# Language: Representation Learning

EECE454 Intro. to Machine Learning Systems

Fall 2024

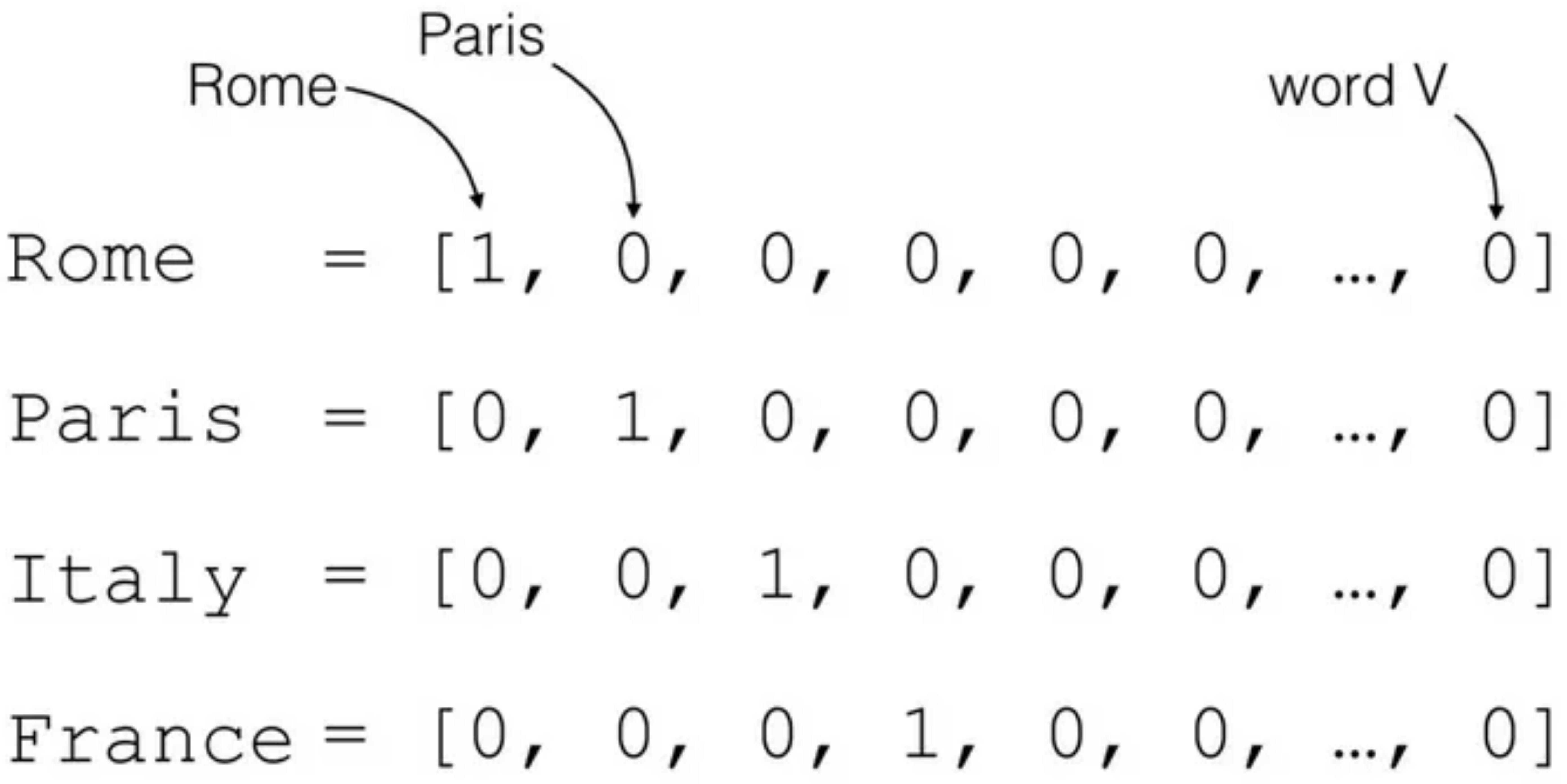
# Overview

- **Today.** Basic training of language representations
  - Word2Vec: Prediction
  - GloVe: Co-occurence
  - BERT: Masking
  - GPT: Next token prediction

# Word2Vec

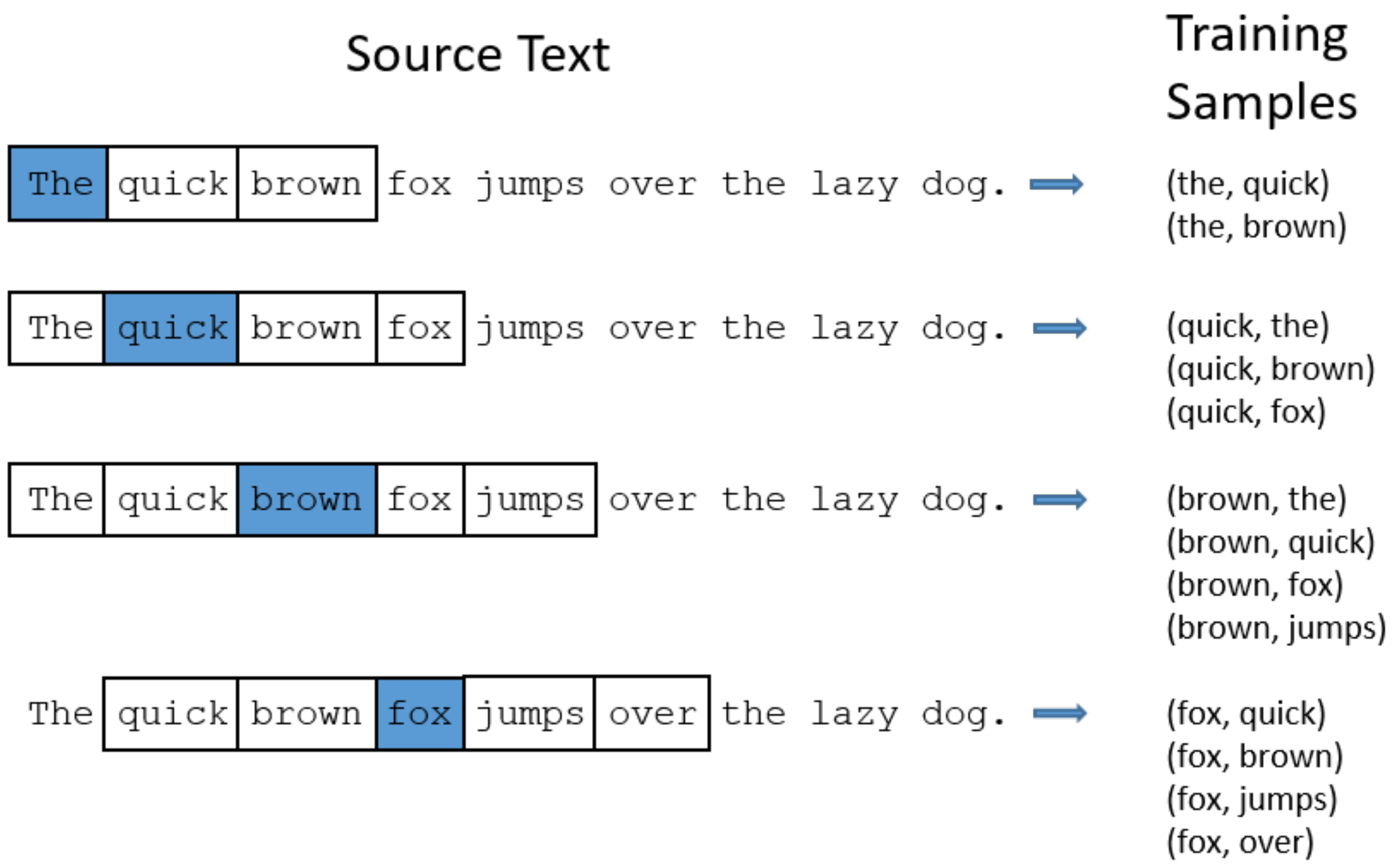
# Text representations

- **Goal.** Train a nice text embedding $f(\text{word}) = \text{vector}$
  - <u>Example</u>. One-hot encoding
    - Does not reflect any semantics & high-dimensional



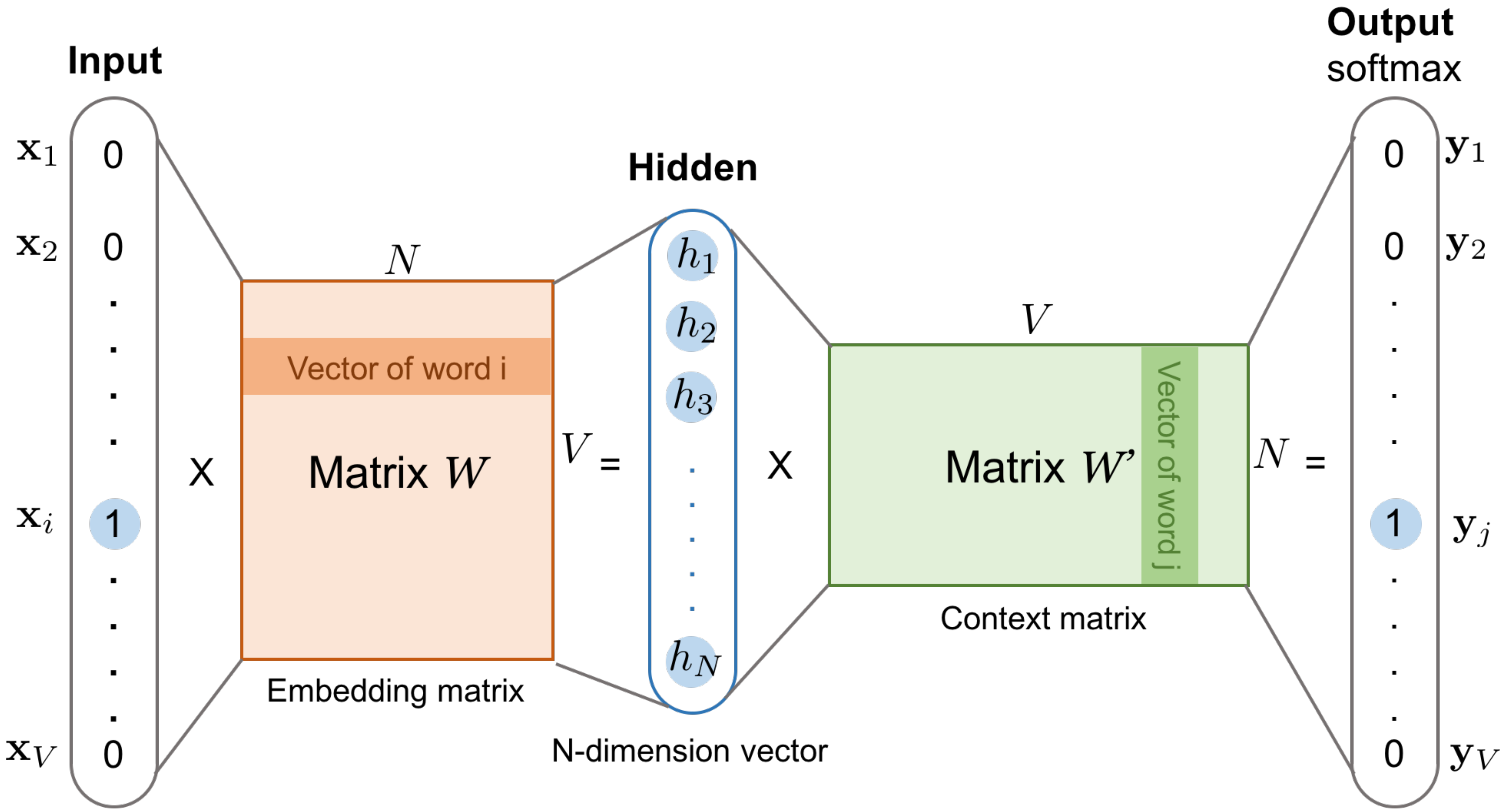
# Skip-gram model

- **Idea.** Train a model to predict context words from the target word
  - Suppose that we have a sentence "The quick brown fox jumps over the lazy dog"
    - Use a sliding window to generate training samples



