香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 4901B

Large Language Models

Language Models

Junxian He

Sep 10, 2025

Discriminative vs. Generative Learning

Discriminative vs. Generative Learning

X

$p(y | x)$

Discriminative

Cat Y

Discriminative vs. Generative Learning

X

$p(y|x)$ Discriminative

Cat Y

Cat Y $p(y)$ prior dist

Generative $p(x|y)$

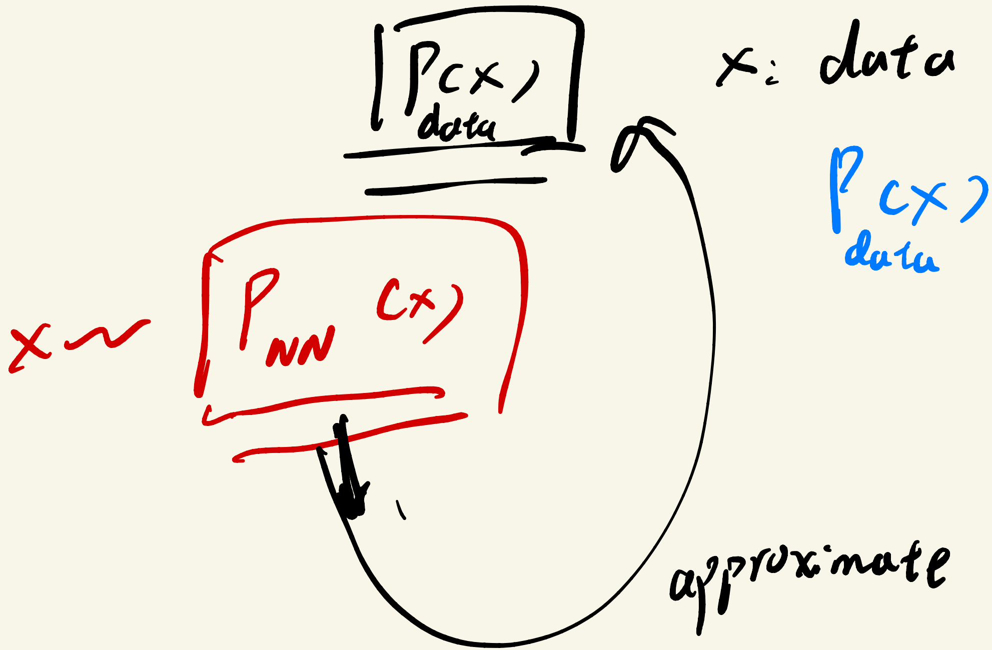
X

$$P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(y)P(x|y)}{P(x)} = \frac{P(y)P(x|y)}{\sum_y P(x,y)} = \frac{P(y)P(x|y)}{\sum_y P(y)P(x|y)}$$

~~$P(x,y)$~~ $P(y|x)$

$P(x,y) = P(y) \cdot P(x|y)$

Generative Modeling



Language Modeling: $p(\text{natural language})$

Generative:

If you can creat sth, → generative

you must understand it. → discriminate

Richard Feynman

If you don't understand sth

You can never generate it

the mouse { ate / ran }

Probability of Sequences

Probability of multiple random variables:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \,|\, x_{1:i-1})$$

Probability of Sequences

Probability of multiple random variables:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \,|\, x_{1:i-1})$$

Probability of language:

Probability of Sequences

Probability of multiple random variables:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \,|\, x_{1:i-1})$$

Probability of language:

$$\begin{aligned} p(\text{the, mouse, ate, the, cheese}) = \; & p(\text{the}) \\ & p(\text{mouse} \mid \text{the}) \\ & p(\text{ate} \mid \text{the, mouse}) \\ & p(\text{the} \mid \text{the, mouse, ate}) \\ & p(\text{cheese} \mid \text{the, mouse, ate, the}). \end{aligned}$$

Probability of Sequences

Probability of multiple random variables:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \mid x_{1:i-1})$$

Probability of language:

$$p(\text{the, mouse, ate, the, cheese}) = p(\text{the})\, p(\text{mouse} \mid \text{the})\, p(\text{ate} \mid \text{the, mouse})\, p(\text{the} \mid \text{the, mouse, ate})\, p(\text{cheese} \mid \text{the, mouse, ate, the}).$$

Autoregressive language models

$$P(\text{the mouse ate the cheese})$$

$$= P(\text{ate})\, P(\text{mouse} \mid \text{ate})$$

$$P(\text{cheese} \mid \text{mouse}, \text{ate})$$

$$P(\text{the} \mid \text{cheese}, \text{mouse}, \text{ate})$$

$$P(\text{the} \mid \text{cheese}, \text{mouse}, \text{ate}, \text{the})$$

Autoregressive Language Models

$$p(\text{the, mouse, ate, the, cheese}) = p(\text{the})\, p(\text{mouse} \mid \text{the})\, p(\text{ate} \mid \text{the, mouse})\, p(\text{the} \mid \text{the, mouse, ate})\, p(\text{cheese} \mid \text{the, mouse, ate, the}).$$

next word prediction

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \mid x_{1:i-1})$$

classification

distribution of next word

Next Word

Context

P(word | context)

Autoregressive Language Models

$$
\begin{aligned}
p(\text{the, mouse, ate, the, cheese}) = \; & p(\text{the}) \\
& p(\text{mouse} \mid \text{the}) \\
& p(\text{ate} \mid \text{the, mouse}) \\
& p(\text{the} \mid \text{the, mouse, ate}) \\
& p(\text{cheese} \mid \text{the, mouse, ate, the}).
\end{aligned}
$$

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \mid x_{1:i-1})$$

Learning a language model is to learn these conditional probabilities, for any language sequence

Autoregressive Language Models

$$
\begin{aligned}
p(\text{the, mouse, ate, the, cheese}) = \; & p(\text{the}) \\
& p(\text{mouse} \mid \text{the}) \\
& p(\text{ate} \mid \text{the, mouse}) \\
& p(\text{the} \mid \text{the, mouse, ate}) \\
& p(\text{cheese} \mid \text{the, mouse, ate, the}).
\end{aligned}
$$

parameters

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \mid x_{1:i-1})$$

Given a dataset, how to find these probabilities?

loss? $\arg\max_{\theta} \log P_{\theta}(x)$

Autoregressive Language Models

$$
\begin{aligned}
p(\text{the, mouse, ate, the, cheese}) = \; & p(\text{the}) \\
& p(\text{mouse} \mid \text{the}) \\
& p(\text{ate} \mid \text{the, mouse}) \\
& p(\text{the} \mid \text{the, mouse, ate}) \\
& p(\text{cheese} \mid \text{the, mouse, ate, the}).
\end{aligned}
$$

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \mid x_{1:i-1})$$

Given a dataset, how to find these probabilities?

Maximum Likelihood Estimation

Count-based Language Models

Count the frequency and divide

$$p(x_i \mid x_{1:i-1}) = \frac{c(x_{1:i})}{c(x_{1:i-1})}$$

P(ate | the mouse)

c(the mouse ate)

c(the mouse)

the mouse ate | 3 times

the mouse ran | 2 times

other

the mouse x | 0 times

3 + 2 = 5

$\frac{3}{5}$

P(ate | the mouse)

P(not ate / not ran | the mouse) = 0

max 3 · log P(ate | the mouse) + 2 × log P(ran | the mouse)

P(ate | the mouse) + P(ran | the mouse) = 1

$$\max\ 3 \cdot \log P(\text{cate} \mid \text{the mouse}) + 2 \cdot \log(1 - P(\text{cate} \mid \text{the mouse}))$$

$$P(\text{cate} \mid \text{the mouse})$$

$$\frac{d}{dP} = \frac{3}{P(\text{cate} \mid \text{the mouse})} + \frac{-2}{1 - P(\text{cate} \mid \text{the mouse})}$$

$$= 0 \Rightarrow P(\text{cate} \mid \text{the mouse}) = \frac{3}{5}$$

Count-based Language Models

Count the frequency and divide

$$p(x_i \mid x_{1:i-1}) = \frac{c(x_{1:i})}{c(x_{1:i-1})}$$

There are infinite number of parameters for language

Count-based Language Models

Count the frequency and divide

$$p(x_i \mid x_{1:i-1}) = \frac{c(x_{1:i})}{c(x_{1:i-1})}$$

There are infinite number of parameters for language

We may see long sequences only once, counting becomes meaningless

n-gram Language Models

n-gram Language Models

Next token probability only depends on the previous n-1 words

n-gram Language Models

Next token probability only depends on the previous n-1 words

Unigram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i)$$

$P(x_i)$ finite

n-gram Language Models

Next token probability only depends on the previous n-1 words

Unigram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i)$$

Each token is independent

n-gram Language Models

Next token probability only depends on the previous n-1 words

Unigram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i)$$

Each token is independent

Bigram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \mid x_{i-1})$$

n-gram Language Models

Next token probability only depends on the previous n-1 words

Unigram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i)$$

Each token is independent

Bigram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \,|\, x_{i-1})$$

Generally for n-gram LM:

$$p(x_1, x_2, \ldots, x_I) = \prod_{i=1}^{I} p(x_i \,|\, x_{i-n+1:i-1})$$

Parameter Estimation for n-gram LM

Count-based:

$$p(x_i \mid x_{i-n+1:i-1}) = \frac{c(x_{i-n+1:i})}{c(x_{i-n+1:i-1})}$$

Parameter Estimation for n-gram LM

Count-based:

$$p(x_i \mid x_{i-n+1:i-1}) = \frac{c(x_{i-n+1:i})}{c(x_{i-n+1:i-1})}$$

finance

Number of parameters decreases, but flexibility decreases as well

" I am from the major computer science,
Now I am year 3 in HKUST. ...
... I like the sea. this semester.
I am taken classes

Parameter Estimation for n-gram LM

Count-based:

$NN(x_{i-n+1:i-1}, x_i)$

$$p(x_i \mid x_{i-n+1:i-1}) = \frac{c(x_{i-n+1:i})}{c(x_{i-n+1:i-1})}$$

Number of parameters decreases, but flexibility decreases as well

Traditionally, we directly compute this probability, but neural language models use neural networks to compute the probability

Neural Language Models

Neural Language Models

Neural language models are typically autoregressive next word prediction

Neural Language Models

Neural language models are typically autoregressive

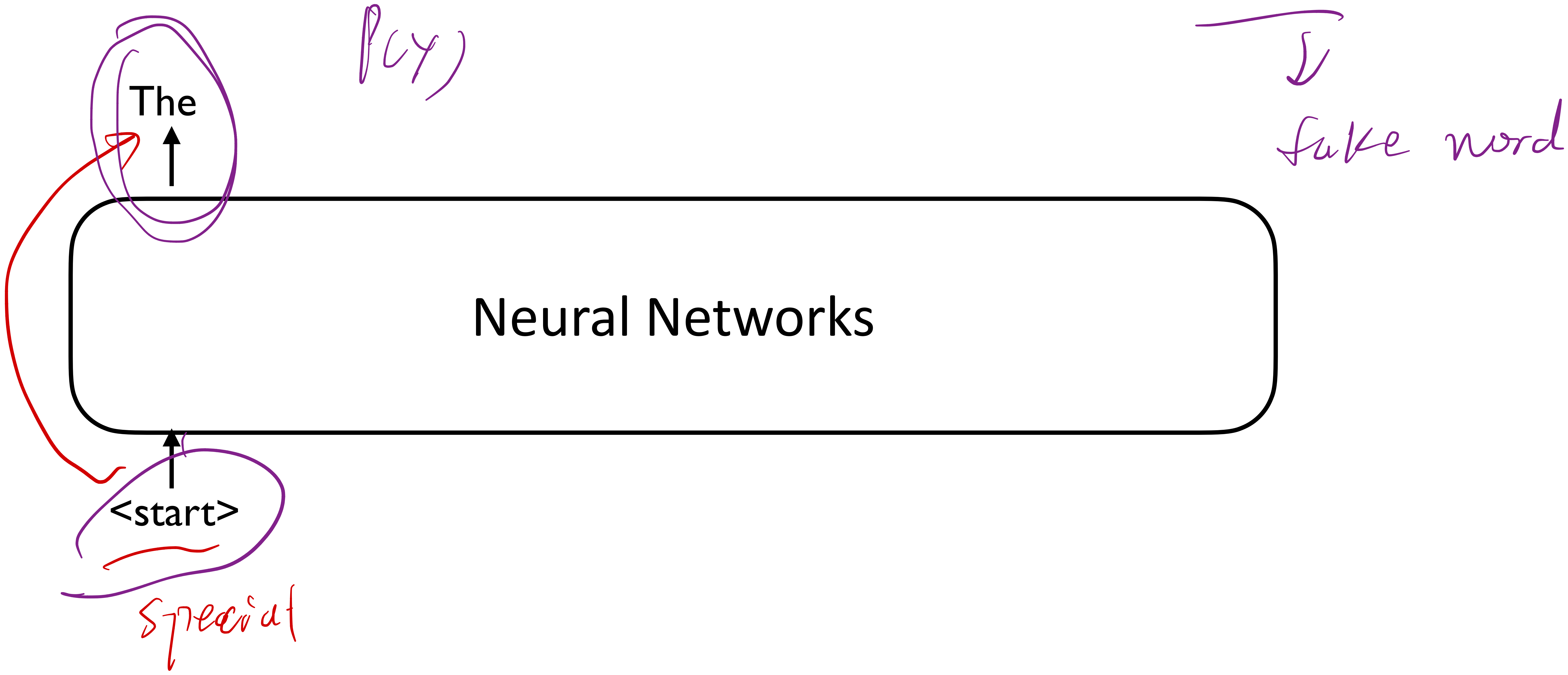
Data: "The mouse ate the cheese ."

