Sampling and Generation

Ling 282/482: Deep Learning for Computational Linguistics

C.M. Downey

Fall 2025

Generation / Decoding

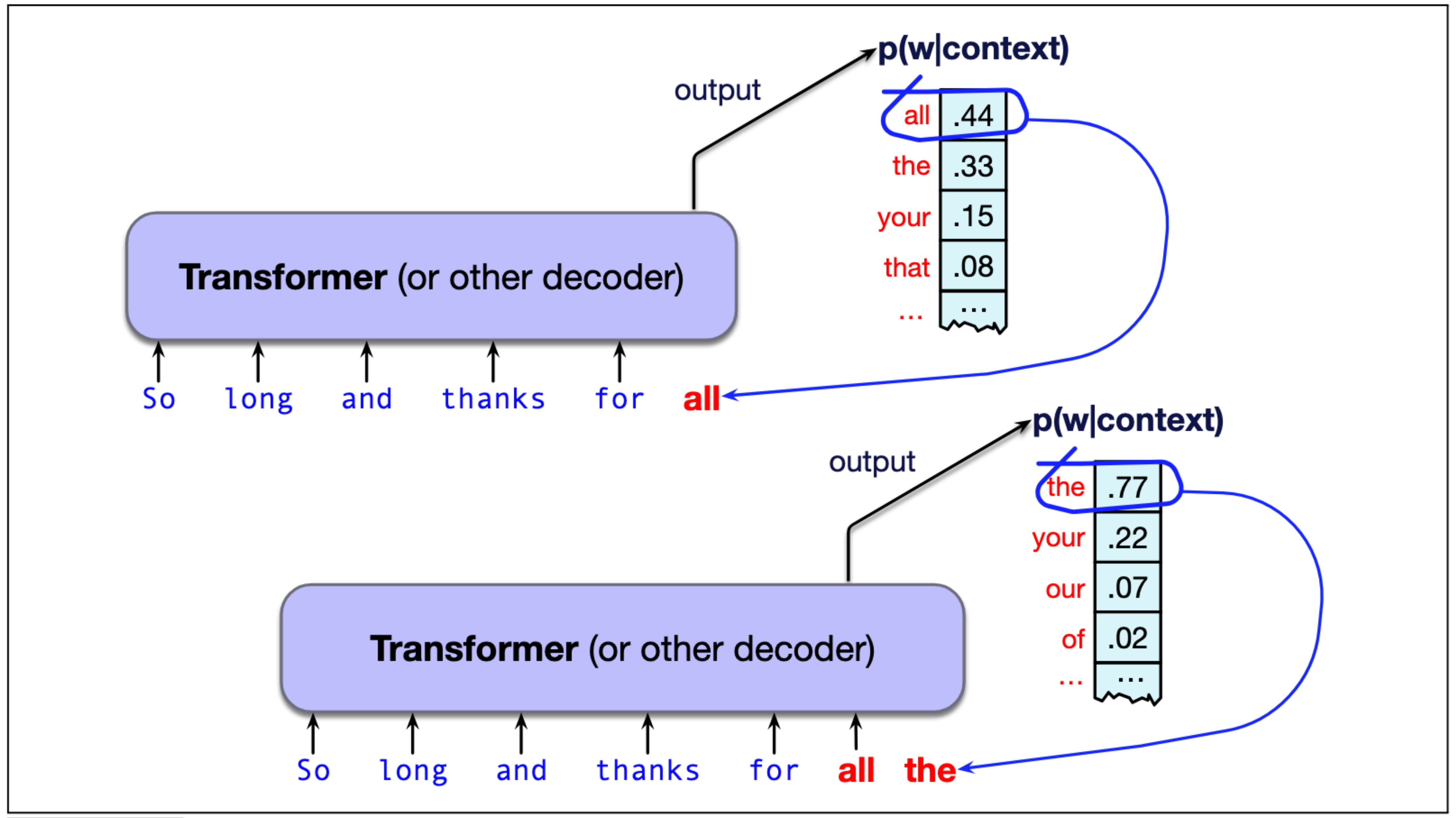


Figure 7.2 Turning a predictive model that gives a probability distribution over next words into a generative model by repeatedly sampling from the distribution. The result is a left-to-right (also called autoregressive) language models. As each token is generated, it gets added onto the context as a prefix for generating the next token.