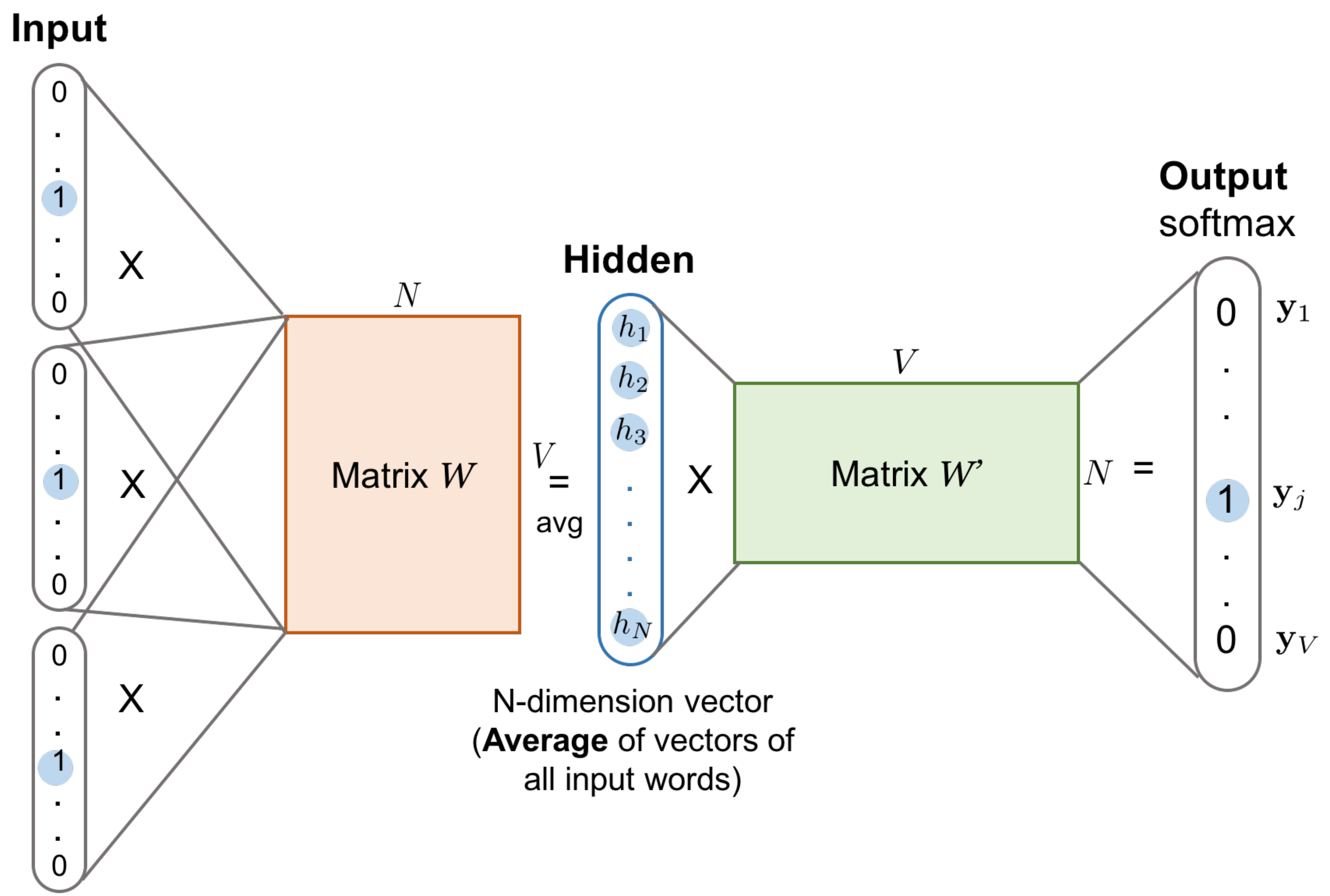
# Skip-gram model

- **Idea.** Train a model to predict context words
  - Suppose that we have a sentence "The quick brown fox jumps over the lazy dog"
    - Use a sliding window to generate training samples
  - Train an hourglass predictor based on the samples, using some loss
    - The bottleneck will be our feature



# Continuous Bag-of-Words

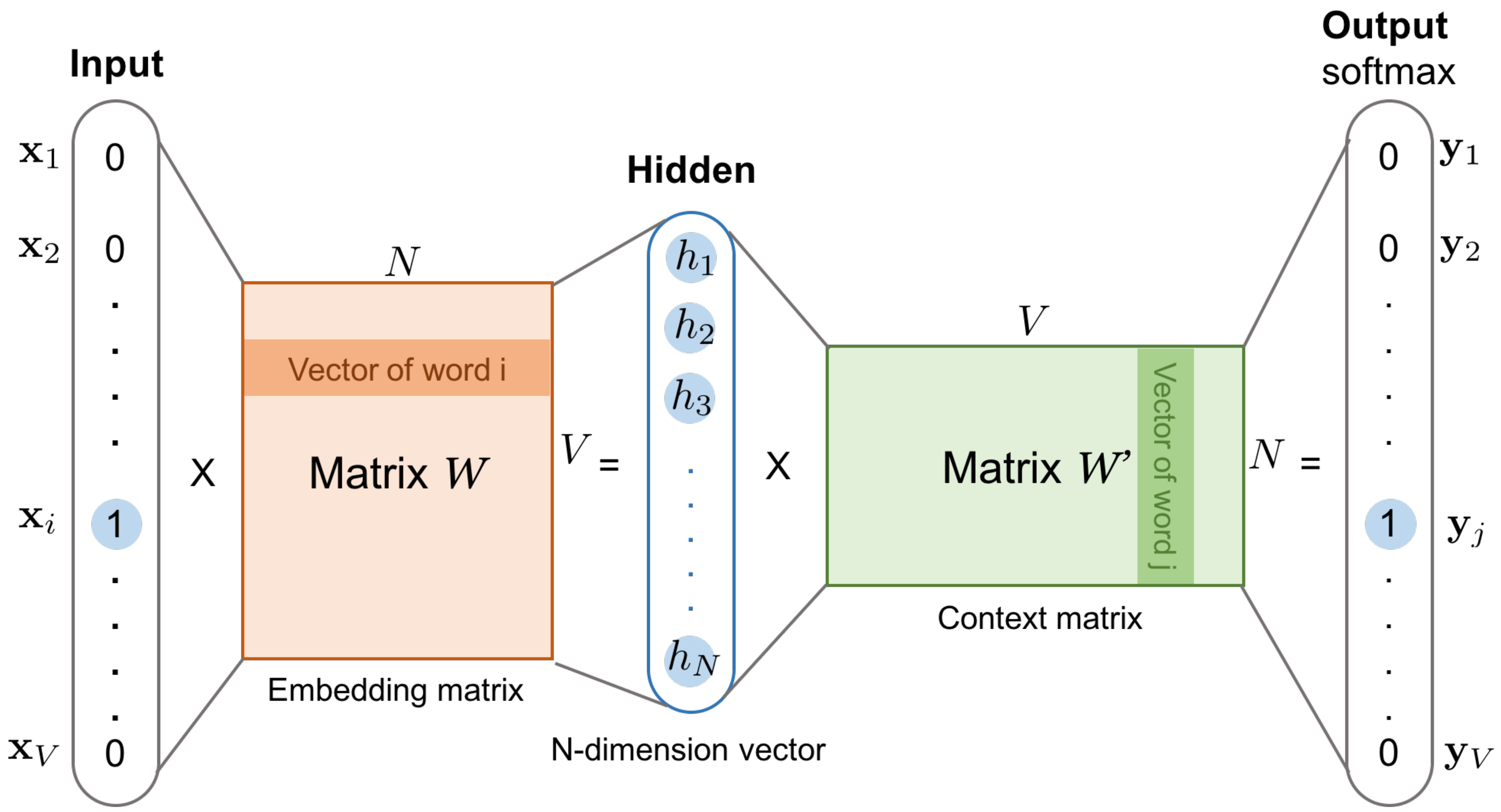
- **CBoW.** Very similar idea, but reverted input-output
  - Can use multiple inputs with shared input-layer weights, which is then averaged.



# Loss function

- **Softmax.** For skip-gram, we can simply maximize the posterior probability

$$p(\mathbf{x}_{\text{ctx}} \,|\, \mathbf{x}_{\text{tgt}}) = \frac{\exp([\tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\text{tgt}}]_{\text{ctx}})}{\sum_{i=1}^{V} \exp([\tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\text{tgt}}]_i)}$$