Neural Language Models

Neural language models are typically autoregressive

Data: "The mouse ate the cheese ."

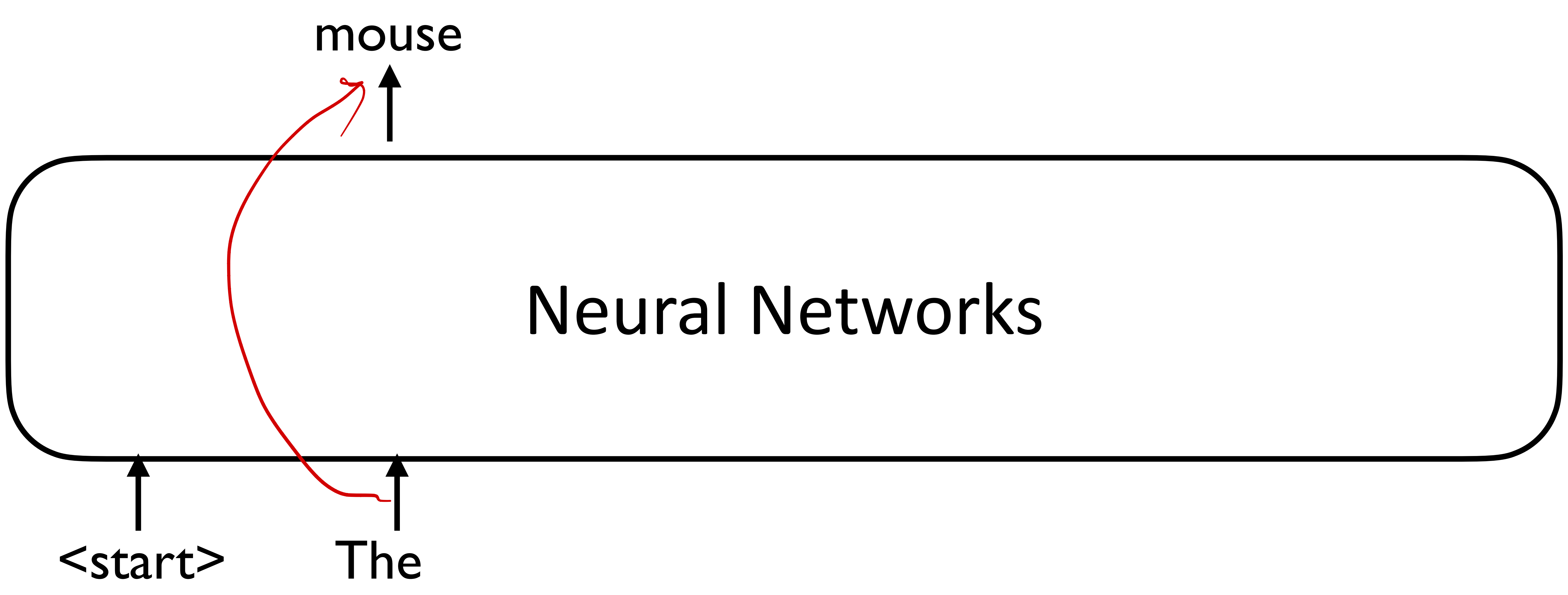
P(the) = P(The | <start>)



Neural Language Models

Neural language models are typically autoregressive

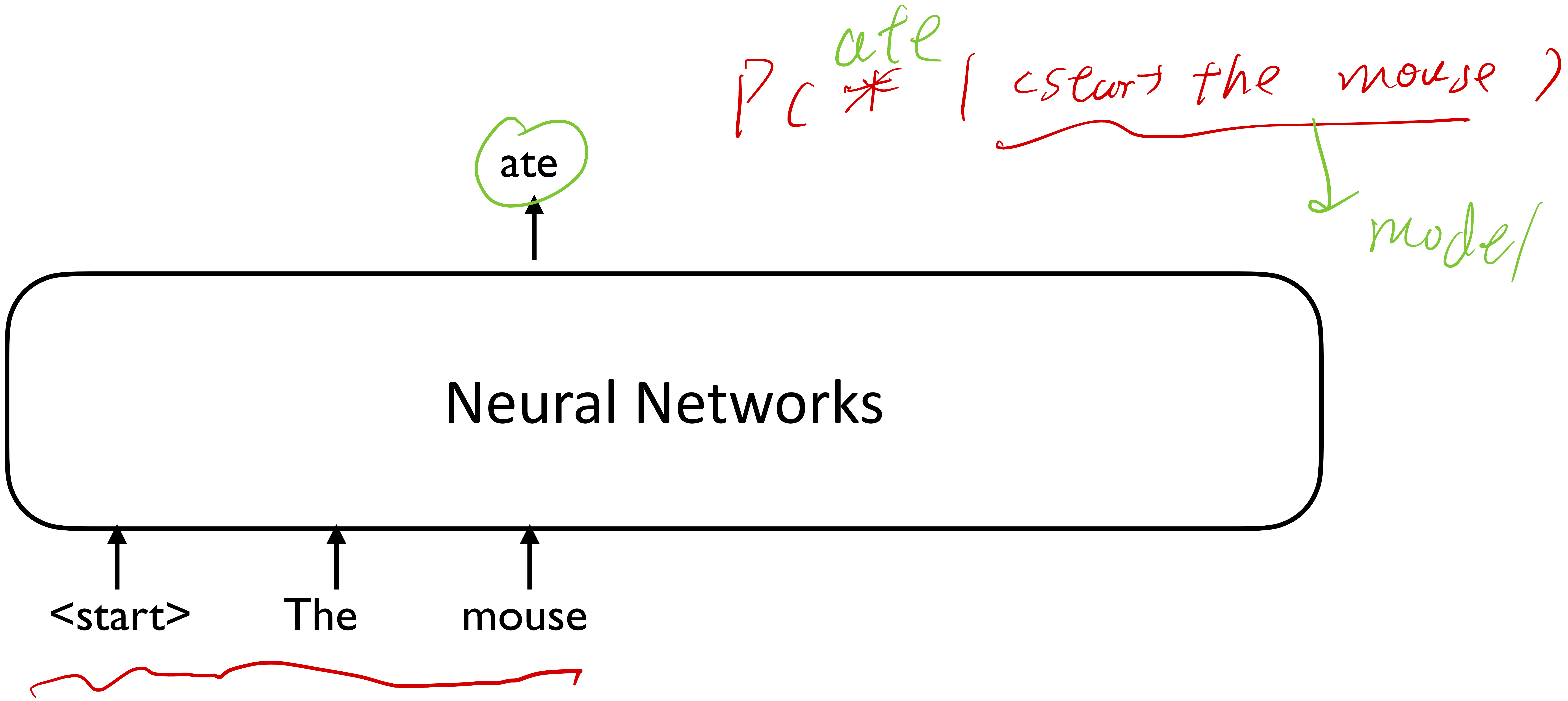
Data: "The mouse ate the cheese ."



Neural Language Models

Neural language models are typically autoregressive

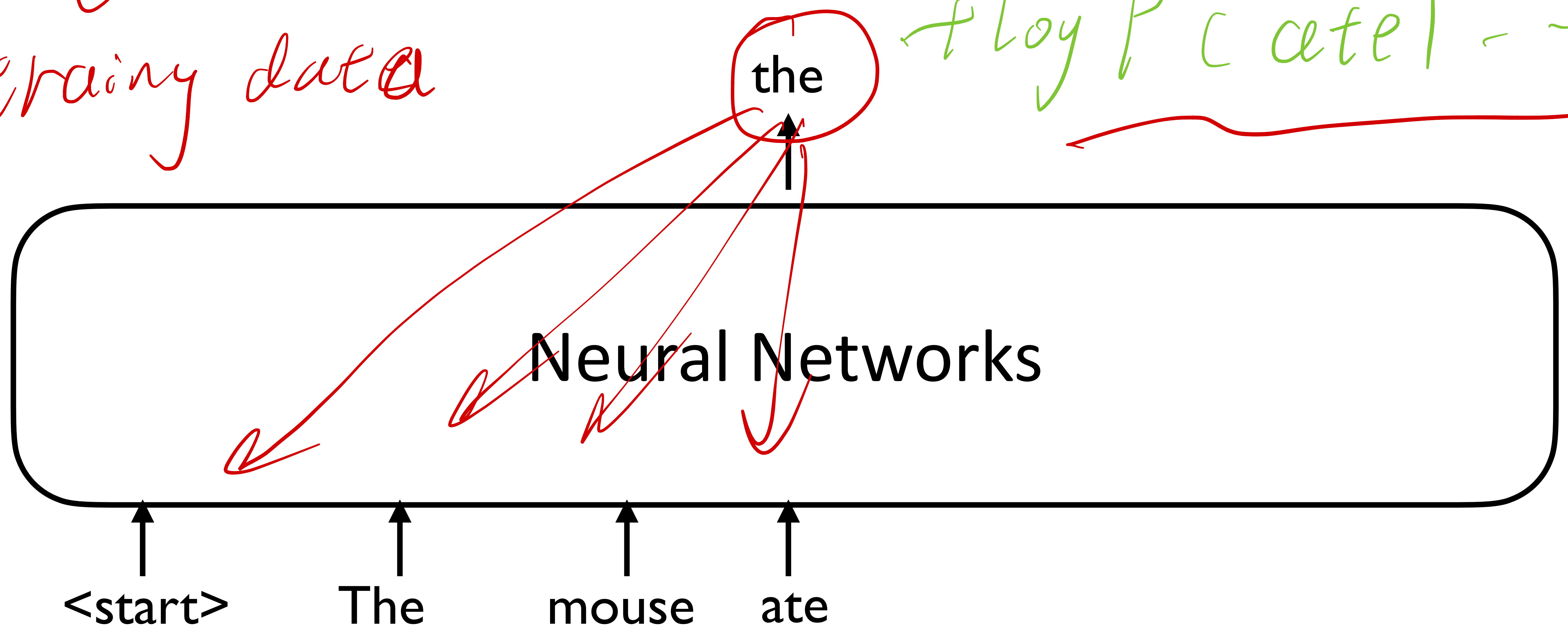
Data: "The mouse ate the cheese ."



Neural Language Models

Neural language models are typically autoregressive

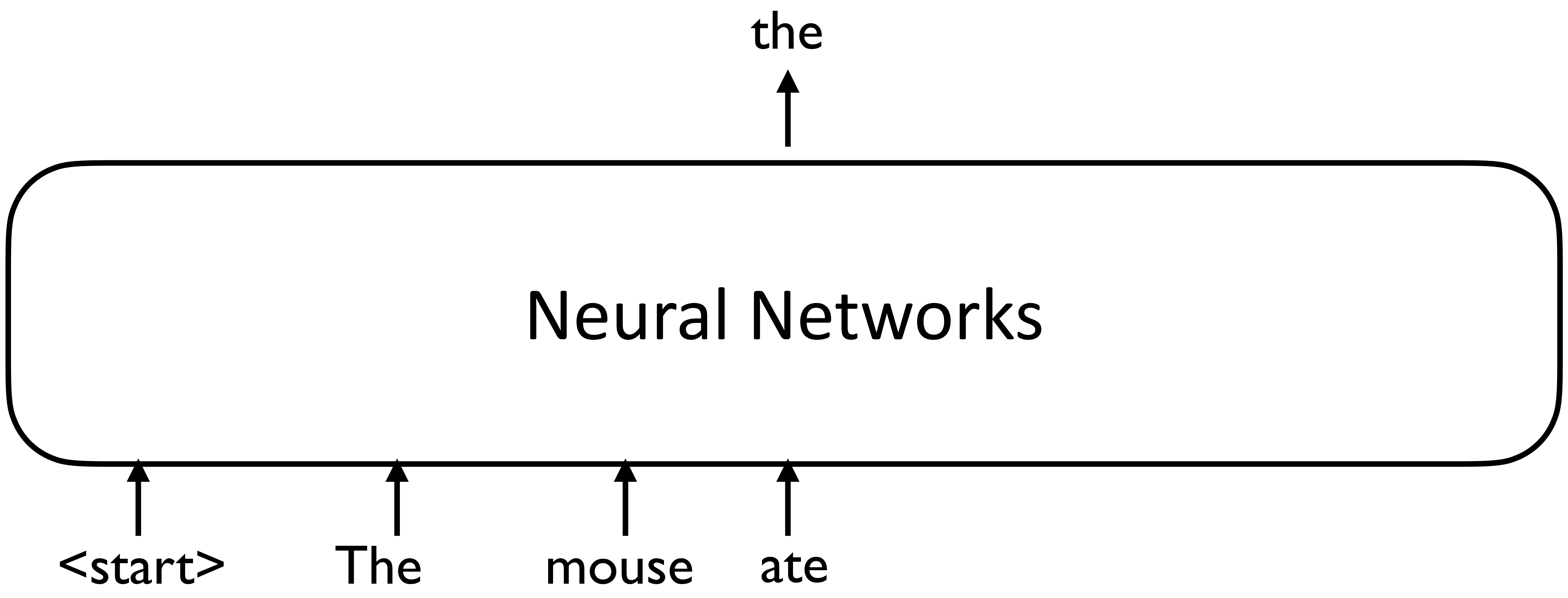
Data: “The mouse ate the cheese .”