Generation / Decoding

- A Language Model outputs a **probability distribution** over possible words
  - The LM encodes the probability for **all possible sequences**
  - Given this, how do we decide what the **predicted sequence** should be? (This process is often called **"decoding"**)

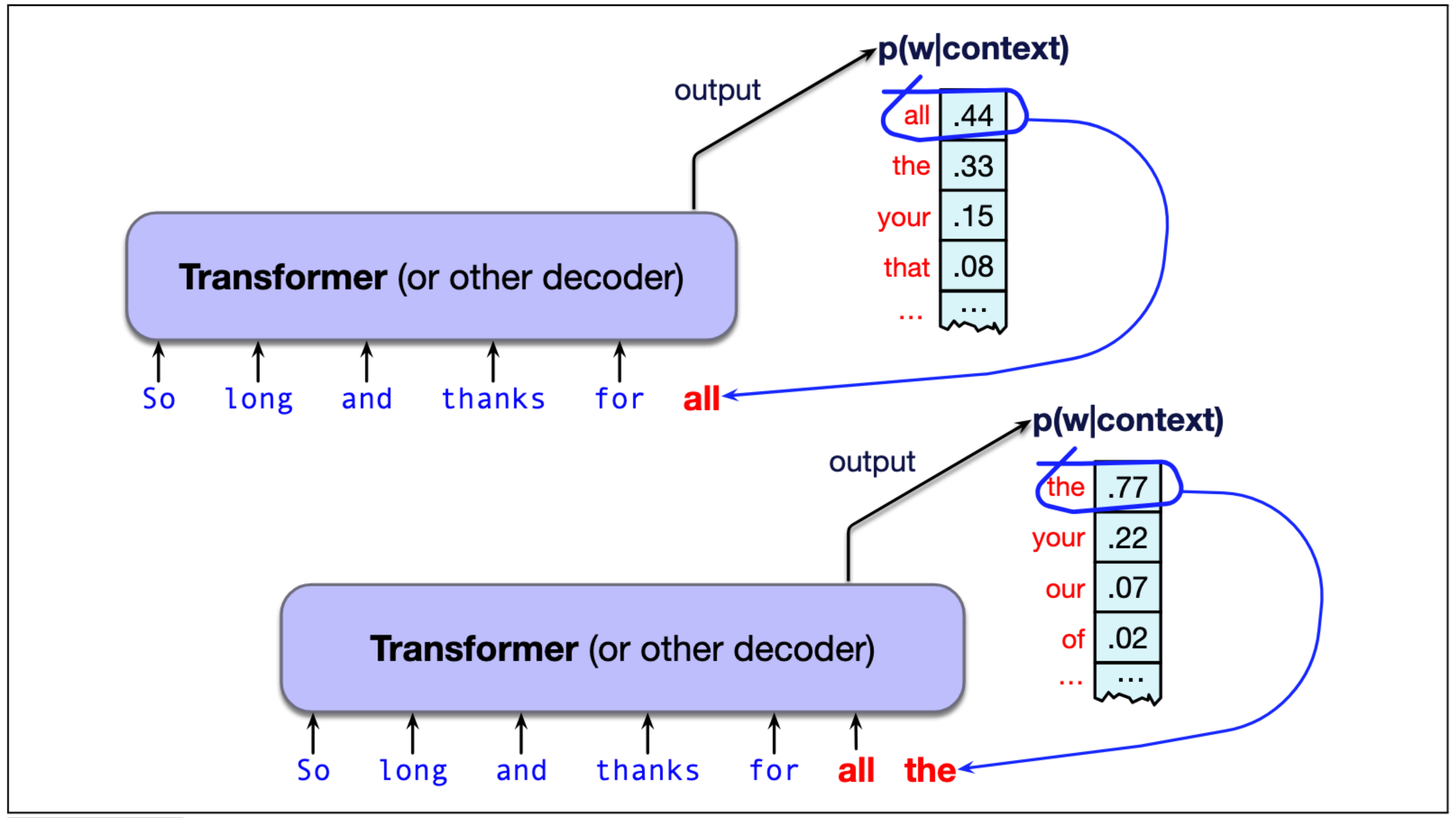


Figure 7.2 Turning a predictive model that gives a probability distribution over next words into a generative model by repeatedly sampling from the distribution. The result is a left-to-right (also called autoregressive) language models. As each token is generated, it gets added onto the context as a prefix for generating the next token.