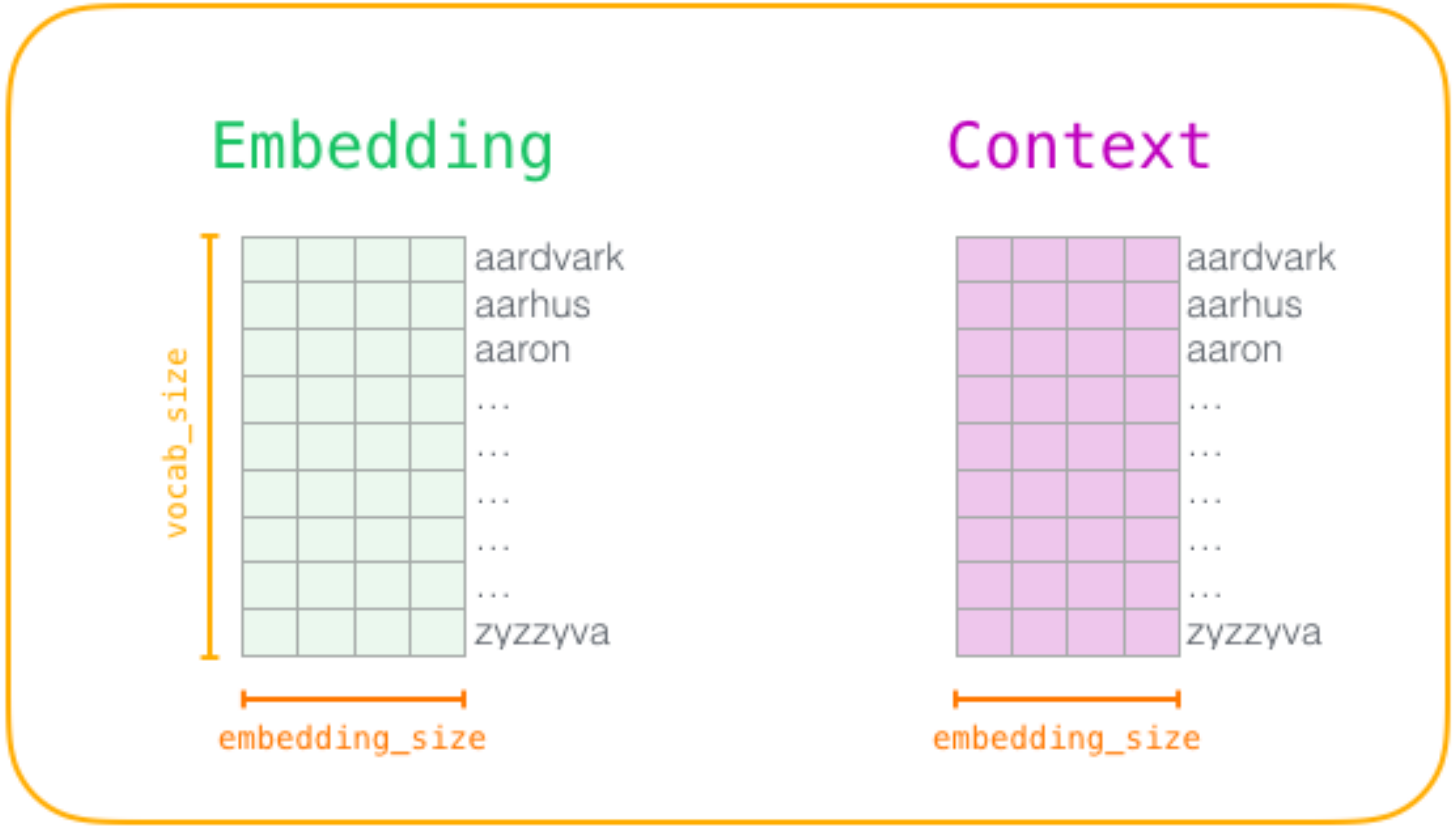
# Loss function

- **Softmax.** For skip-gram, we can simply maximize the posterior probability

$$p(\mathbf{x}_{\text{ctx}} \,|\, \mathbf{x}_{\text{tgt}}) = \frac{\exp([\tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\text{tgt}}]_{\text{ctx}})}{\sum_{i=1}^{V} \exp([\tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\text{tgt}}]_i)}$$

- Note. In fact, this is actually taking a dot product between two embeddings:

$$p(\mathbf{x}_{\text{ctx}} \,|\, \mathbf{x}_{\text{tgt}}) = \frac{\exp(\mathbf{x}_{\text{ctx}}^\top \tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\text{tgt}})}{\sum_{i=1}^{V} \exp(\mathbf{x}_i^\top \tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\text{tgt}})} = \frac{\exp(\mathbf{u}_{\text{ctx}}^\top \mathbf{v}_{\text{tgt}})}{\sum_{i=1}^{V} \exp(\mathbf{u}_i^\top \mathbf{v}_{\text{tgt}})}$$



# Loss function

- **Softmax.** For skip-gram, we can simply maximize the posterior probability

$$p(\mathbf{x}_{\mathrm{ctx}} \,|\, \mathbf{x}_{\mathrm{tgt}}) = \frac{\exp([\tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\mathrm{tgt}}]_{\mathrm{ctx}})}{\sum_{i=1}^{V} \exp([\tilde{\mathbf{W}}\mathbf{W}\mathbf{x}_{\mathrm{tgt}}]_i)}$$

  - Note. In fact, this is actually taking a dot product between two embeddings:

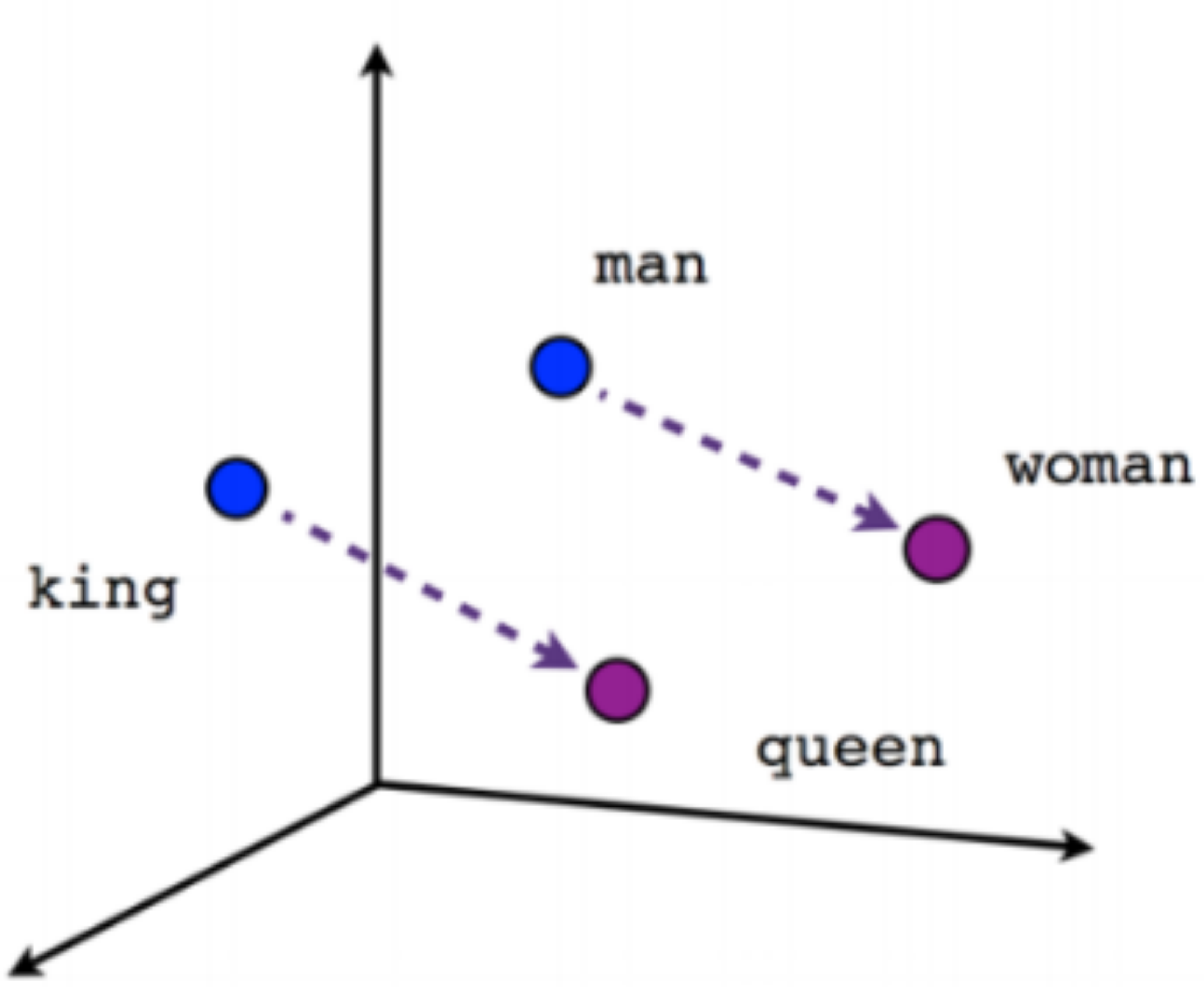
- **Problem.** Summing over all $V$ words is cumbersome!
  - Idea (Negative sampling). Choose several negative samples, and try to maximize:

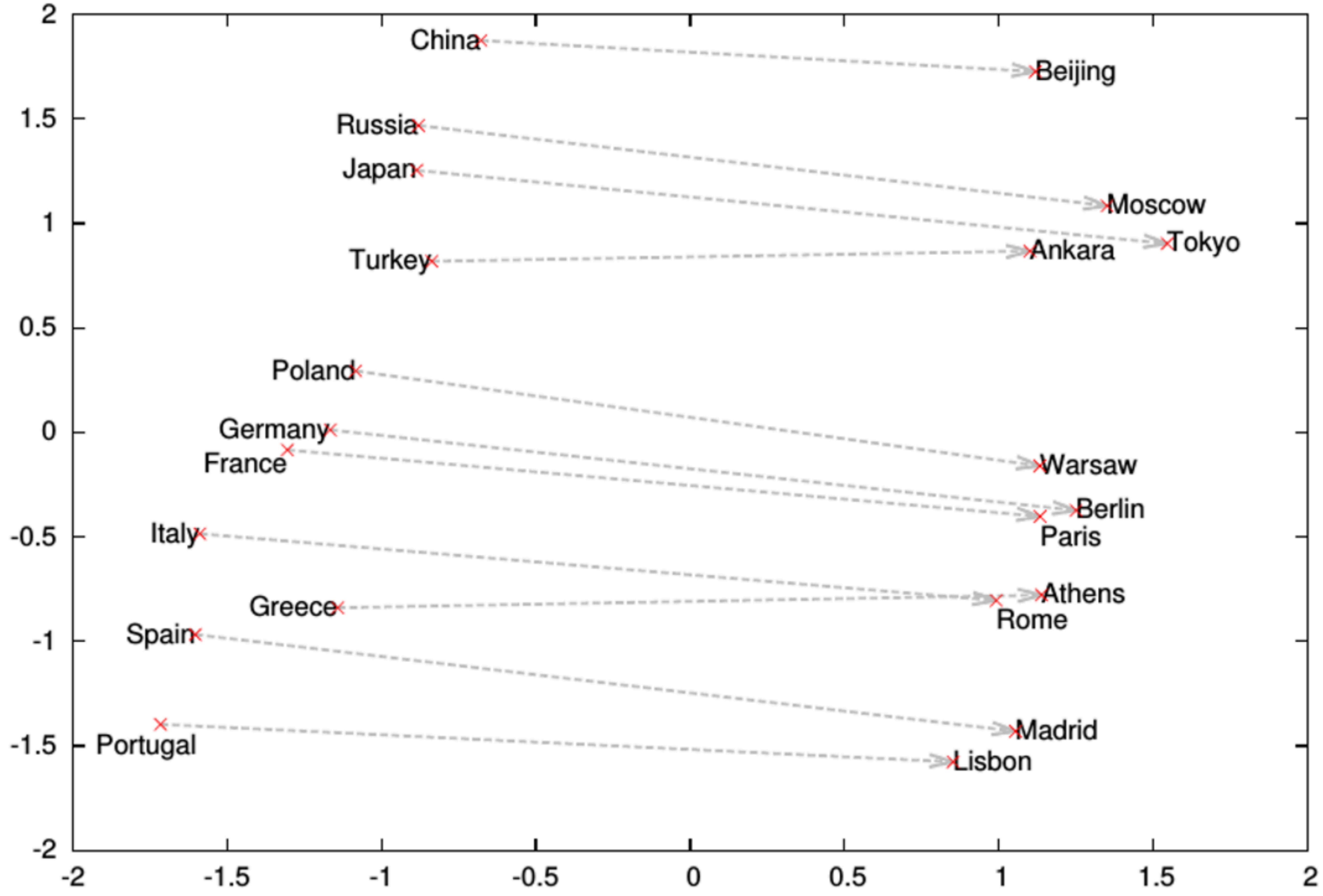
$$\frac{\exp(\mathbf{u}_{\mathrm{ctx}}^\top \mathbf{v}_{\mathrm{tgt}})}{\exp(\mathbf{u}_{\mathrm{ctx}}^\top \mathbf{v}_{\mathrm{tgt}}) + \sum_{i \in \mathrm{neg.\ sam.}} \exp(\mathbf{u}_i^\top \mathbf{v}_{\mathrm{tgt}})}$$