Neural Language Models

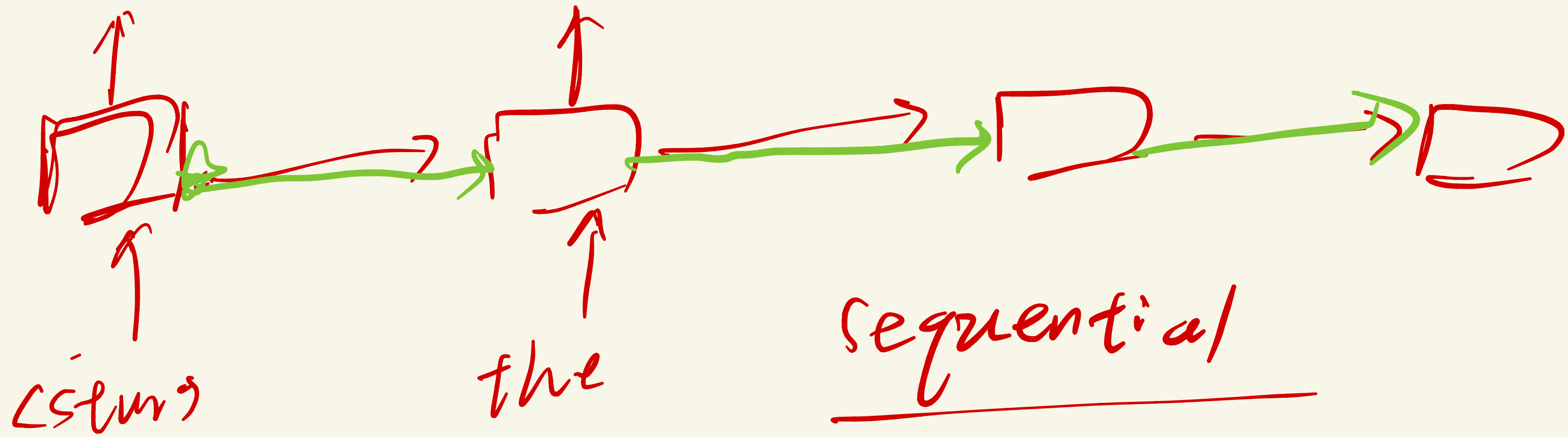
Neural language models are typically autoregressive

Data: "The mouse ate the cheese ."



We can compute the loss on every token in parallel

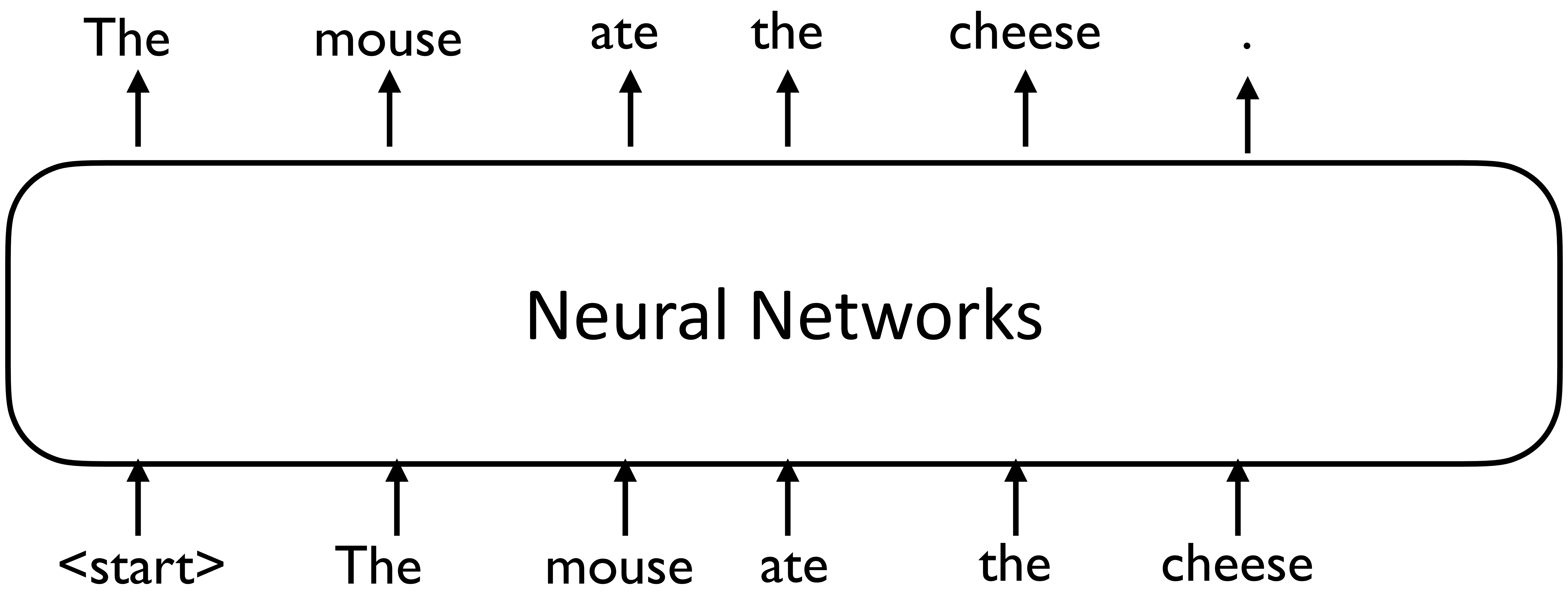
RNN



Neural Language Models

Neural language models are typically autoregressive

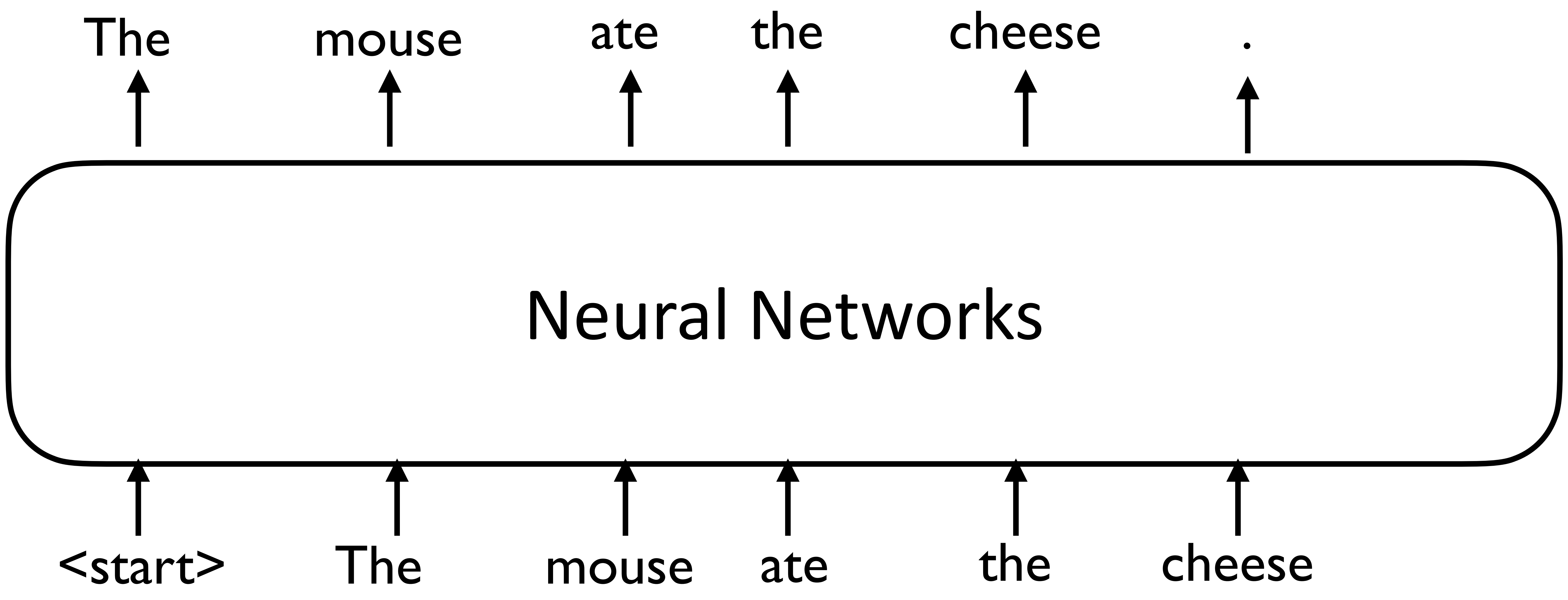
Data: "The mouse ate the cheese ."



Neural Language Models

Neural language models are typically autoregressive

Data: "The mouse ate the cheese ."



Each prediction only sees the inputs on its left

Neural Language Models

Neural Language Models

Are language models generative models? $P(x)$

Neural Language Models

Are language models generative models?

Can we compute p(x) given x? Can we sample new x?

$\prod P(\text{next word} \mid \text{context})$

Neural Language Models

Are language models generative models? ✅

Can we compute $p(x)$ given x? Can we sample new x? ✅

Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

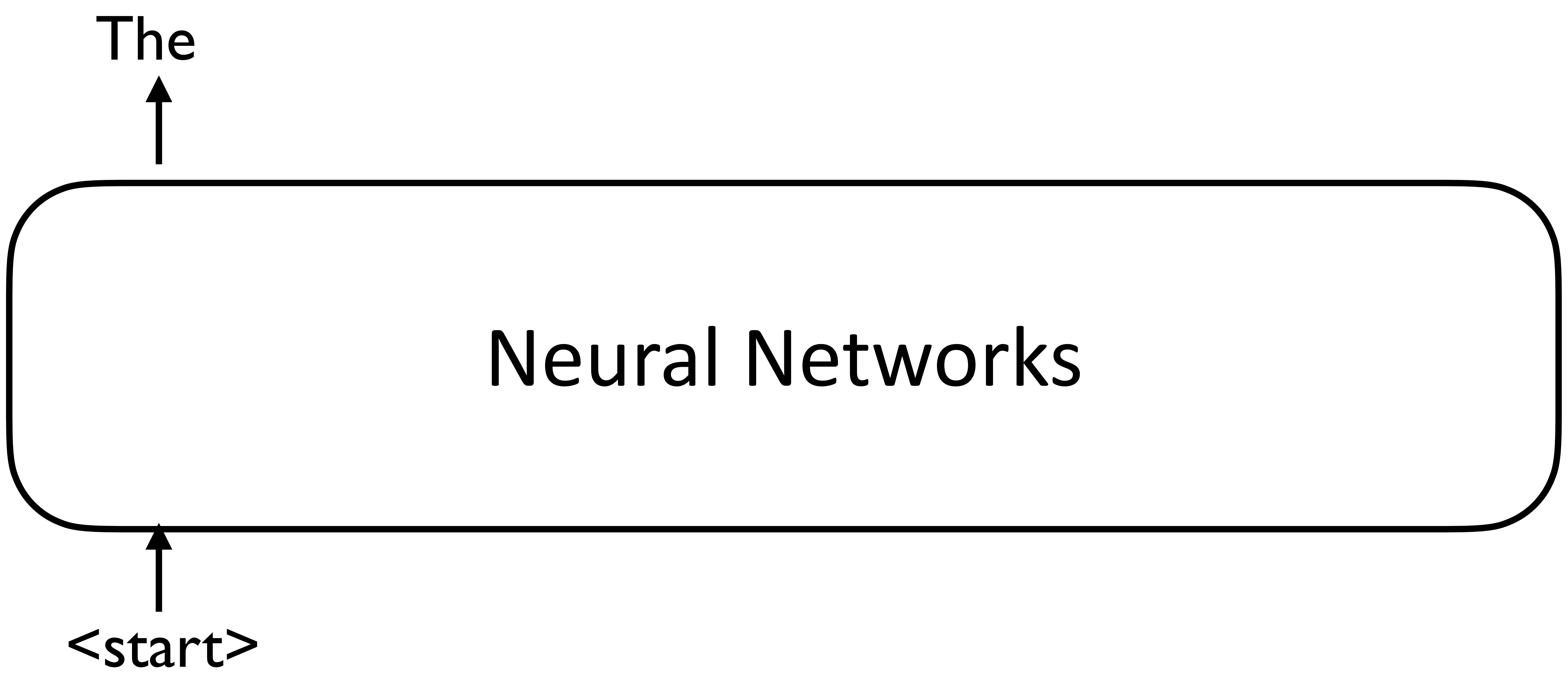
At inference time, to generate:

Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

At inference time, to generate:

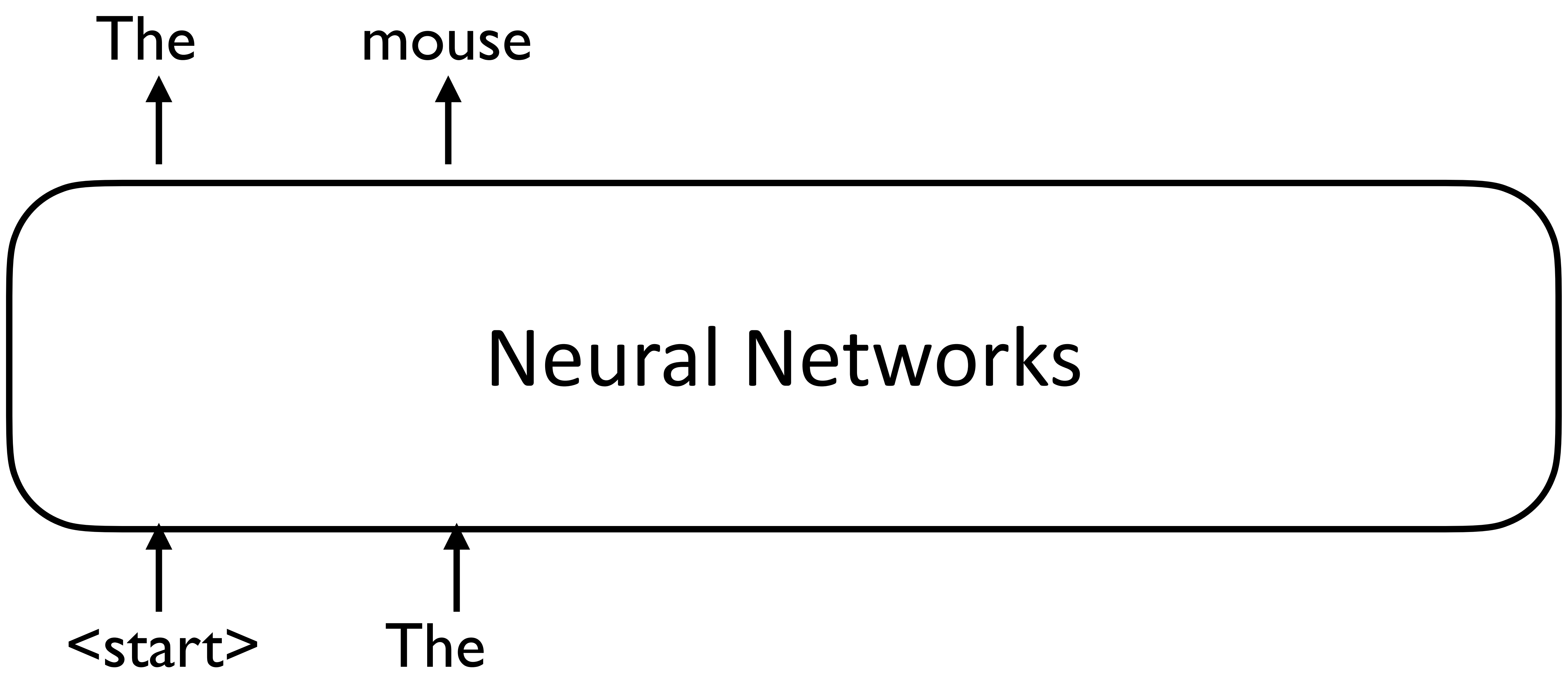


Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

At inference time, to generate:

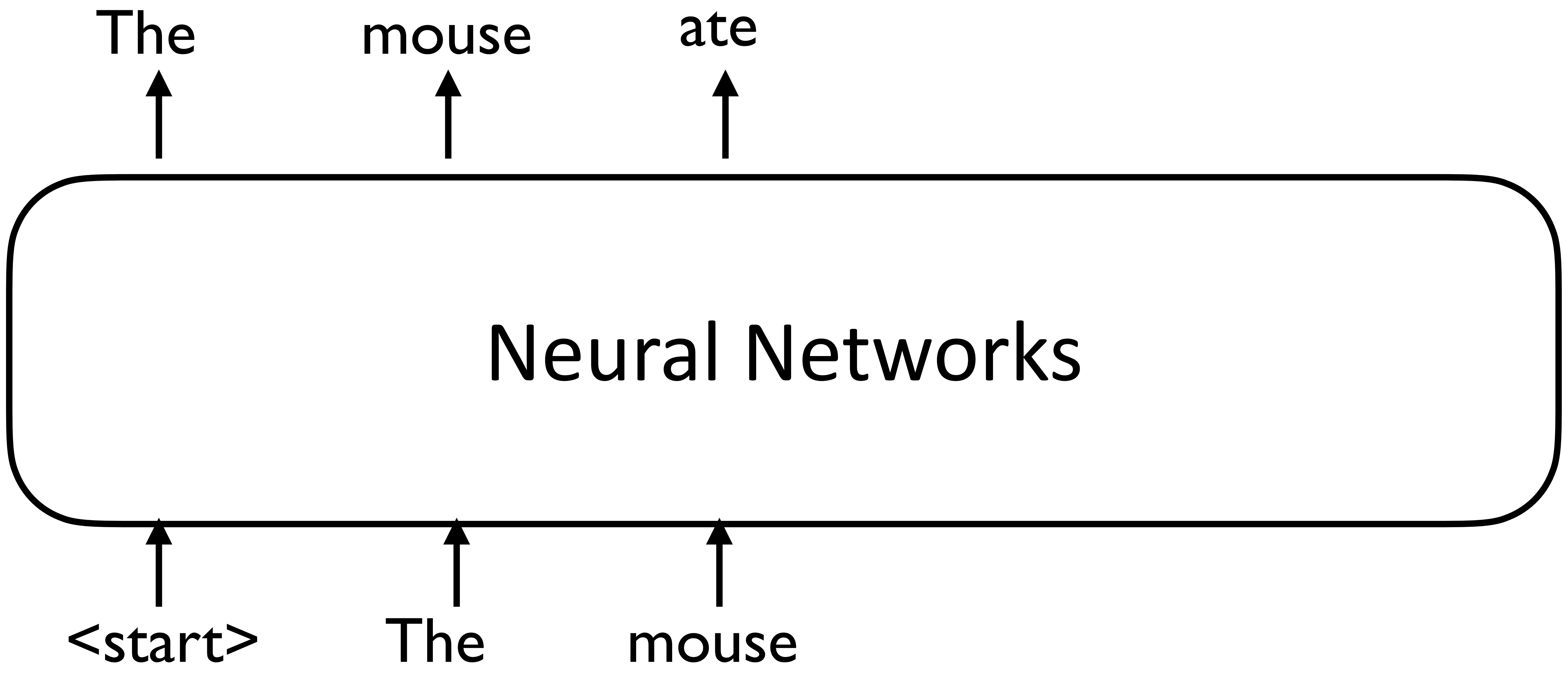


Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

At inference time, to generate:

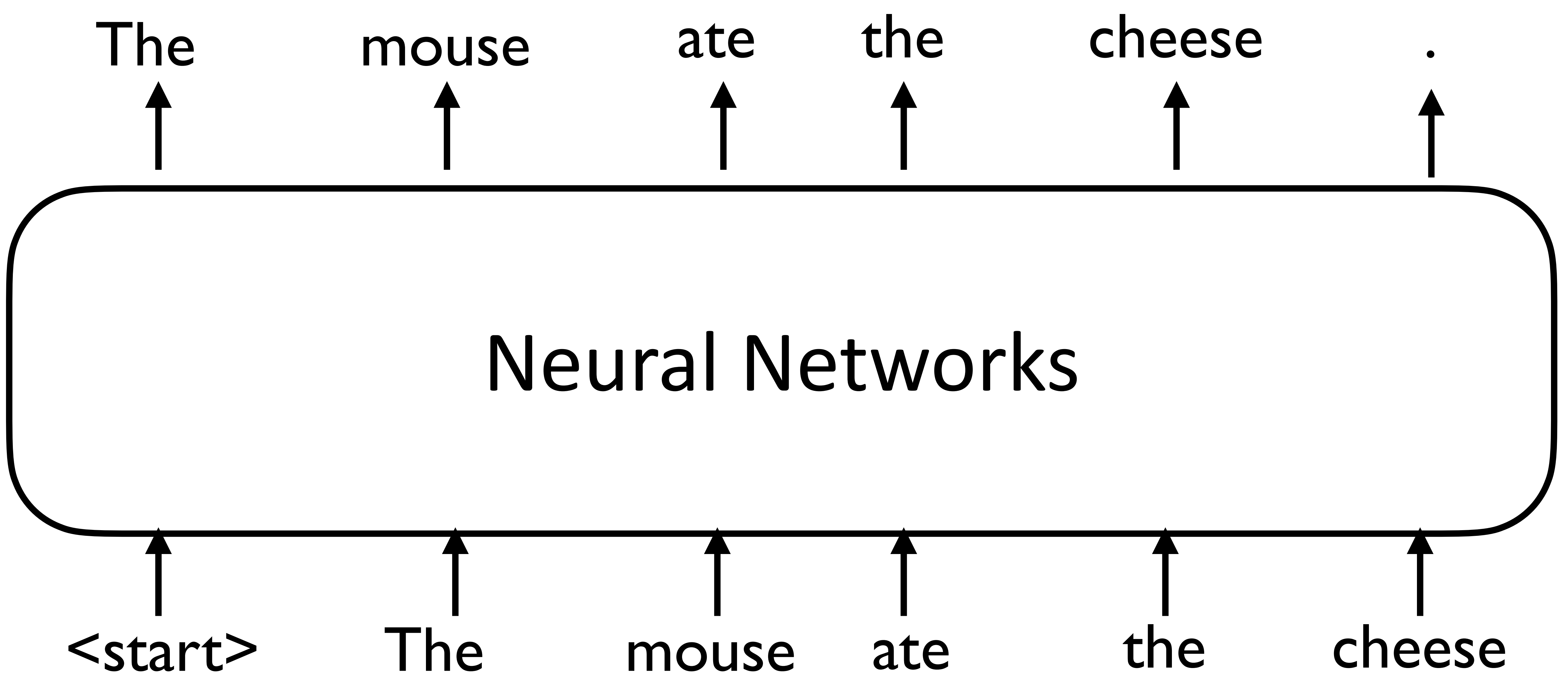


Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

At inference time, to generate:

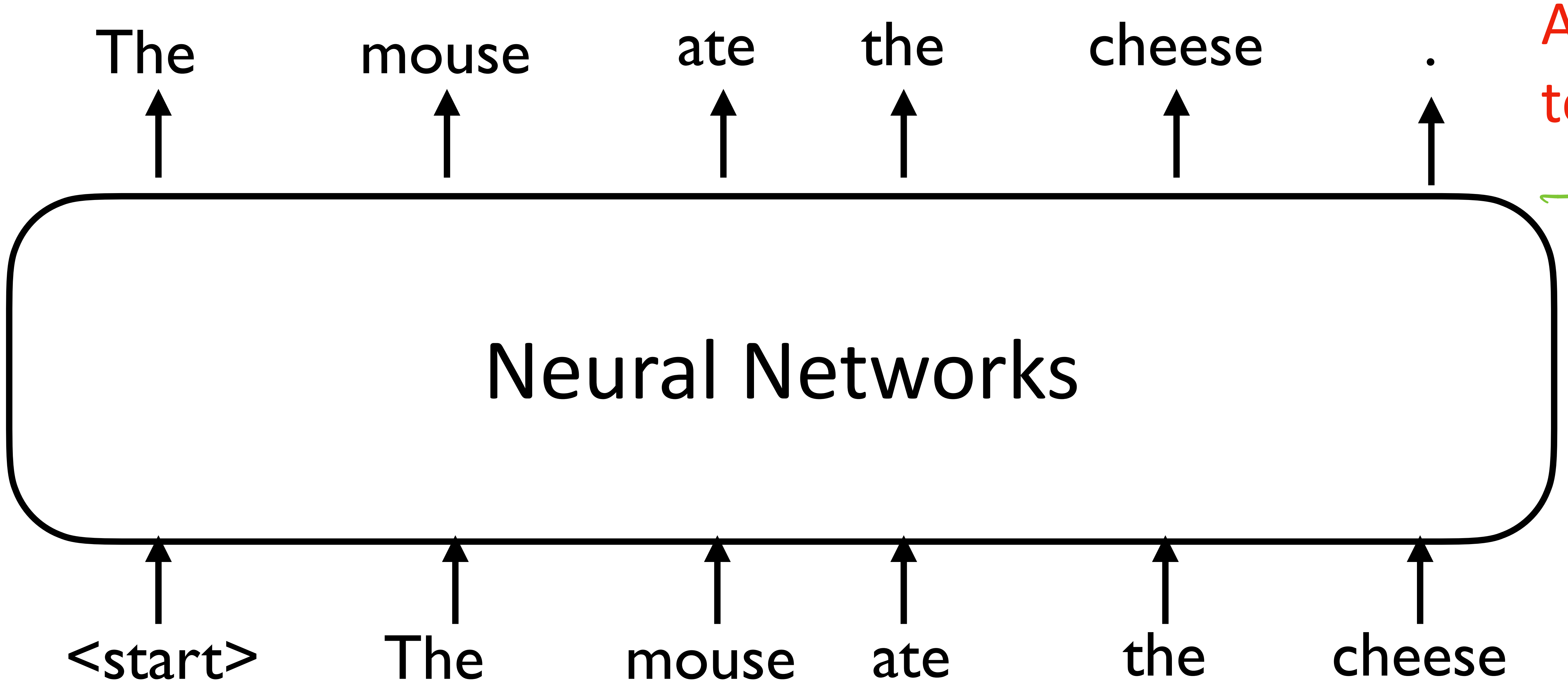


Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

At inference time, to generate:



Autoregressive generation has to generate token by token

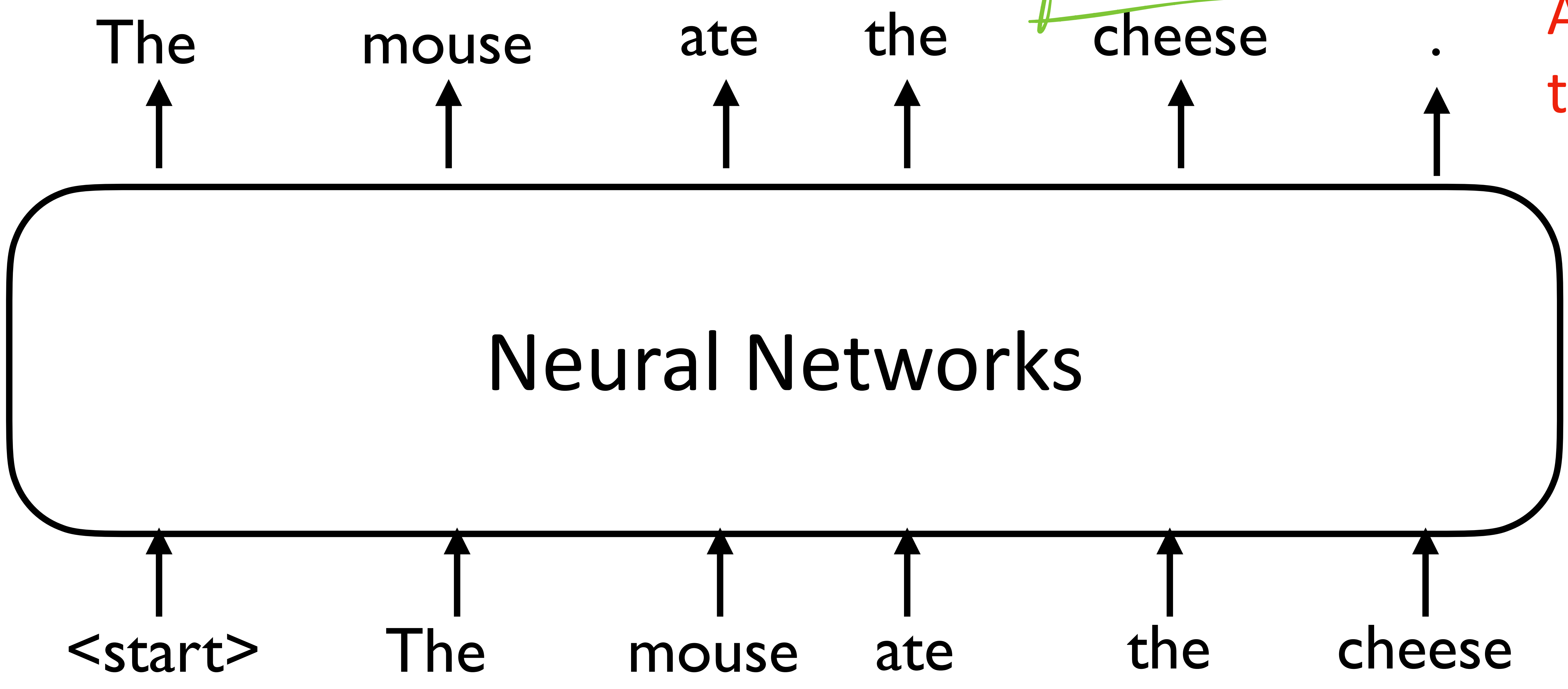
Neural Language Models

Are language models generative models? ✅

Can we compute p(x) given x? Can we sample new x? ✅

At inference time, to generate: O(n)

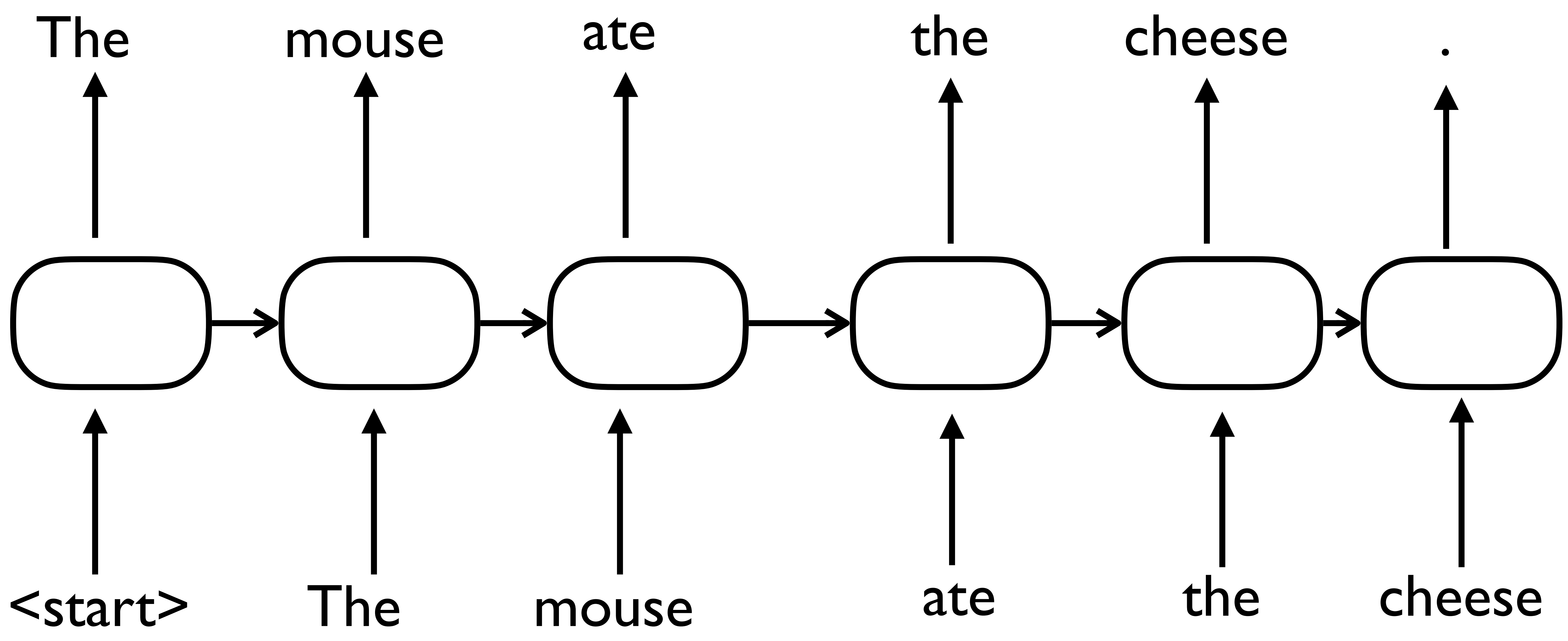
sequential



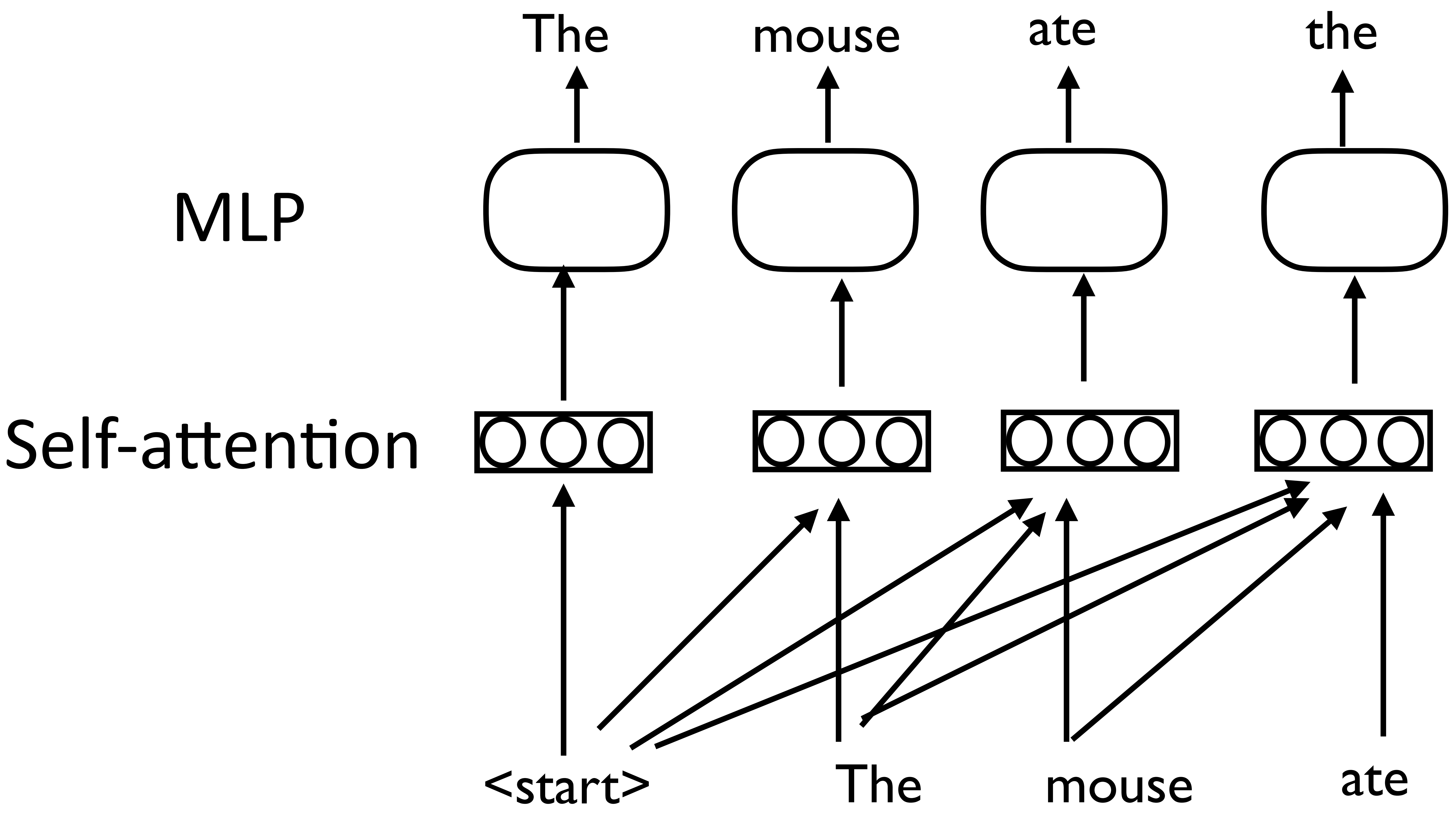
Autoregressive generation has to generate token by token

Cann't parallelize, efficiency of autoregressive decoding is still an important research topic

