgpt/

Winograd Schema

- ”Artificial language processing remains ten years away” – Tom Scott, 2020
- GPT-3 performance: 68.8%
- GPT-4 performance: 94.4%
- Today, 22 models outperform human baselines on the GLUE benchmark