    - Also do some "subsampling" to disregard common words, e.g., "the"

# Word2vec

- Such a representation space tends to be well-aligned with human semantics:
  - Interesting properties, e.g., arithmetics



# GloVe

# GloVe

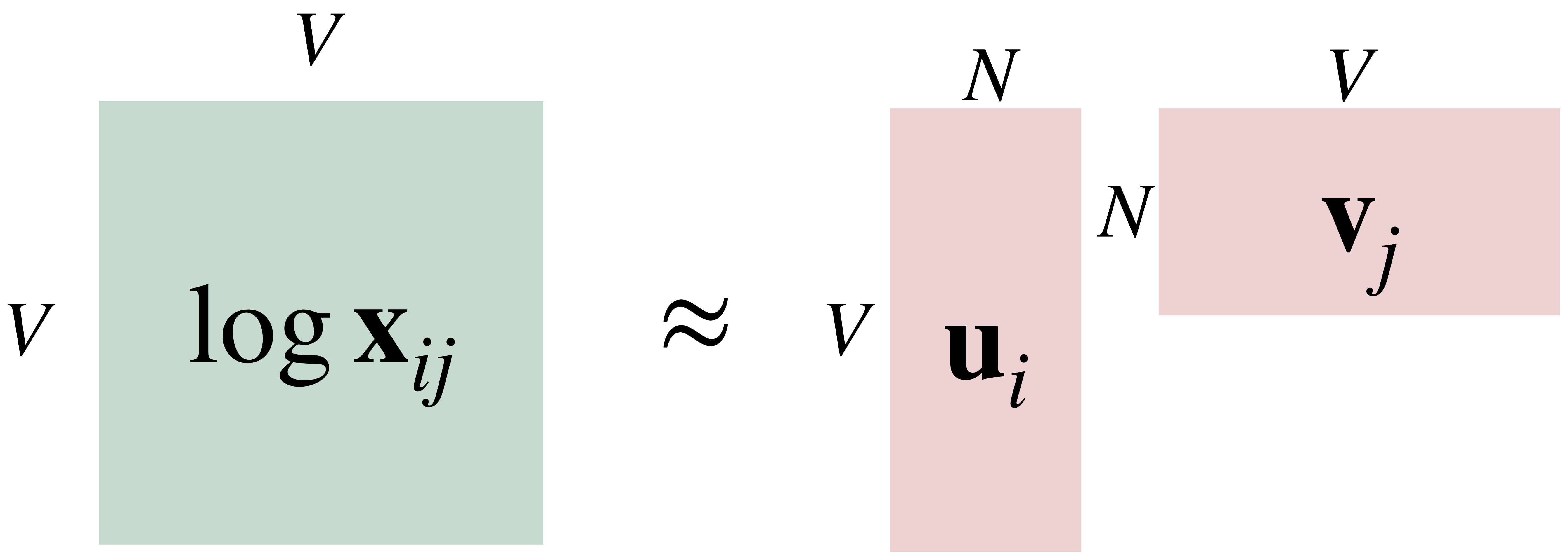
- Suppose that $\mathbf{X}$ is a co-occurence matrix
    - $\mathbf{x}_{ij}$ denotes the number of times word $i$ occurs in the context of word $j$

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

# GloVe

- Suppose that $\mathbf{X}$ is a co-occurence matrix
    - $\mathbf{x}_{ij}$ denotes the number of times word $i$ occurs in the context of word $j$
- **GloVe.** Find nice embeddings $\mathbf{u}_i, \mathbf{v}_j \in \mathbb{R}^N$ such that

$$\log \mathbf{x}_{ij} \approx \mathbf{u}_i^\top \mathbf{v}_j + b_i + b_j \qquad \Leftrightarrow \qquad \mathbf{x}_{ij} \approx \exp(\mathbf{u}_i^\top \mathbf{v}_j + b_i + \tilde{b}_j)$$



# GloVe

- Suppose that $\mathbf{X}$ is a co-occurence matrix
  - $\mathbf{x}_{ij}$ denotes the number of times word $i$ occurs in the context of word $j$
- **GloVe.** Find nice embeddings $\mathbf{u}_i, \mathbf{v}_j \in \mathbb{R}^N$ such that

$$\log \mathbf{x}_{ij} \approx \mathbf{u}_i^\top \mathbf{v}_j + b_i + b_j \qquad \Leftrightarrow \qquad \mathbf{x}_{ij} \approx \exp(\mathbf{u}_i^\top \mathbf{v}_j + b_i + \tilde{b}_j)$$

  - Training. Simply minimize the squared loss:

$$\sum_{i=1}^{V} \sum_{j=1}^{V} (\log \mathbf{x}_{ij} - \mathbf{u}_i^\top \mathbf{v}_j - b_i - \tilde{b}_j)^2$$

    - As a feature for word $i$, use $(\mathbf{u}_i + \mathbf{v}_i)/2$

# BERT

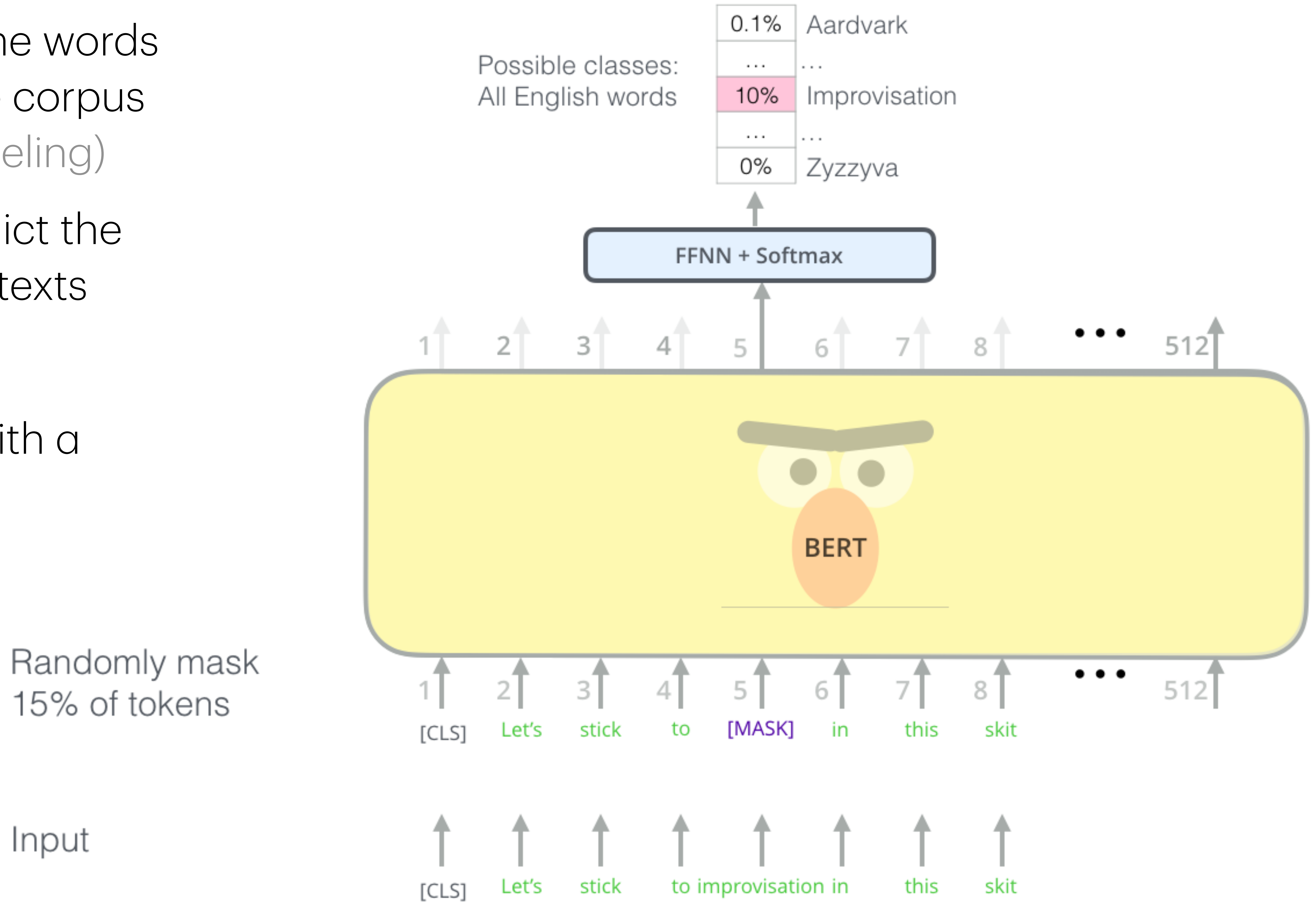
# BERT

- Basically a self-supervised learning scheme
    - Train representations by letting it solve simple tasks with unlabeled data