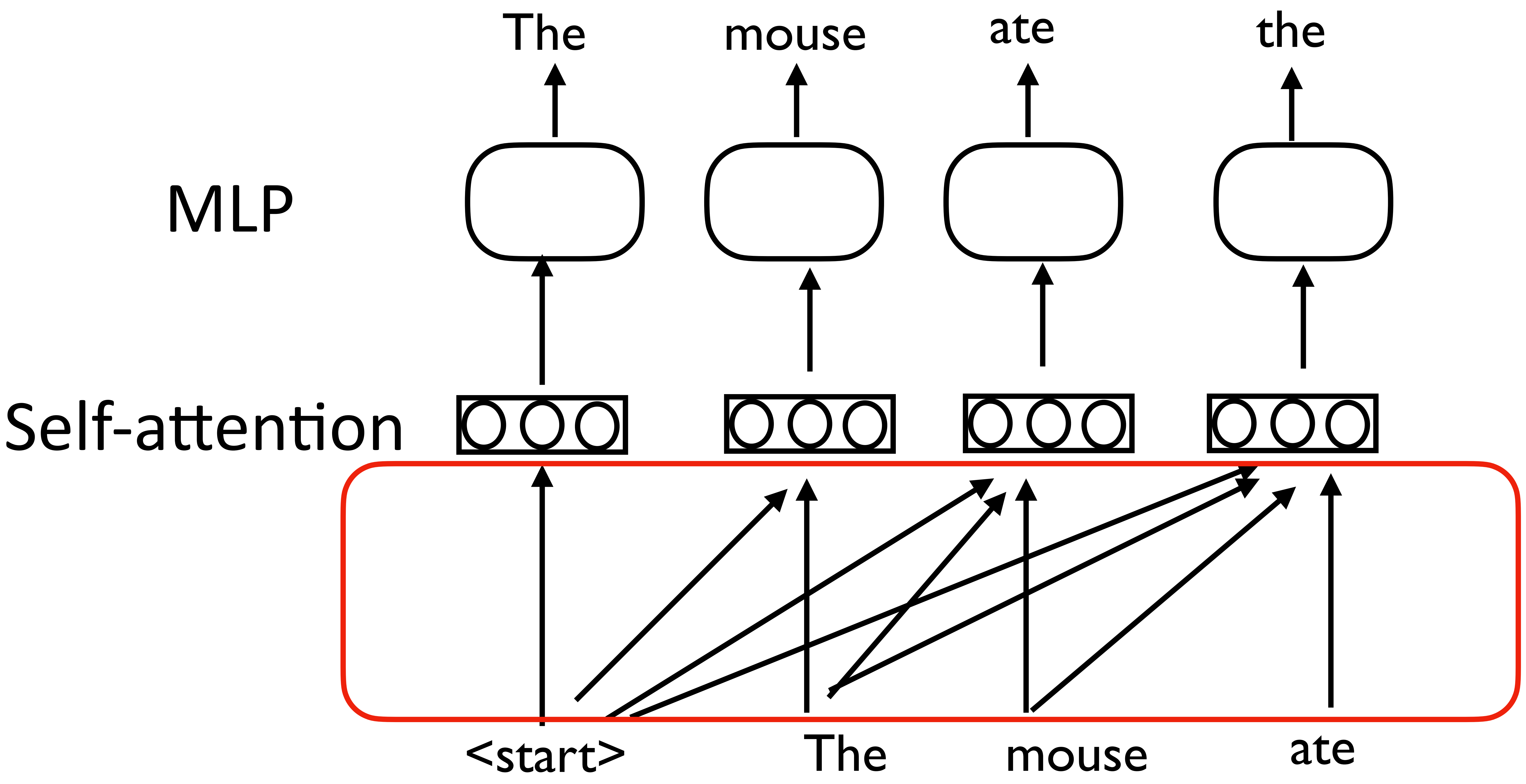
RNN Language Models



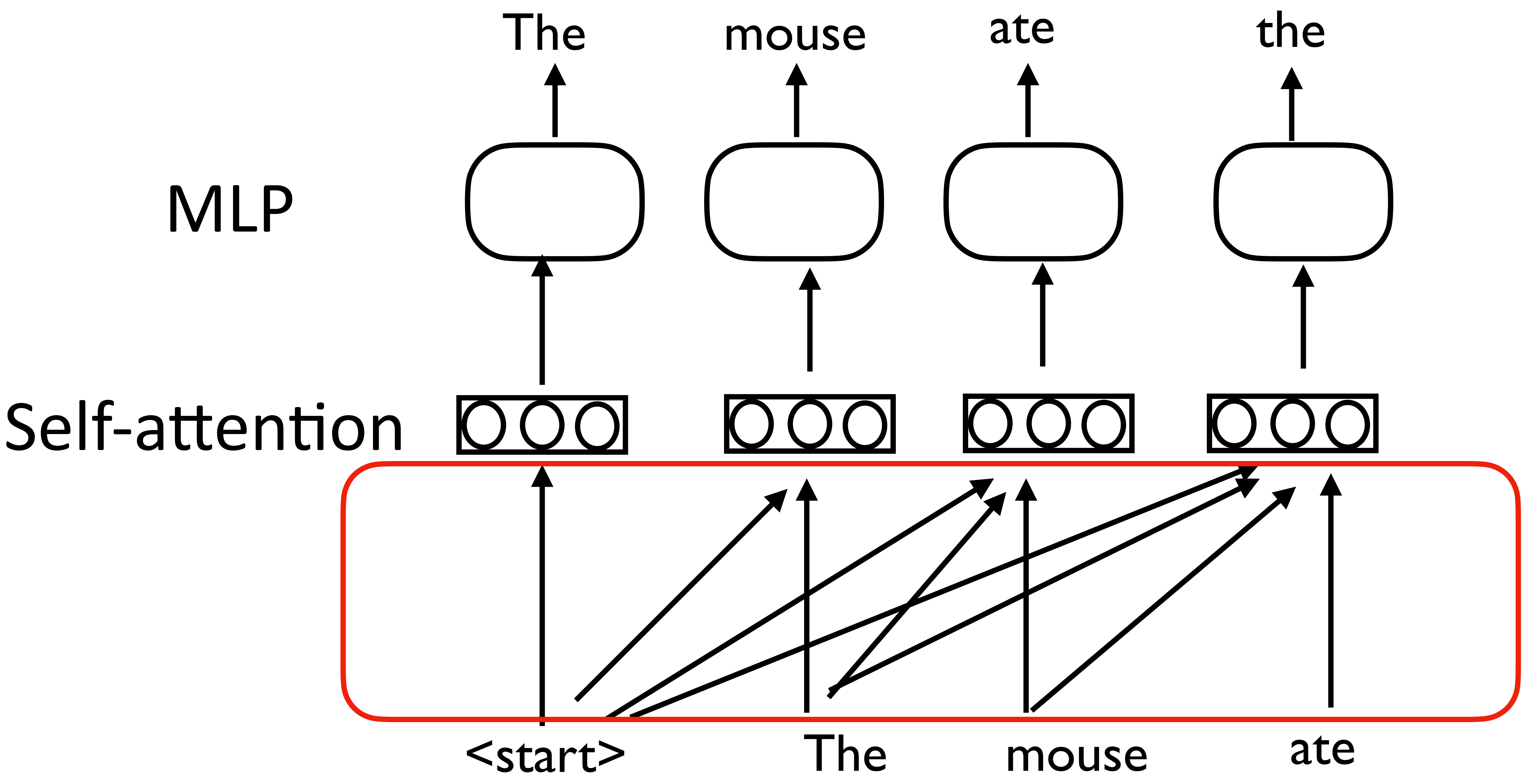
Transformer Language Models



Transformer Language Models



Transformer Language Models



Self-attention only attends to the tokens on the left (masked attention)

Neural Language Models

Language model is the fundamental block to model language distribution p(x)

Neural Language Models

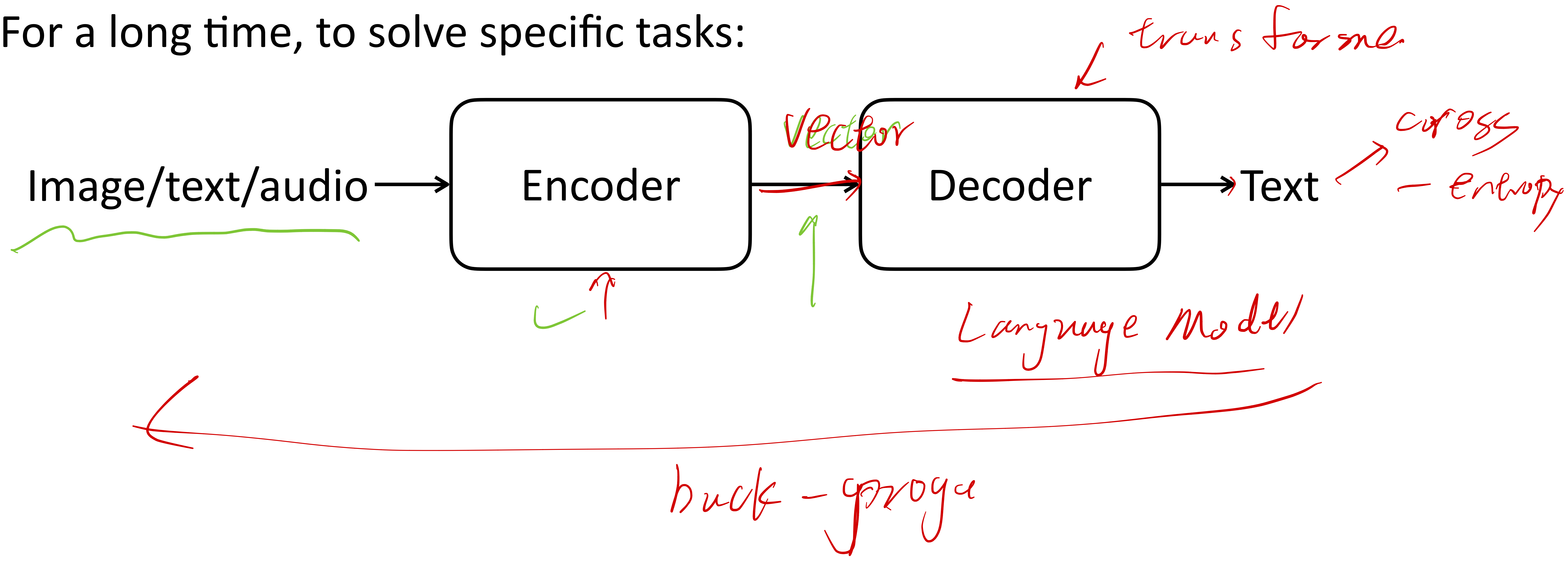
Language model is the fundamental block to model language distribution p(x)

For a long time, to solve specific tasks:

Neural Language Models

Language model is the fundamental block to model language distribution p(x)

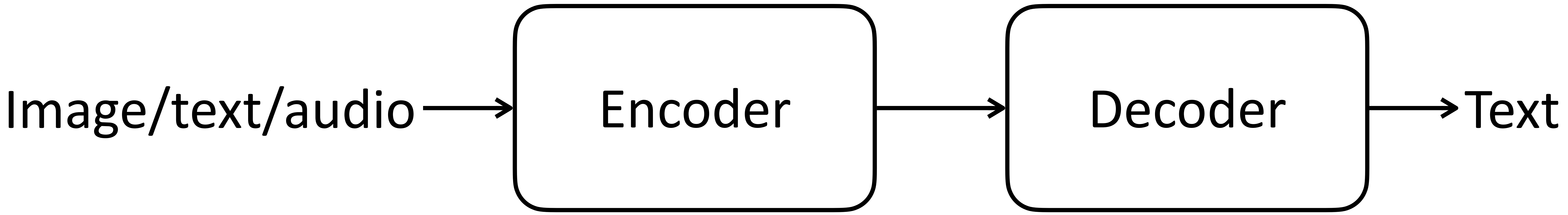
For a long time, to solve specific tasks:



Neural Language Models

Language model is the fundamental block to model language distribution p(x)

For a long time, to solve specific tasks:

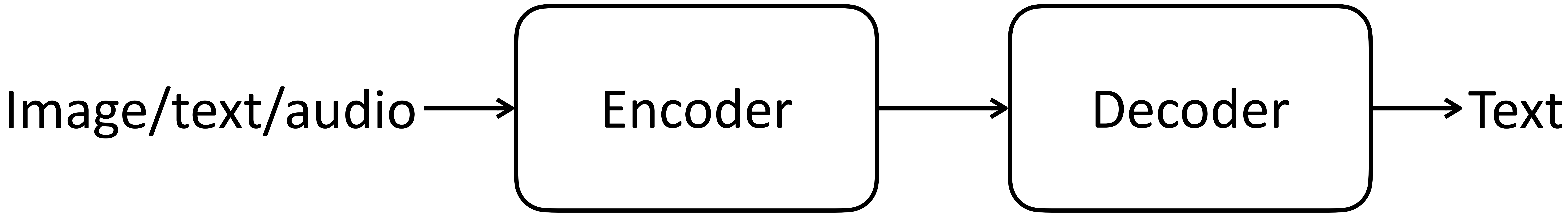


When we have a better arch/training for LM, we can have a better decoder

Neural Language Models

Language model is the fundamental block to model language distribution p(x)

For a long time, to solve specific tasks:



When we have a better arch/training for LM, we can have a better decoder

Not long ago, some people think purely language models is useless because it does not directly address tasks, and LM performance may not transfer to downstream tasks

translation, summarization, QA

Is Next Token Prediction Useful?

Ok, language modeling can be used as pretraining, but is a language model itself useful for some tasks directly?

Is Next Token Prediction Useful?

Ok, language modeling can be used as pretraining, but is a language model itself useful for some tasks directly?

In the late 1980s the Hong Kong Government anticipated a strong demand for university graduates to fuel an economy increasingly based on services. Sir Sze-Yuen Chung and Sir Edward Youde, the then Governor of Hong Kong, conceived the idea of another university in addition to the pre-existing two universities, The University of Hong Kong and The Chinese University of Hong Kong.
Planning for the "Third University", named The Hong Kong University of Science and Technology later, began in 1986. Construction began at the Kohima Camp site in Tai Po Tsai on the Clear Water Bay Peninsula. The site was earmarked for the construction of a new [ ]

Is Next Token Prediction Useful?

Ok, language modeling can be used as pretraining, but is a language model itself useful for some tasks directly?

In the late 1980s the Hong Kong Government anticipated a strong demand for university graduates to fuel an economy increasingly based on services. Sir Sze-Yuen Chung and Sir Edward Youde, the then Governor of Hong Kong, conceived the idea of another university in addition to the pre-existing two universities, The University of Hong Kong and The Chinese University of Hong Kong.
Planning for the "Third University", named The Hong Kong University of Science and Technology later, began in 1986. Construction began at the Kohima Camp site in Tai Po Tsai on the Clear Water Bay Peninsula. The site was earmarked for the construction of a new [ ]

Completion

Is Next Token Prediction Useful?

Ok, language modeling can be used as pretraining, but is a language model itself useful for some tasks directly?