Generation / Decoding

- A Language Model outputs a **probability distribution** over possible words
  - The LM encodes the probability for **all possible sequences**
  - Given this, how do we decide what the **predicted sequence** should be? (This process is often called **"decoding"**)
- How do we generate **new sequences?**
  - During training, we always know what the next word should be
  - For many real-world tasks (e.g. Chatbots), we want the model to **generate novel text**

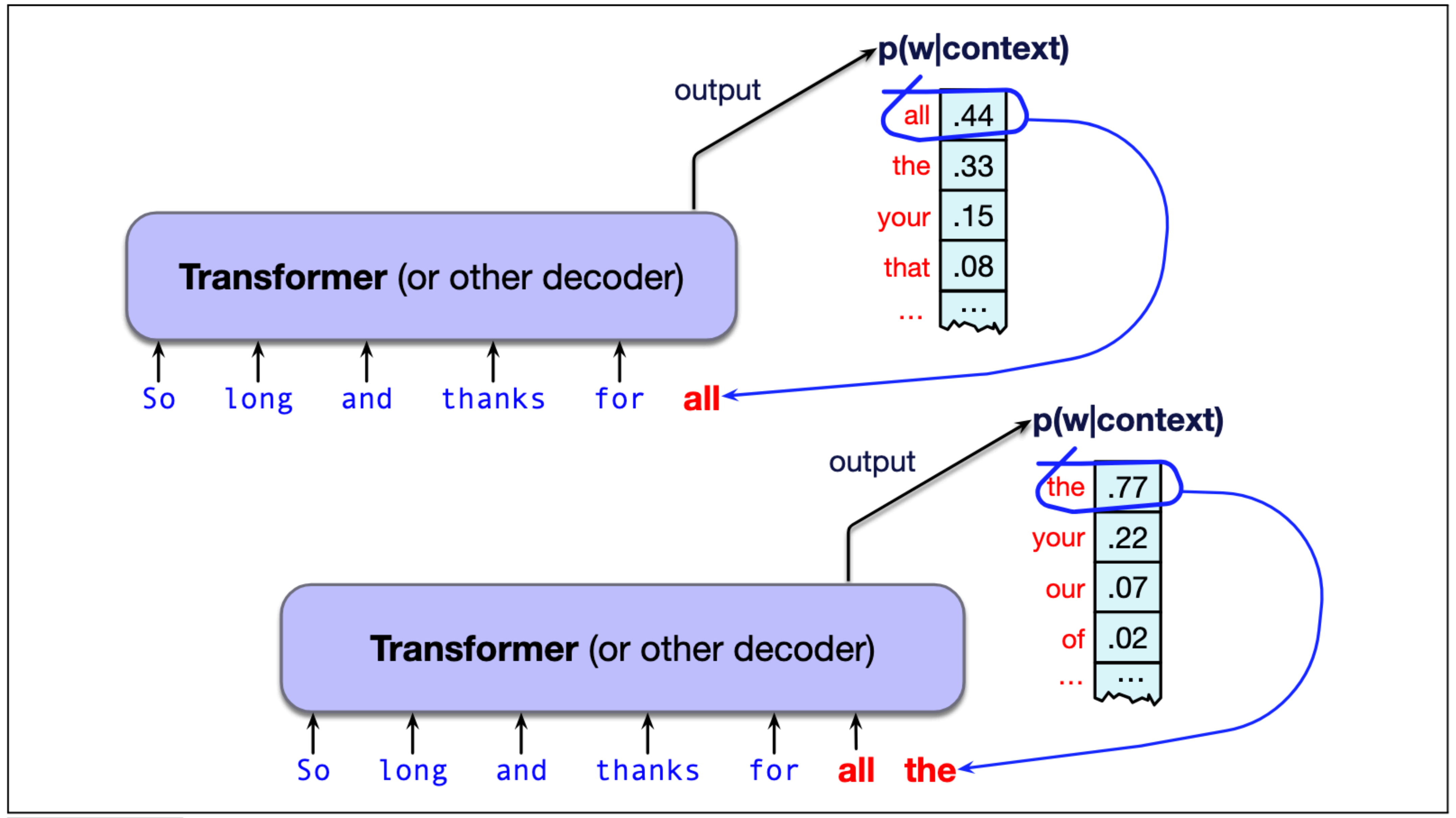


Figure 7.2 Turning a predictive model that gives a probability distribution over next words into a generative model by repeatedly sampling from the distribution. The result is a left-to-right (also called autoregressive) language models. As each token is generated, it gets added onto the context as a prefix for generating the next token.