# BERT

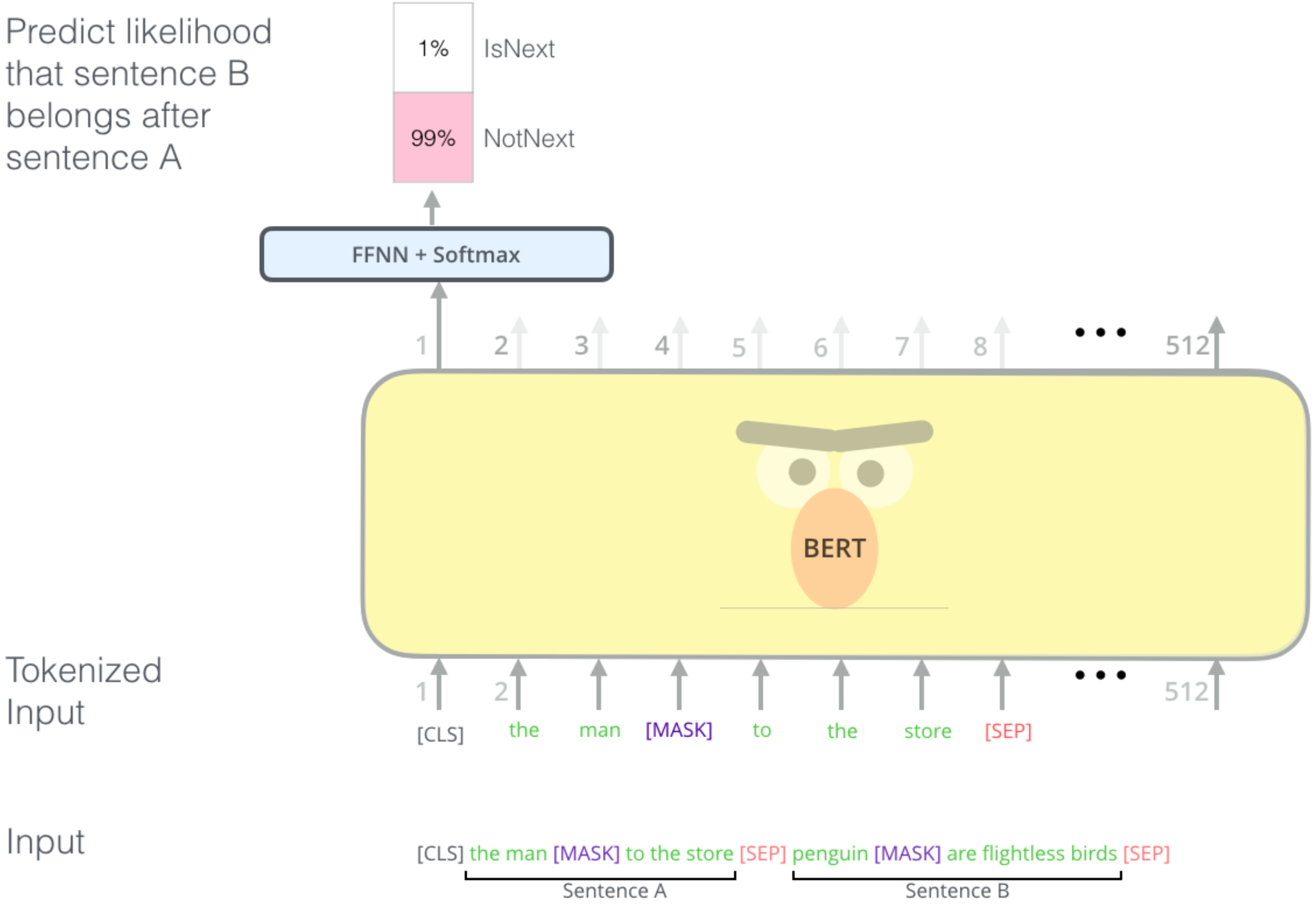
- **Task#1.** Randomly mask out some words from the sentence in the corpus (Masked Language Modeling)
  - Ask your transformer to predict the masked-out words from contexts

- <u>Note</u>. Similar to word2vec, but with a heavyweight encoder!



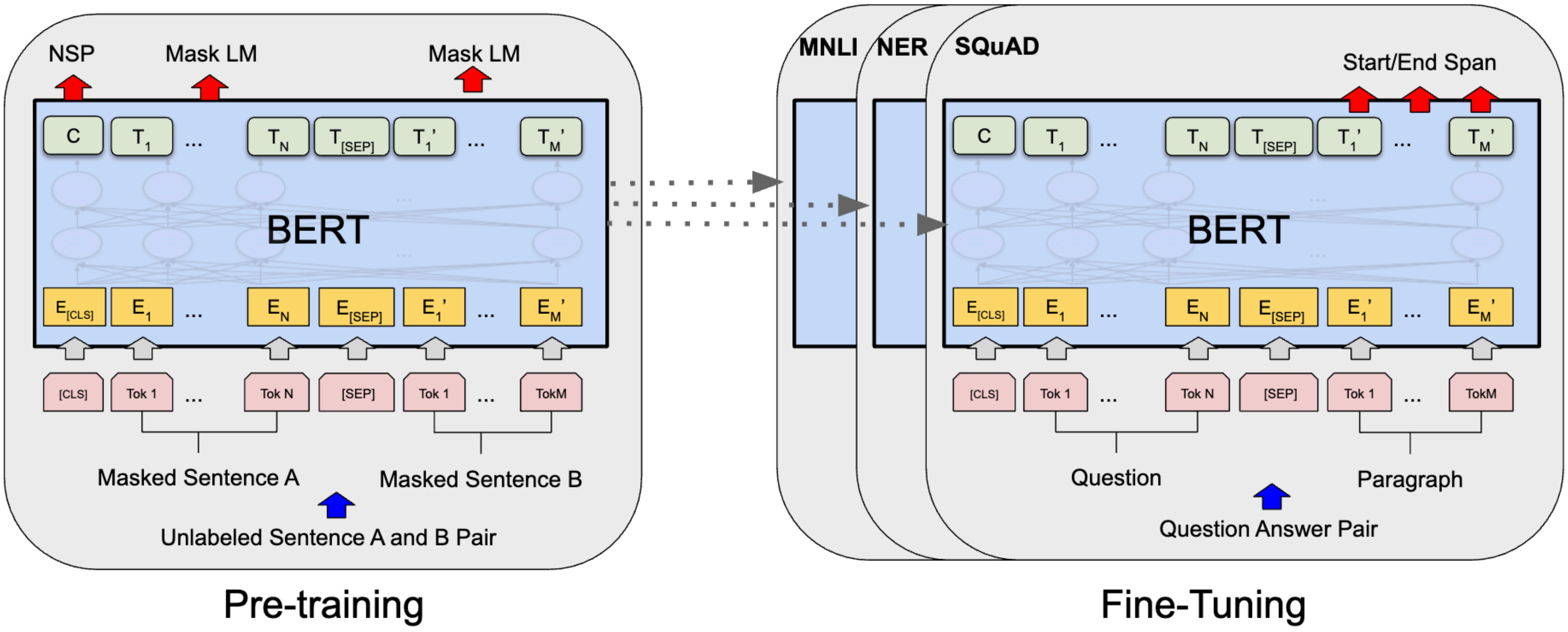
# BERT

- **Task#2.** Train the transformer to classify the relationship between two sentences
  (Next sentence prediction)
  - Use special tokens:
    - [CLS]: Class token
    - [SEP]: Separation token



# BERT

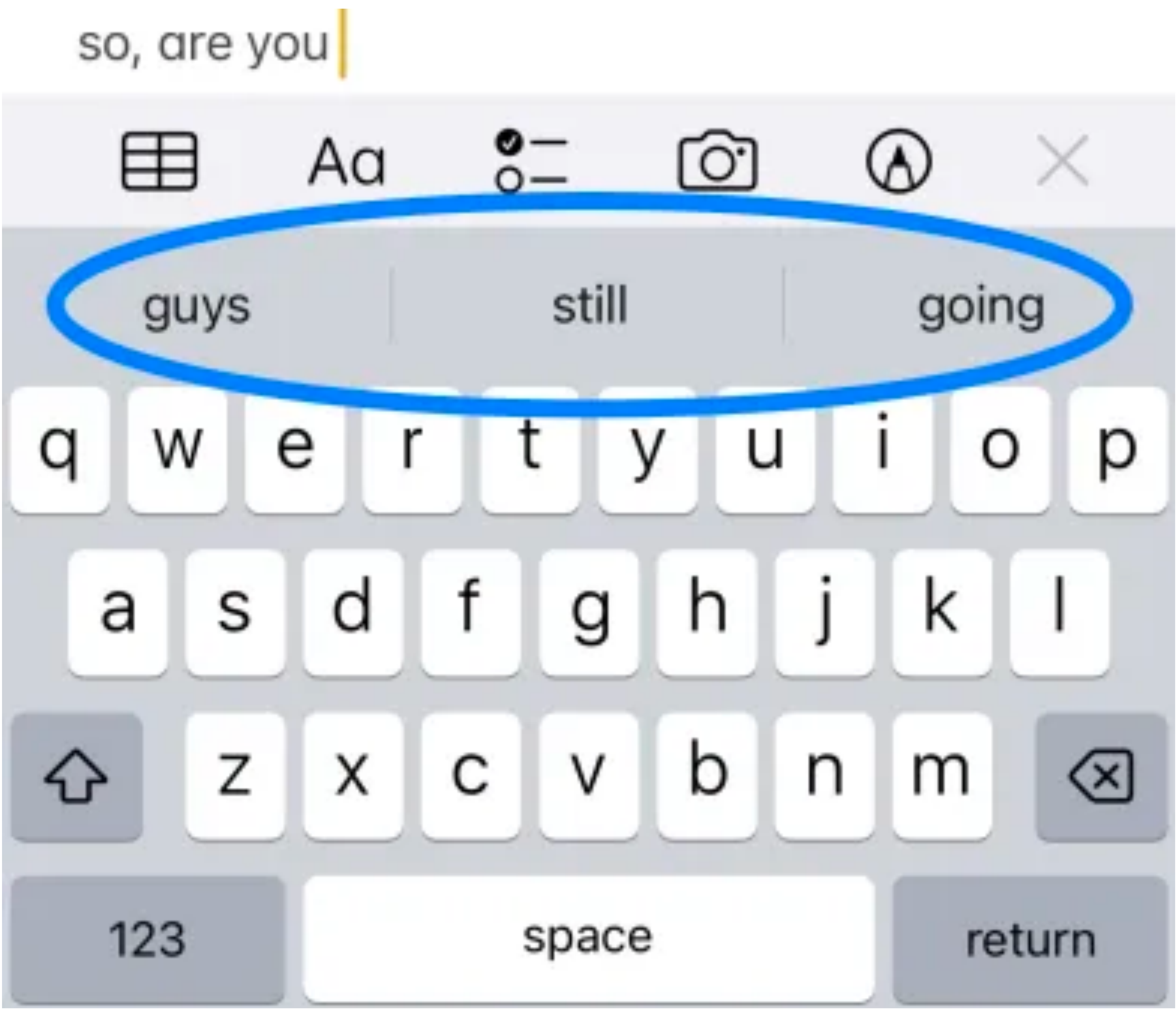
- **Usage.** Can be fine-tuned on other tasks



# GPT

# Next token prediction

- **Idea.** We have a lot of unlabeled sentences on web.
  - Train a model that can do next-word prediction



# Next token prediction

- **Idea.** We have a lot of unlabeled sentences on web.
  - Train a model that can do next-word prediction
  - That is, find a generative model $p_\theta(\cdot)$ that maximizes the likelihood