In the late 1980s the Hong Kong Government anticipated a strong demand for university graduates to fuel an economy increasingly based on services. Sir Sze-Yuen Chung and Sir Edward Youde, the then Governor of Hong Kong, conceived the idea of another university in addition to the pre-existing two universities, The University of Hong Kong and The Chinese University of Hong Kong.
Planning for the "Third University", named The Hong Kong University of Science and Technology later, began in 1986. Construction began at the Kohima Camp site in Tai Po Tsai on the Clear Water Bay Peninsula. The site was earmarked for the construction of a new [ ]

Completion

This task seems useless in practice

Language Models are Zero-Shot Learners

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Radford et al. Language Models are Unsupervised Multitask Learners. 2019 (GPT2)

GPT-2

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

GPT-2

Next token prediction can unify many tasks

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

GPT-2

Next token prediction can unify many tasks

Machine translation:

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

GPT-2

Next token prediction can unify many tasks

Machine translation:

Chinese: 今天是学期的最后一天。 English:

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

GPT-2

Next token prediction can unify many tasks

Machine translation:

Chinese: 今天是学期的最后一天。
English:

Question answering:

Q: What is the capital of the United States?
A:

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

GPT-2

Next token prediction can unify many tasks

Machine translation:

Chinese: 今天是学期的最后一天。
English:

Completion is very general

Question answering:

Q: What is the capital of the United States?
A:

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

GPT-2

Next token prediction can unify many tasks

Machine translation:

> Chinese: 今天是学期的最后一天。
> English:

Question answering:

> Q: What is the capital of the United States?
> A:

Completion is very general

This was an early form of prompting, that is widely discussed today

Radford et al. Language Models are Unsupervised Multitask Learners. 2018.

Language Models Are Few-Shot Learners

Brown et al. Language models are few-shot learners. 2020

Language Models Are Few-Shot Learners

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:      ← task description
cheese =>                         ← prompt
```

<1B

175B

large language models

Brown et al. Language models are few-shot learners. 2020

Language Models Are Few-Shot Learners

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
cheese => ........               ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
sea otter => loutre de mer       ← example
cheese => ........               ← prompt
```


Brown et al. Language models are few-shot learners. 2020

Language Models Are Few-Shot Learners

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
cheese => ......................  ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
sea otter => loutre de mer       ← example
cheese => ......................  ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:          ← task description
sea otter => loutre de mer            ← examples
peppermint => menthe poivrée          ←
plush girafe => girafe peluche        ←
cheese => ......................       ← prompt
```

Brown et al. Language models are few-shot learners. 2020

Language Models Are Few-Shot Learners

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:      ← task description
2  cheese => ..................      ← prompt
```

One-shot

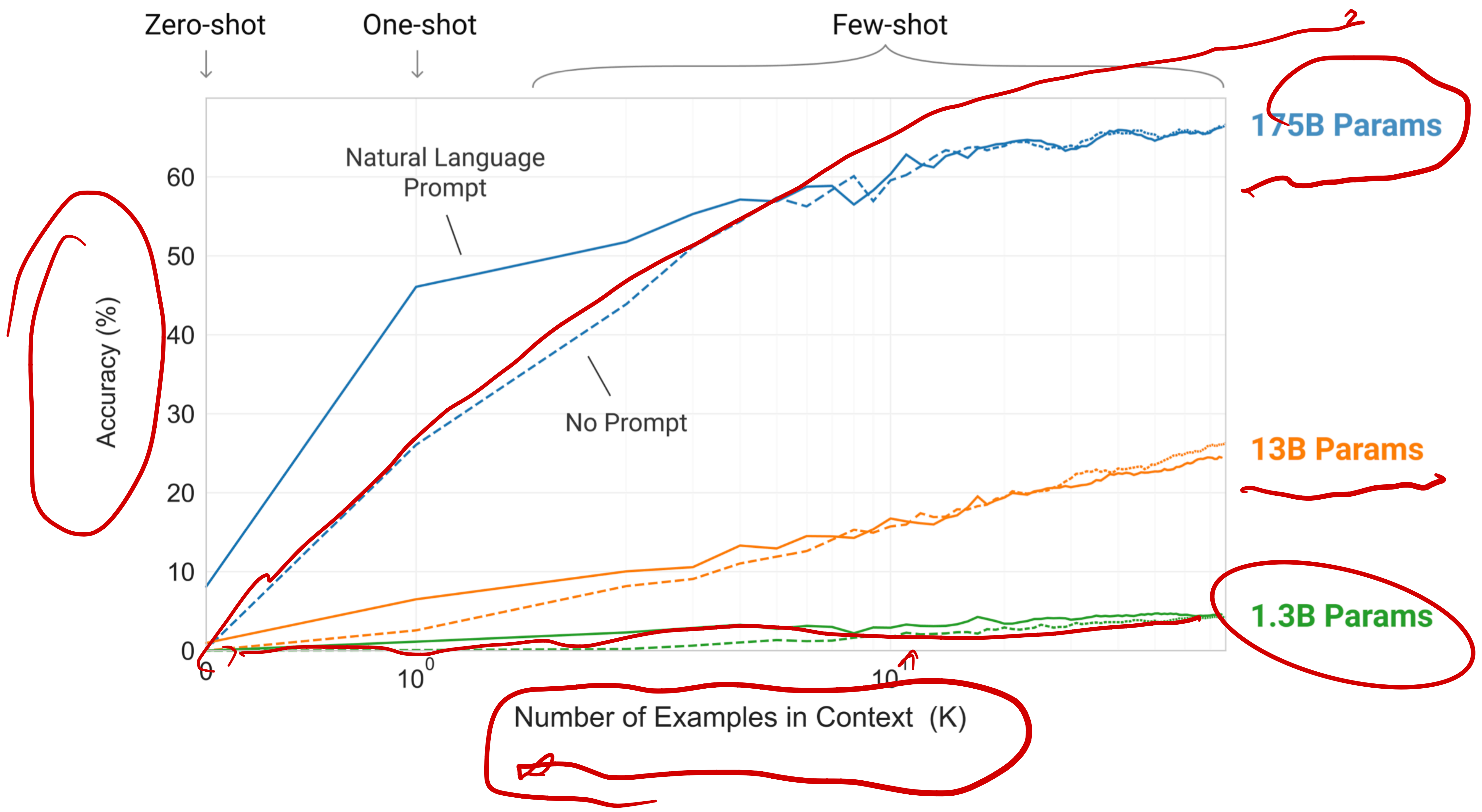
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:      ← task description
2  sea otter => loutre de mer        ← example
3  cheese => ..................      ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:            ← task description
2  sea otter => loutre de mer              ← examples
3  peppermint => menthe poivrée            ←
4  plush girafe => girafe peluche          ←
5  cheese => ..................            ← prompt
```



Brown et al. Language models are few-shot learners. 2020

Language Models Are Few-Shot Learners

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:     ← task description
2  cheese => .................      ← prompt
```