Greedy Decoding

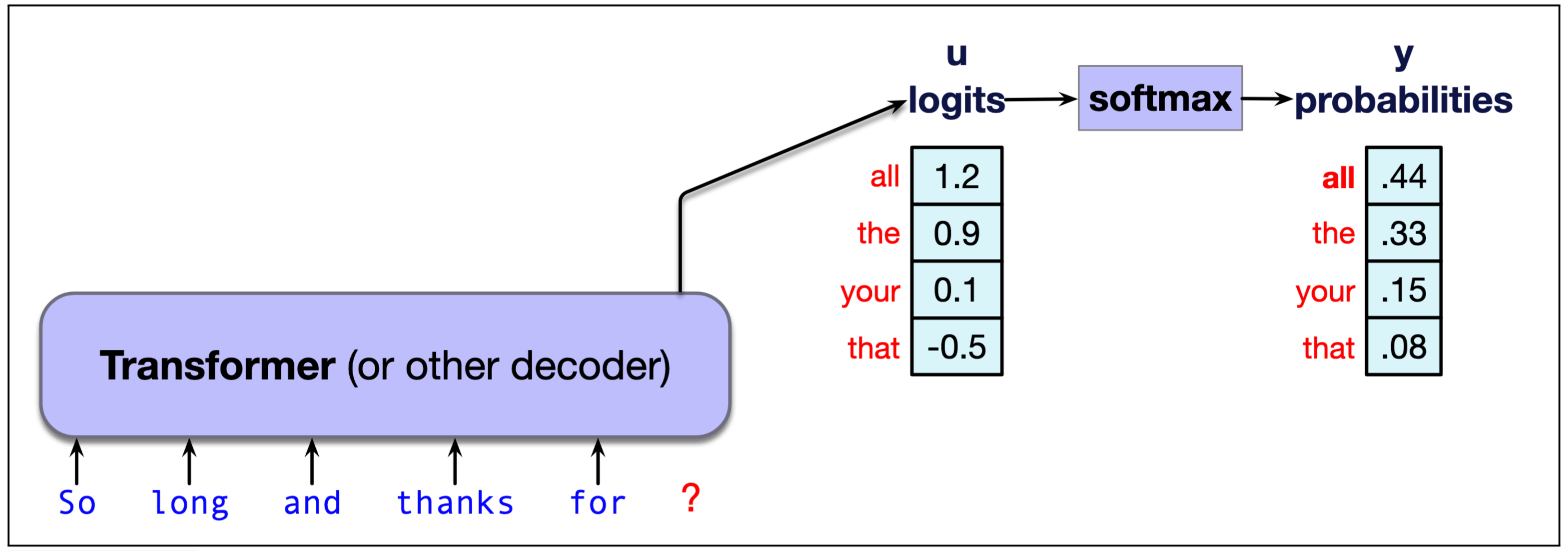


Figure 7.7 Taking the logit vector **u** and using the softmax to create a probability vector **y**.

$$\hat{w}_t = \text{argmax}_{w \in V} P(w \mid w_{<t})$$

Greedy Decoding

- **"Greedy" decoding** is the simplest strategy
  - At each time step, choose the **highest-probability word**
  - "Greedy" because it does **NOT** guarantee the highest-probability sequence