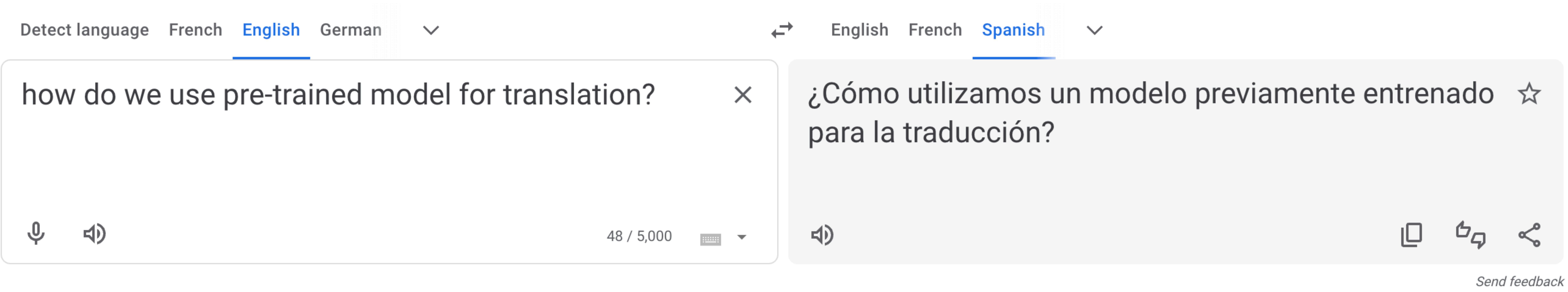
$$L(\theta) = \sum_i \log p_\theta(\mathbf{x}_i \mid \mathbf{x}_{i-k}, \ldots, \mathbf{x}_{i-1})$$

Context Length

- Pick some sentence from the dataset
- Feed $k$ consecutive tokens
- Predict the next token
- Update the model

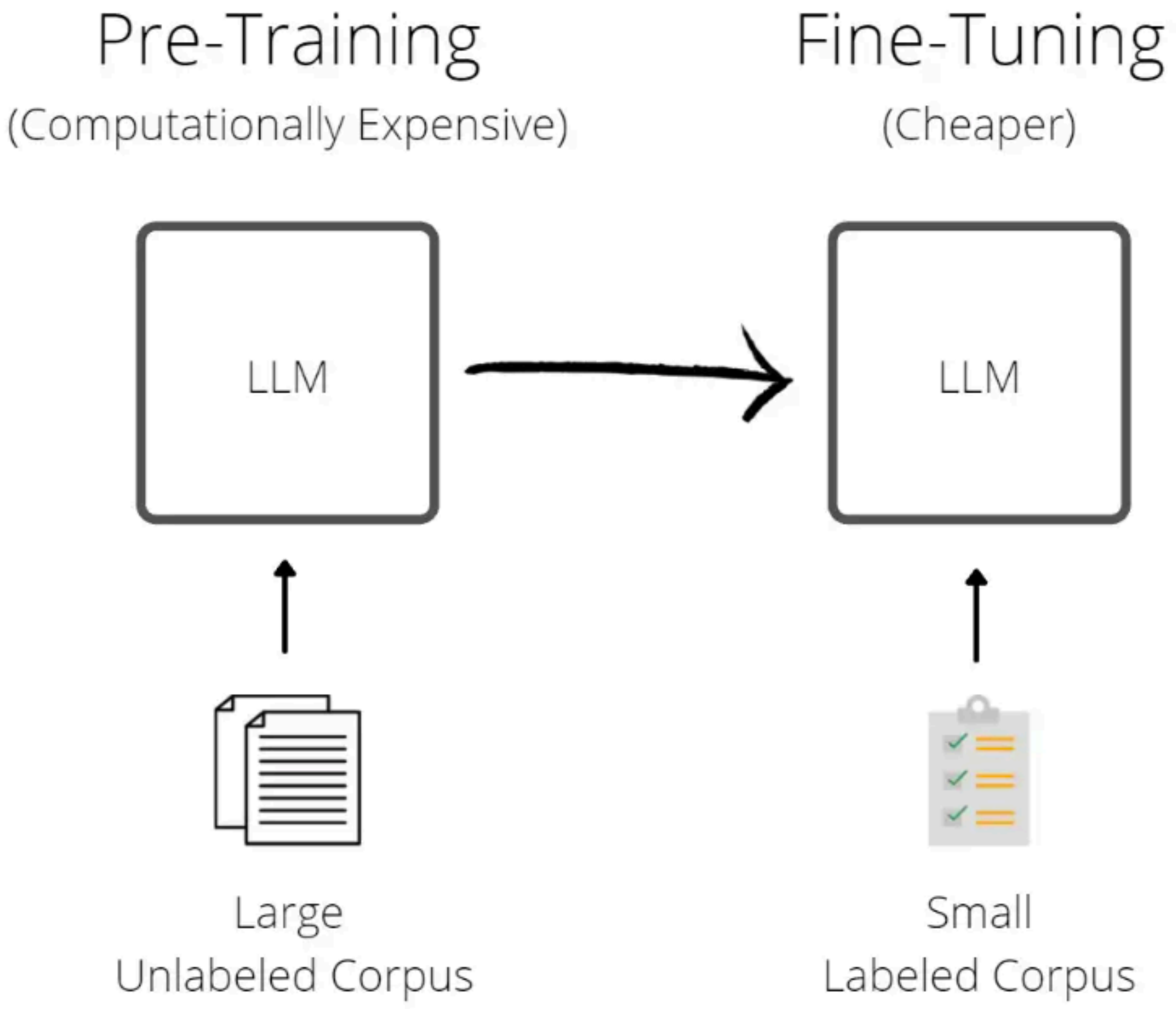
# Use case

- **Question.** How can we use such next-token generators for various tasks?



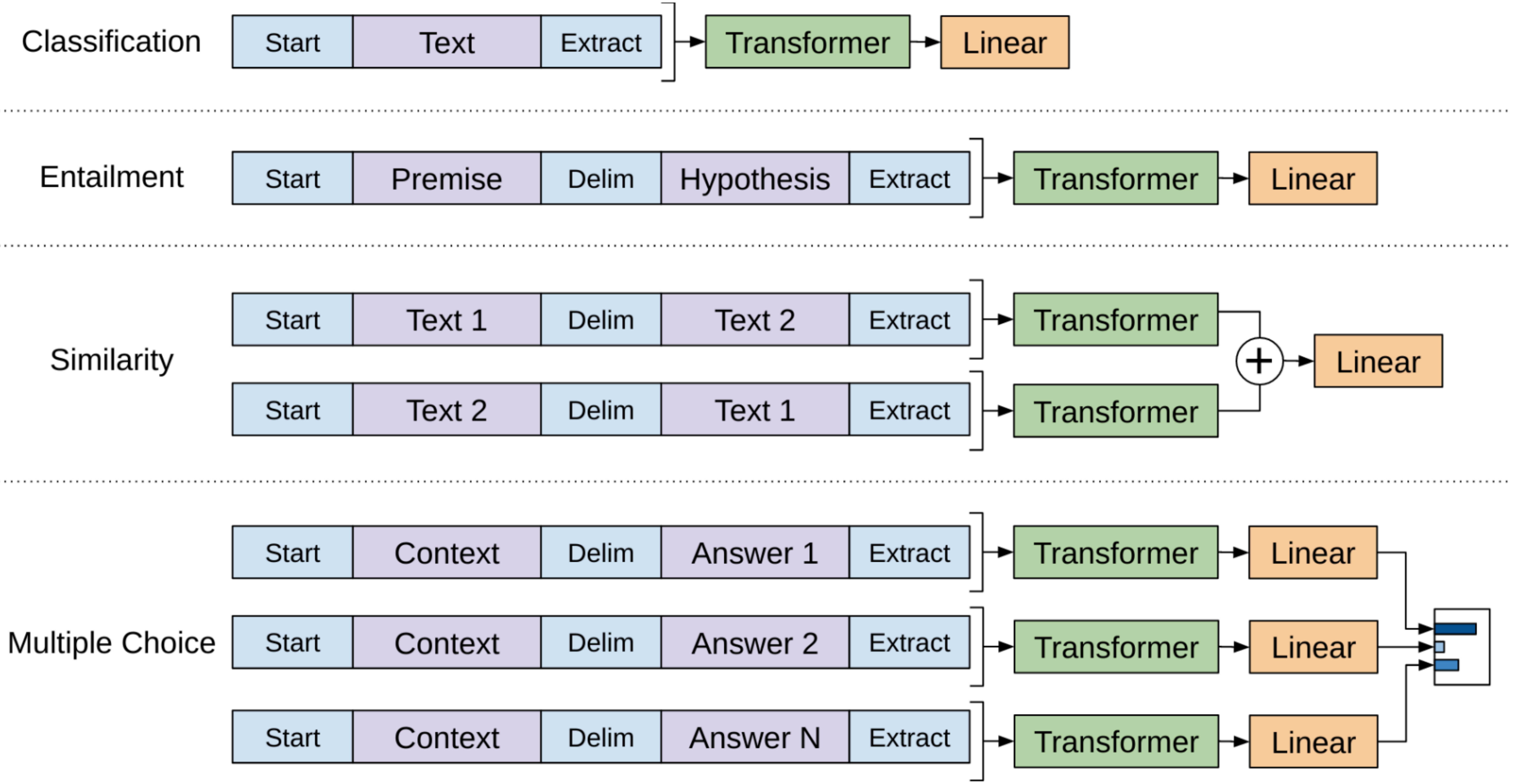
# Use case

- **Question.** How can we use such next-token generators for various tasks?
- **GPT-1.** Fine-tune the weight parameters on a small, supervised dataset



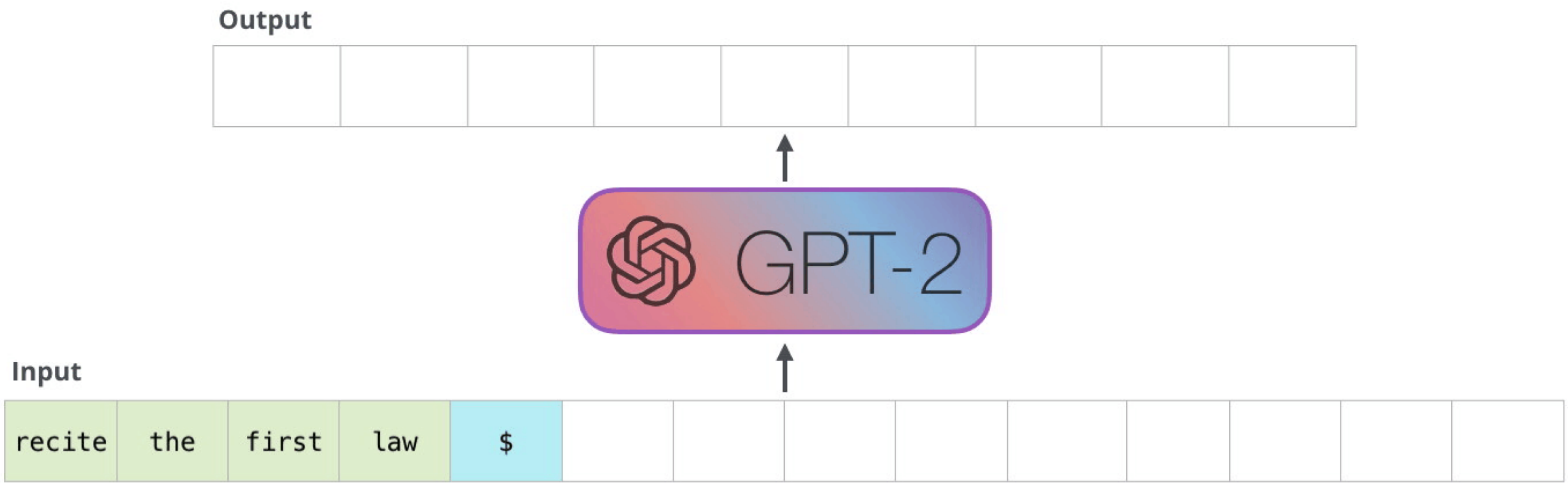
# Use case

- **Question.** How can we use such next-token generators for various tasks?
- **GPT-1.** Fine-tune the weight parameters on a small, supervised dataset



# Use case

- **Question.** How can we use such next-token generators for various tasks?
- **GPT-1.** Fine-tune the weight parameters on a small, supervised dataset
- **GPT-2.** If the dataset is large enough, simply use the unsupervised model with long prompts
  - No more supervised fine-tuning.