One-shot

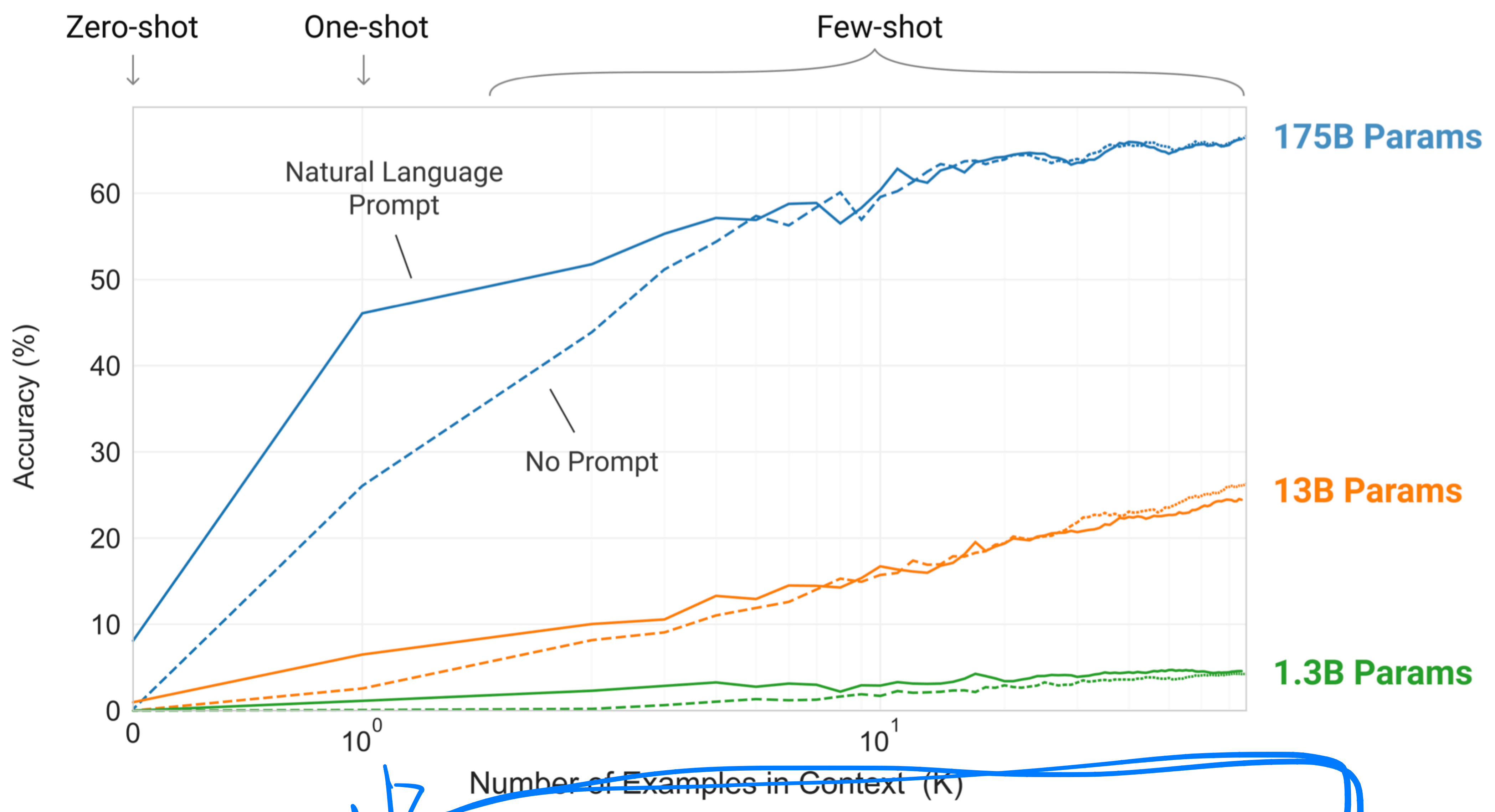
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:     ← task description
2  sea otter => loutre de mer       ← example
3  cheese => .................      ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:          ← task description
2  sea otter => loutre de mer            ← examples
3  peppermint => menthe poivrée          ←
4  plush girafe => girafe peluche        ←
5  cheese => .................           ← prompt
```



In-Context Learning

Brown et al. Language models are few-shot learners. 2020

Pretraining

Pretraining

Target Data B

Pretraining

Source Data A (maybe a different task)

Target Data B

Pretraining

language model

supervised

translation

Source Data A (maybe a different task)

Target Data B

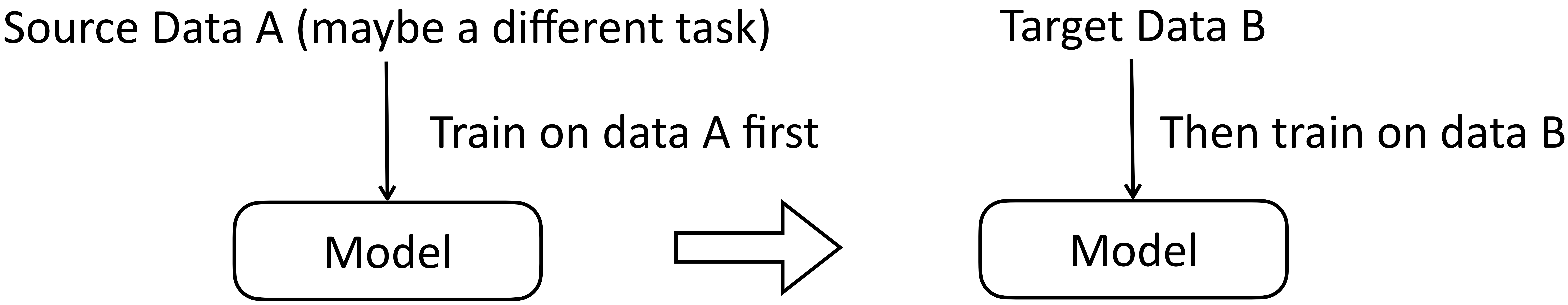
Train on data A first

Model

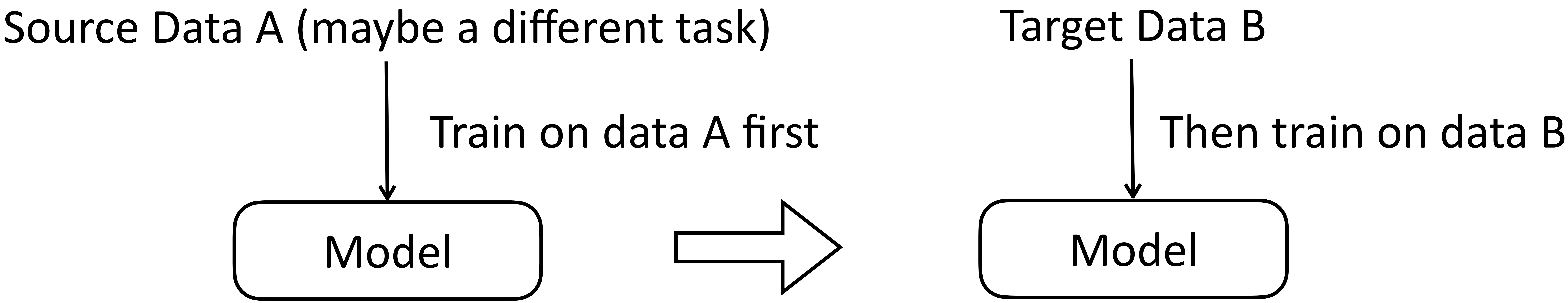
Unsupervised language modeling:

the mouse ate the cheese

Pretraining

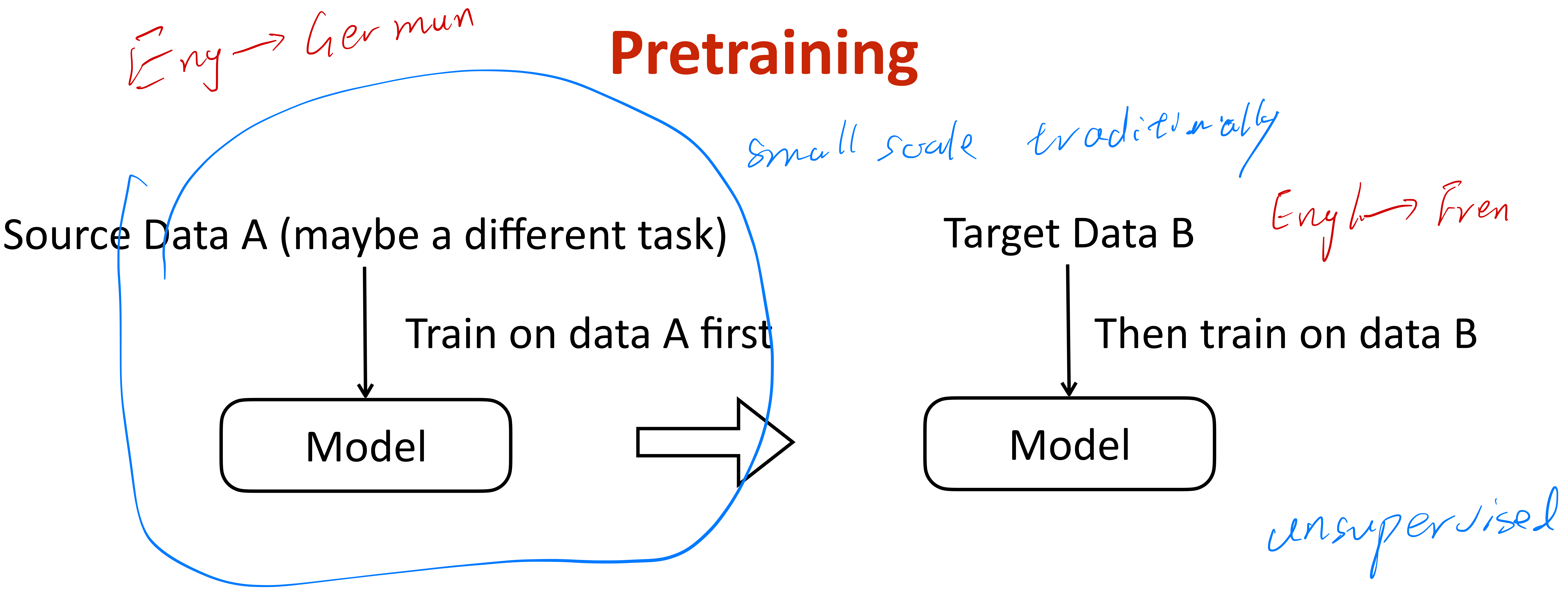


Pretraining



Classically, this is transfer Learning

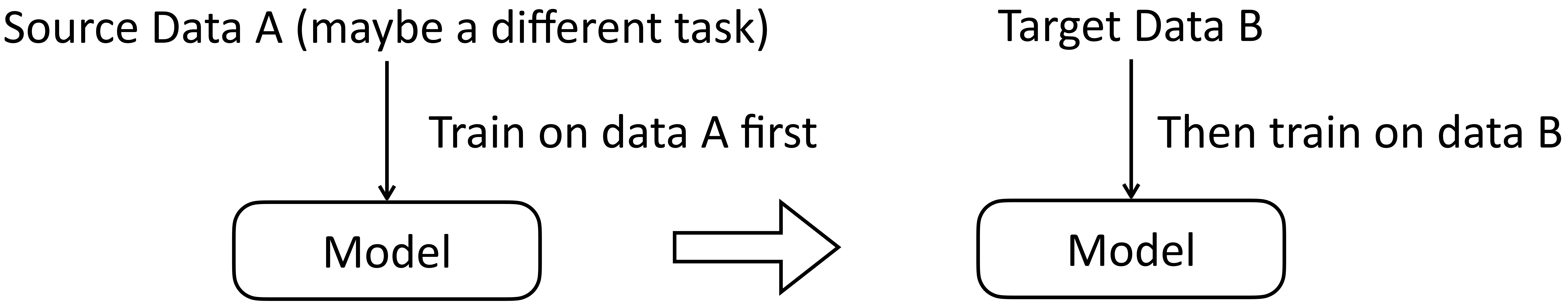
Pretraining



Classically, this is transfer Learning

It is now called pretraining because of the scale of A

Pretraining



For supervised training, data A is often limited

How can we find large-scale data A to train?

Prompt Breaks Task Boundaries

Raffle et al. Exploring the Limits of Transfer Learning. 2020

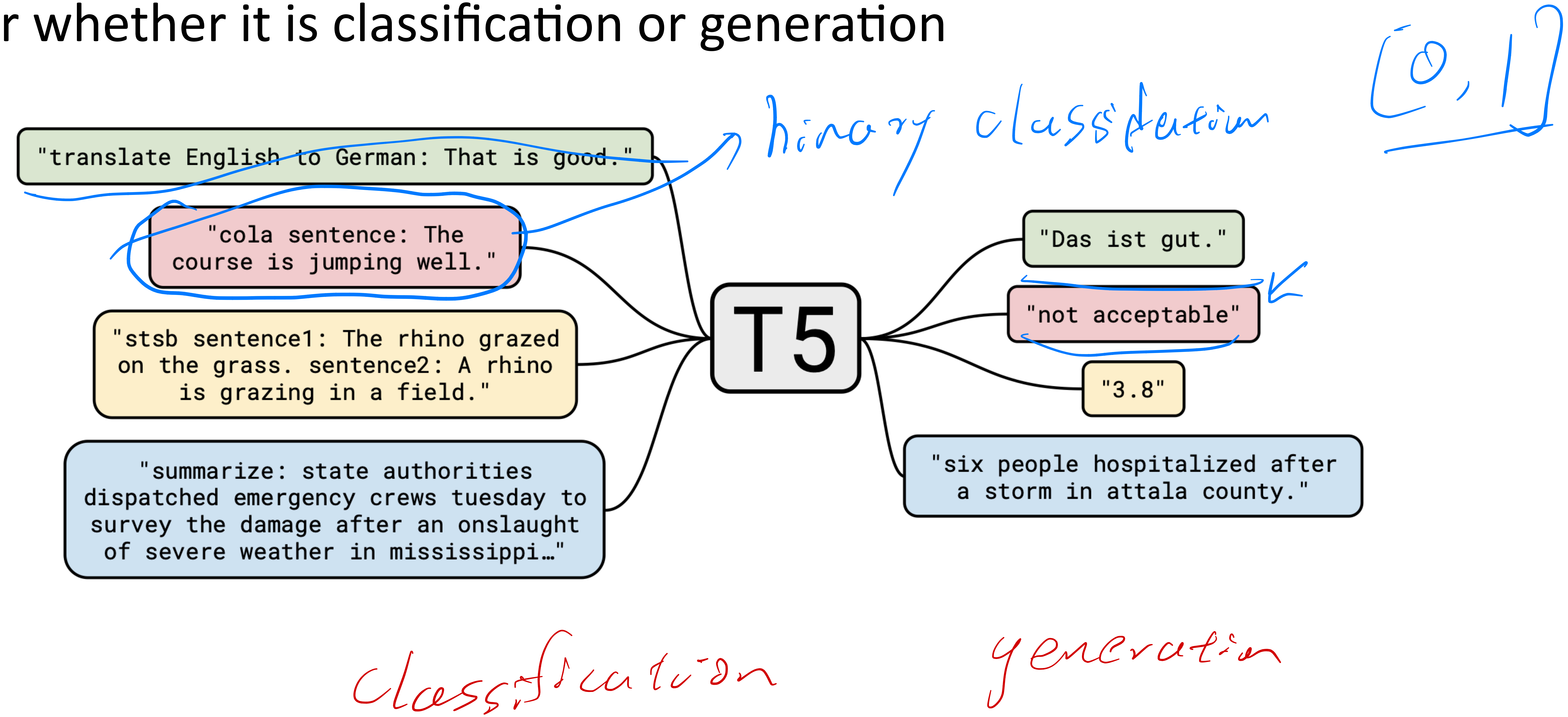
Prompt Breaks Task Boundaries

Almost all text tasks can be expressed with a unified format, no matter whether it is classification or generation

Raffle et al. Exploring the Limits of Transfer Learning. 2020

Prompt Breaks Task Boundaries

Almost all text tasks can be expressed with a unified format, no matter whether it is classification or generation



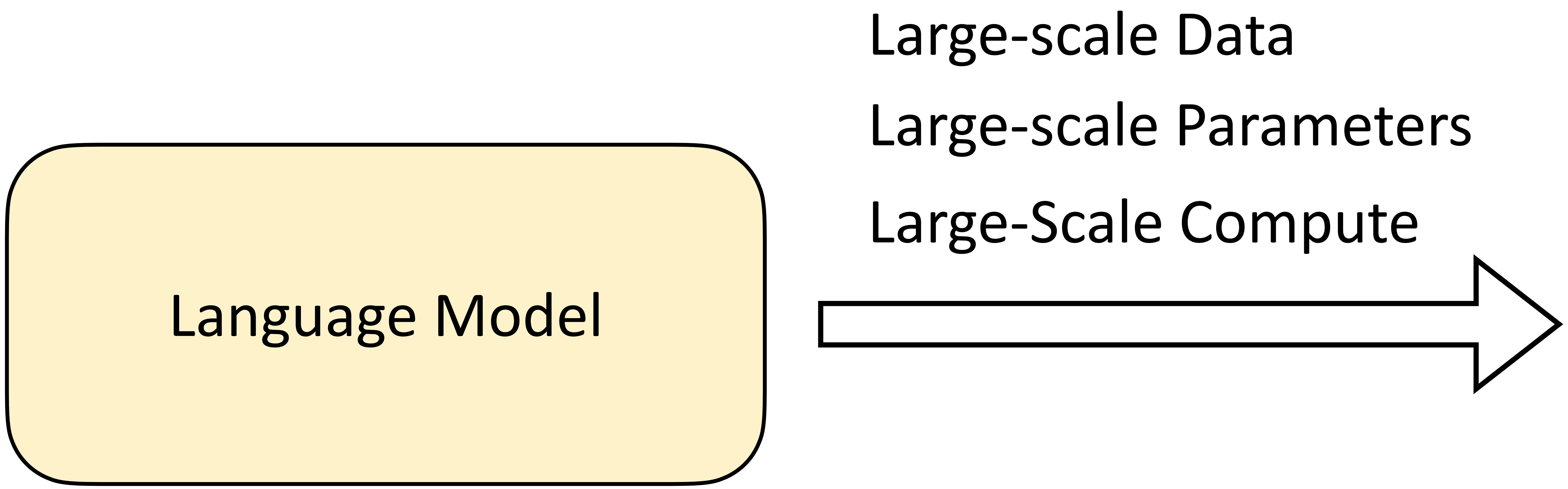
Raffle et al. Exploring the Limits of Transfer Learning. 2020

Large Language Models

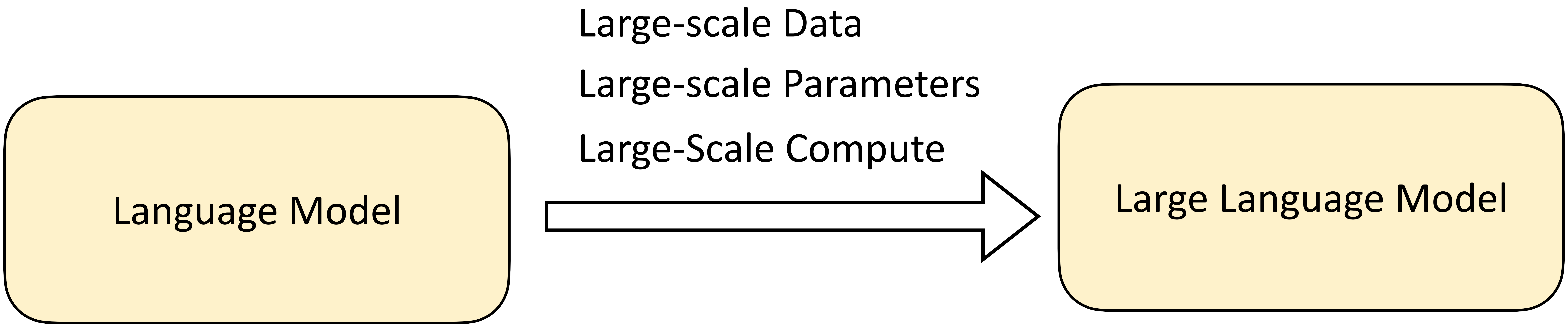
Large Language Models

Language Model

Large Language Models



Large Language Models



Thank You!