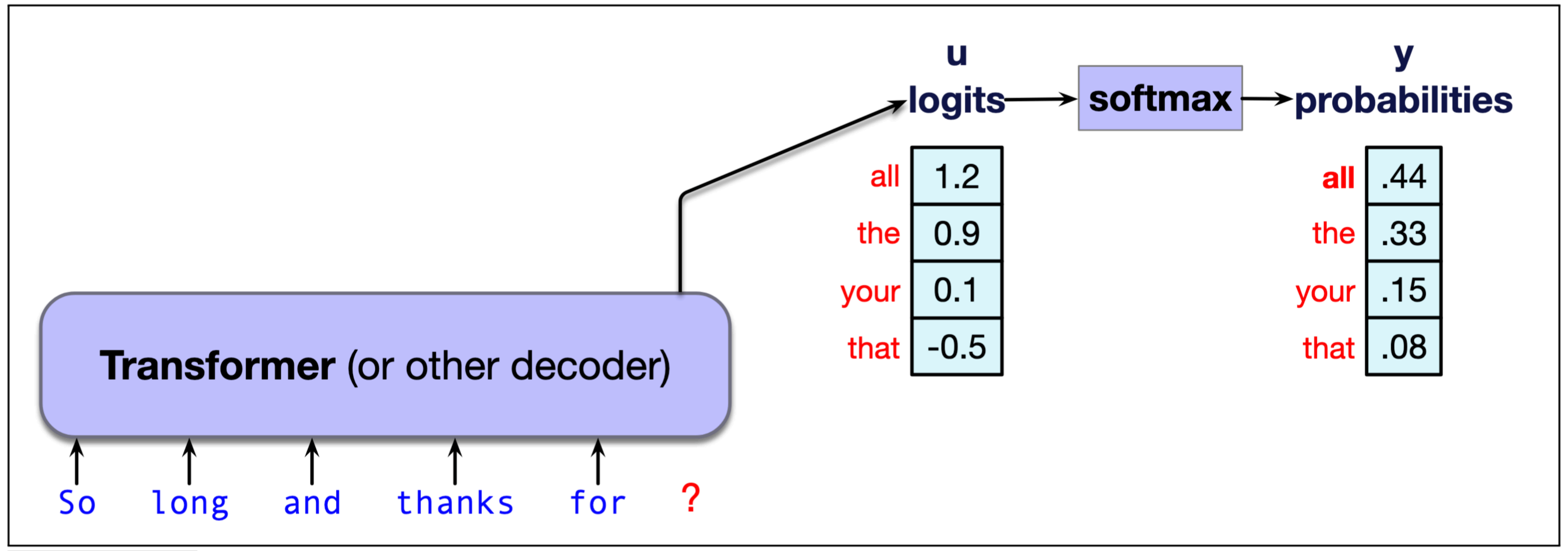


Figure 7.7 Taking the logit vector **u** and using the softmax to create a probability vector **y**.

$$\hat{w}_t = \mathrm{argmax}_{w \in V} P(w \mid w_{<t})$$

Greedy Decoding

- **"Greedy" decoding** is the simplest strategy
  - At each time step, choose the **highest-probability word**
  - "Greedy" because it does **NOT** guarantee the highest-probability sequence
- **No randomness** involved (same context gives the same completion)