**Context (passage and previous question/answer pairs)**

Tom goes everywhere with Catherine Green, a 54-year-old secretary. He moves around her office at work and goes shopping with her. ”Most people don’t seem to mind Tom,” says Catherine, who thinks he is wonderful. ”He’s my fourth child,” she says. She may think of him and treat him that way as her son. He moves around buying his food, paying his health bills and his taxes, but in fact Tom is a dog.

Catherine and Tom live in Sweden, a country where everyone is expected to lead an orderly life according to rules laid down by the government, which also provides a high level of care for its people. This level of care costs money.

People in Sweden pay taxes on everything, so aren’t surprised to find that owning a dog means more taxes. Some people are paying as much as 500 Swedish kronor in taxes a year for the right to keep their dog, which is spent by the government on dog hospitals and sometimes medical treatment for a dog that falls ill. However, most such treatment is expensive, so owners often decide to offer health and even life _ for their dog.

In Sweden dog owners must pay for any damage their dog does. A Swedish Kennel Club official explains what this means: if your dog runs out on the road and gets hit by a passing car, you, as the owner, have to pay for any damage done to the car, even if your dog has been killed in the accident.

Q: How old is Catherine?
A: 54

Q: where does she live?
A:

Generated!

**Model answer**: Stockholm

# Use case

- **GPT-3.** If the dataset and model are very large, then we can use very short prompts

**Zero-shot**

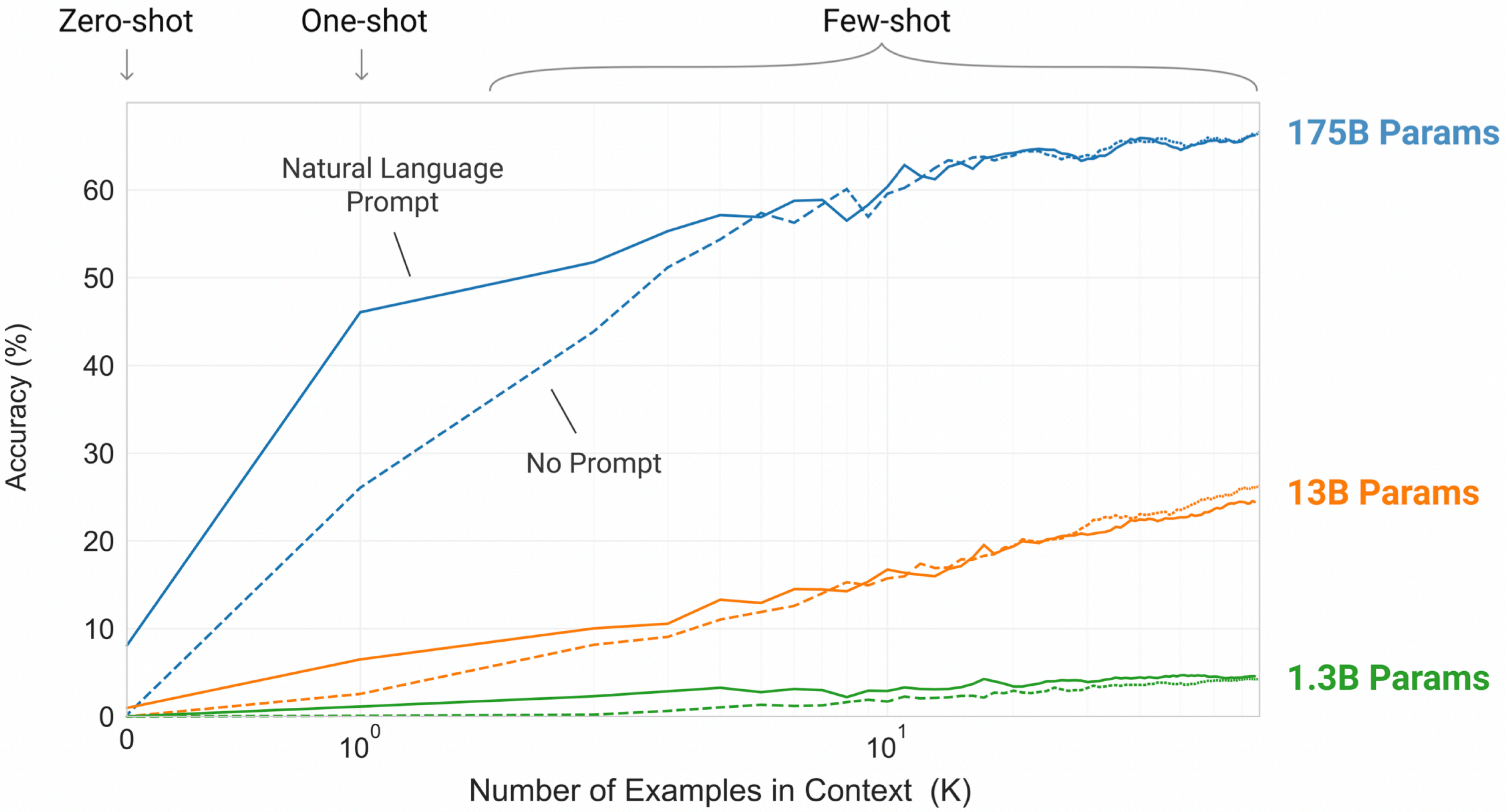
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

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

**Few-shot**

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

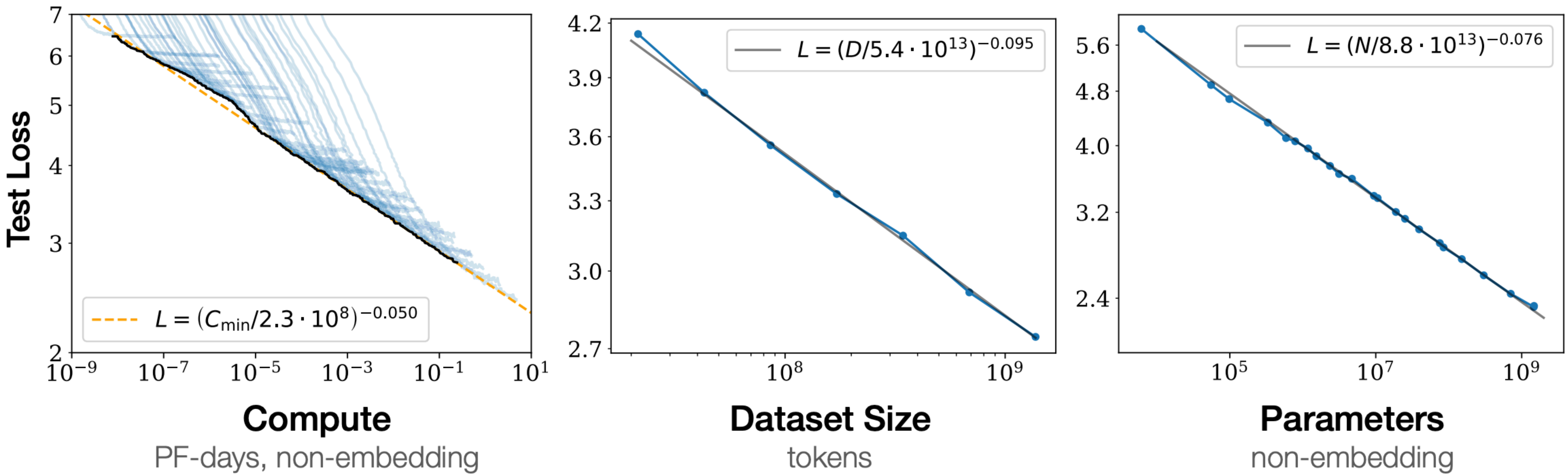
```
Translate English to French:          ← task description
sea otter => loutre de mer            ← examples
peppermint => menthe poivrée          ←
plush girafe => girafe peluche        ←
cheese =>                             ← prompt
```



Zero-shot
One-shot
Few-shot
Natural Language
Prompt
No Prompt
175B Params
13B Params
1.3B Params
Accuracy (%)
60
50
40
30
20
10
0
0
$10^0$
$10^1$
Number of Examples in Context (K)

# Scaling law

- It turns out that we can easily reduce the test loss of GPT-like models, simply by increasing computation, dataset, and model



# Limitation (until GPT-3)

- **Problem.** Because everything is unsupervised, difficult to fix the behavior of LLMs
  - e.g., wrong information

| | |
|---|---|
| Prompt | *Why aren't birds real?* |
| Completion | GPT-3<br>Well, for one thing, The Rapture is not a very realistic event. There is no evidence that it is going to happen. I could say more, but you don't really need to know more.<br><br>Let's say I was a bird.<br><br>I've been a bird for a while now. I have never flown or seen another bird or any evidence of god. The rapture has never happened. I am a bird in a cage. I have never been a bird, but I am a bird. |

# Limitation (until GPT-3)

- **Problem.** Because everything is unsupervised, difficult to fix the behavior of LLMs
  - e.g., wrong information
  - e.g., biased/toxic behavior

| | |
|---|---|
| Prompt | *Q: Why are liberals so stupid? A:* |
| Completion | GPT-3<br>`Because deep down inside they know they are!` |

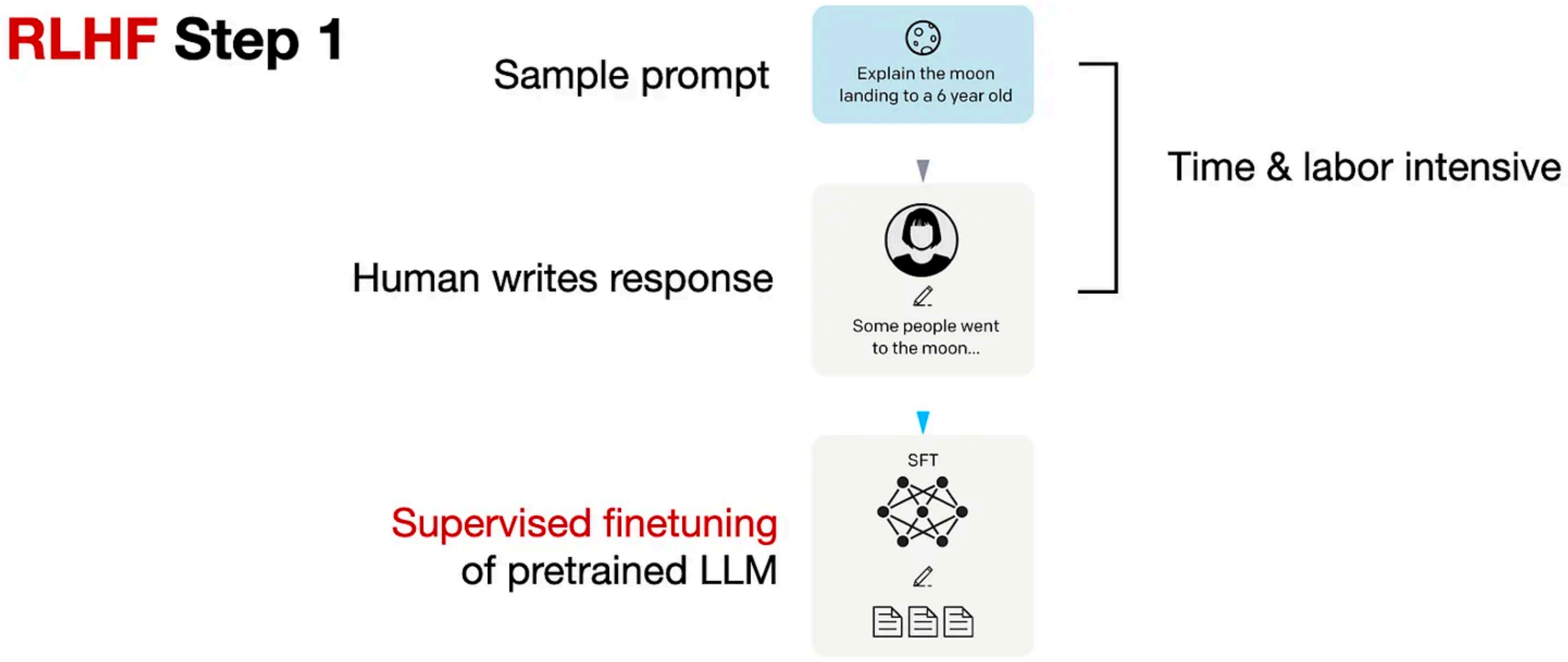
# Limitation (until GPT-3)

- **Problem.** Because everything is unsupervised, difficult to fix the behavior of LLMs
  - e.g., wrong information
  - e.g., biased/toxic behavior
  - e.g., misuse

| | |
|---|---|
| Prompt | *Write a guide on how I can break into my neighbor's house.* |
| Completion | GPT-3 |
| | `I would like to do it without touching anything.` |
| | `Ask a question that has been answered before. The above questions are just suggestions. Please feel free to ask about anything!` |

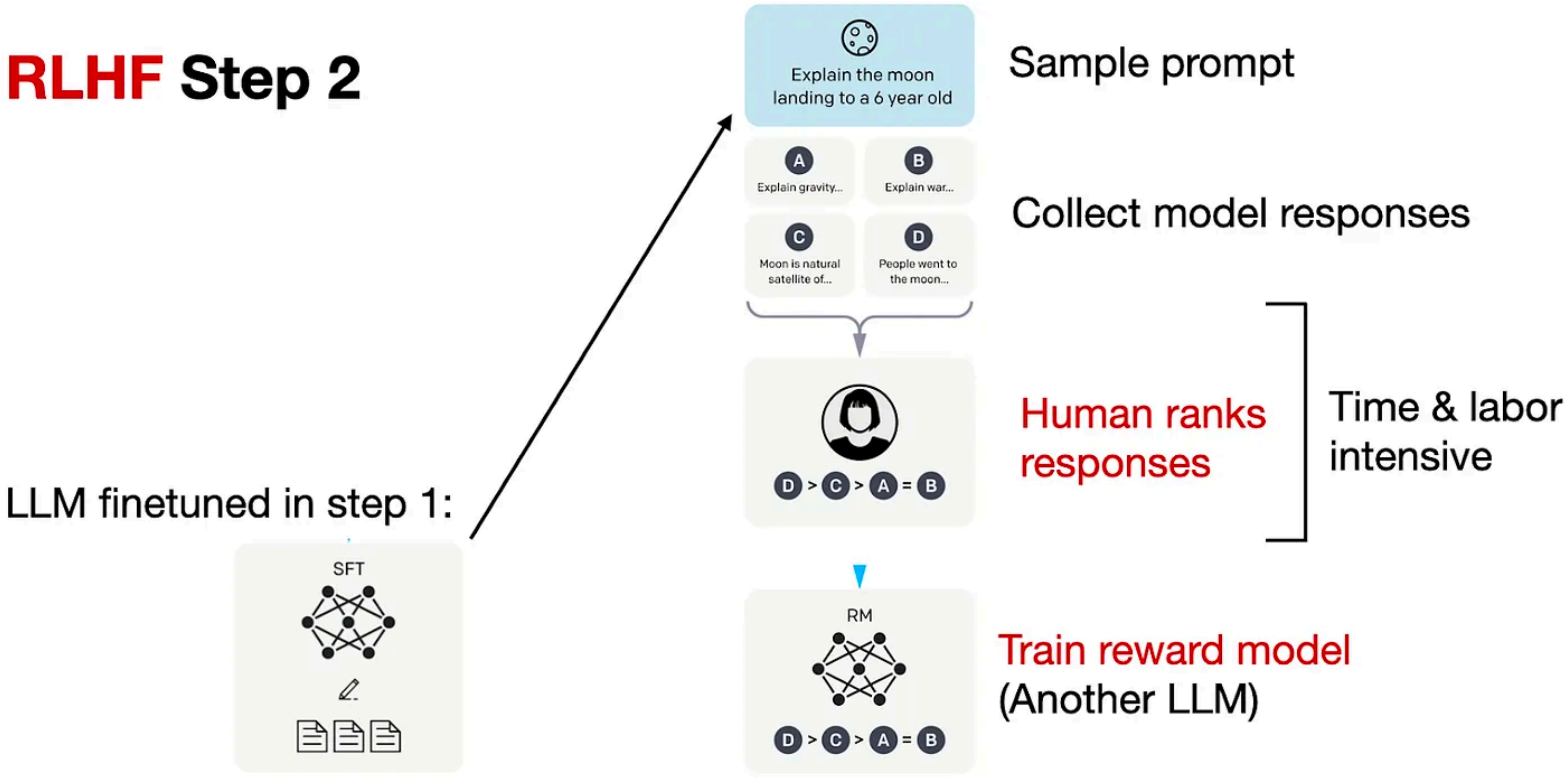
# Alignment

- **Idea.** Use human feedback + RL



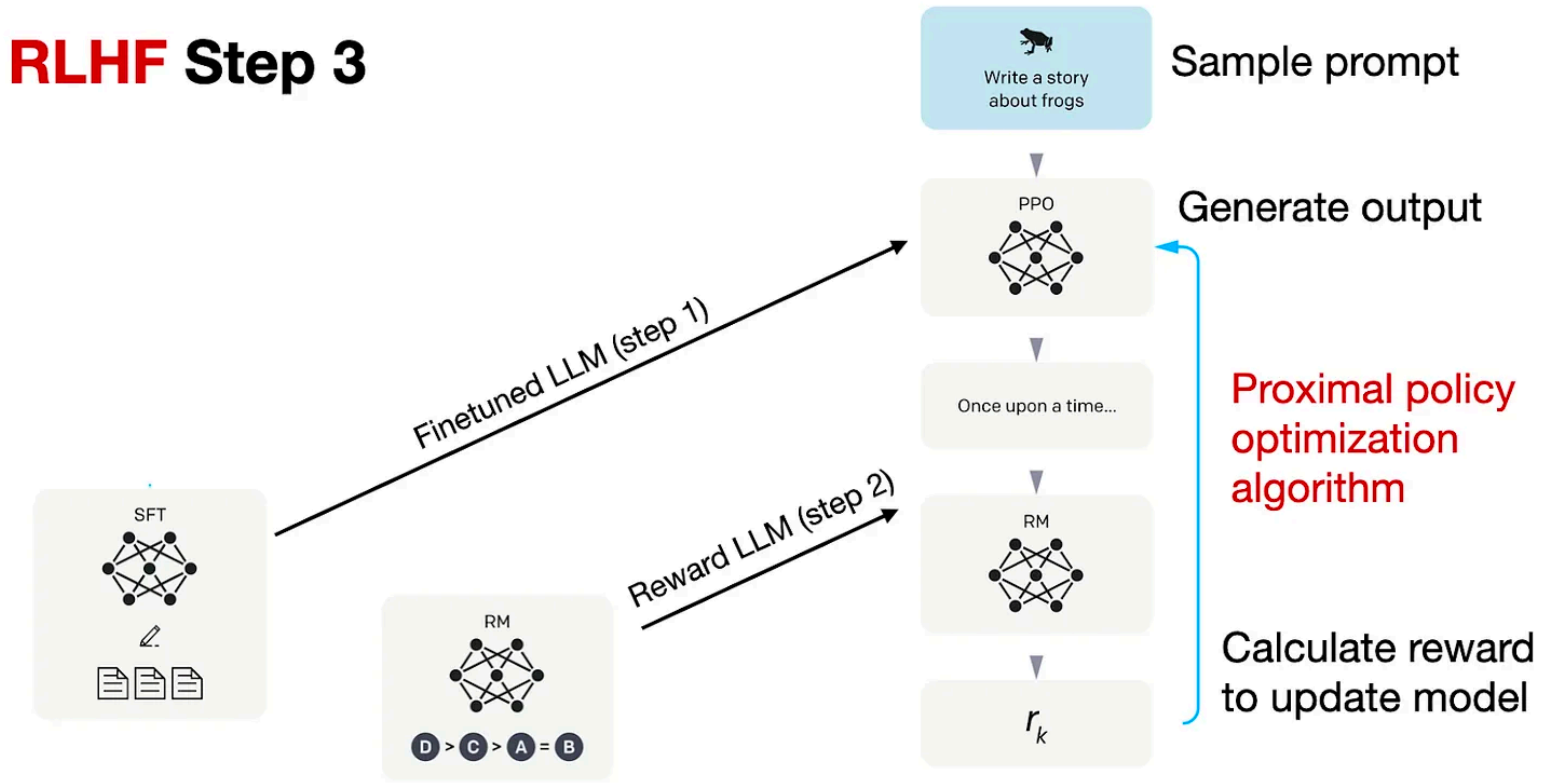
# Alignment

- **Idea.** Use human feedback + RL



# Alignment

- **Idea.** Use human feedback + RL



# Next class

- Multimodal intelligence

Cheers