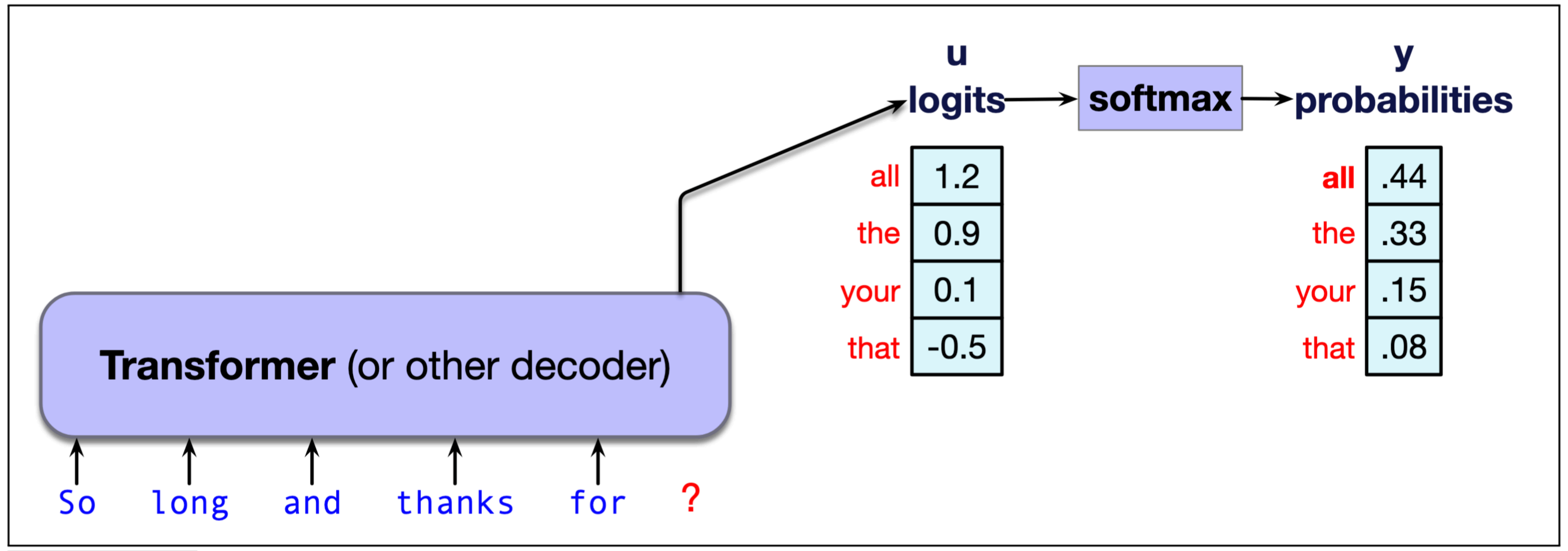


Figure 7.7 Taking the logit vector **u** and using the softmax to create a probability vector **y**.

$$\hat{w}_t = \text{argmax}_{w \in V} P(w \mid w_{<t})$$

Greedy Decoding

- **"Greedy" decoding** is the simplest strategy
  - At each time step, choose the **highest-probability word**
  - "Greedy" because it does **NOT** guarantee the highest-probability sequence
- **No randomness** involved (same context gives the same completion)
- Tends to generate **boring or repetitive** text
  - Or text that's been **exactly copied** from the training data

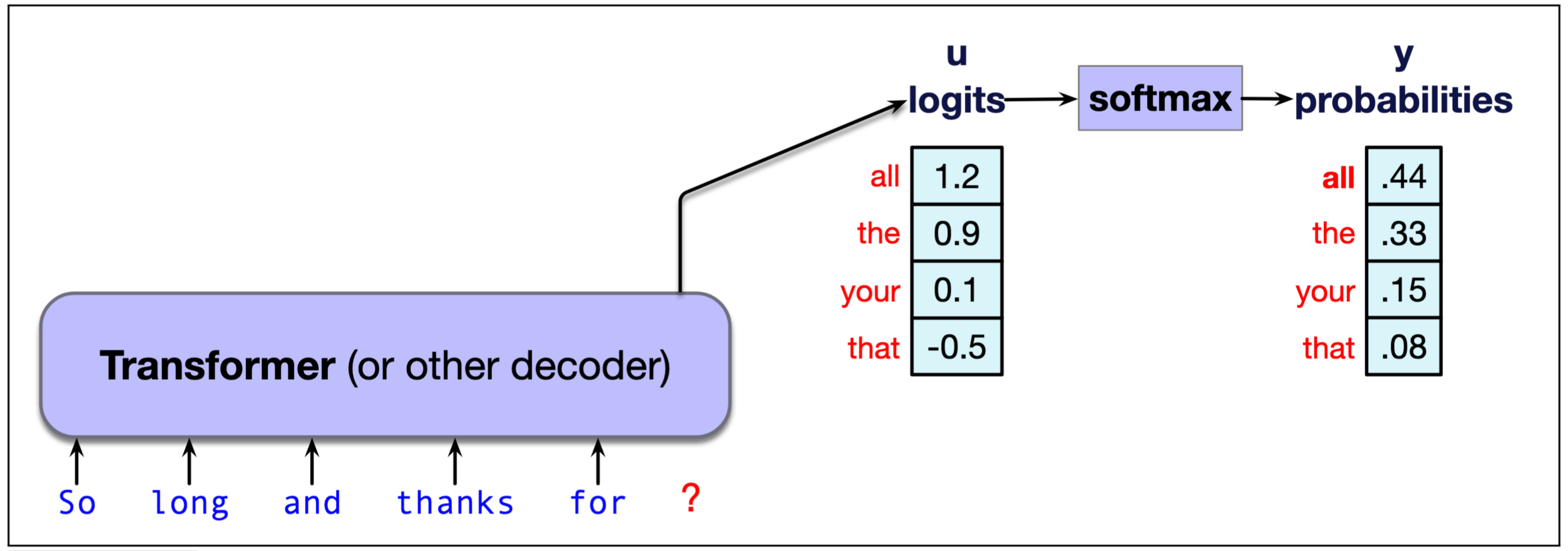


Figure 7.7 Taking the logit vector **u** and using the softmax to create a probability vector **y**.

$$\hat{w}_t = \text{argmax}_{w \in V} P(w \mid w_{<t})$$

Random Sampling

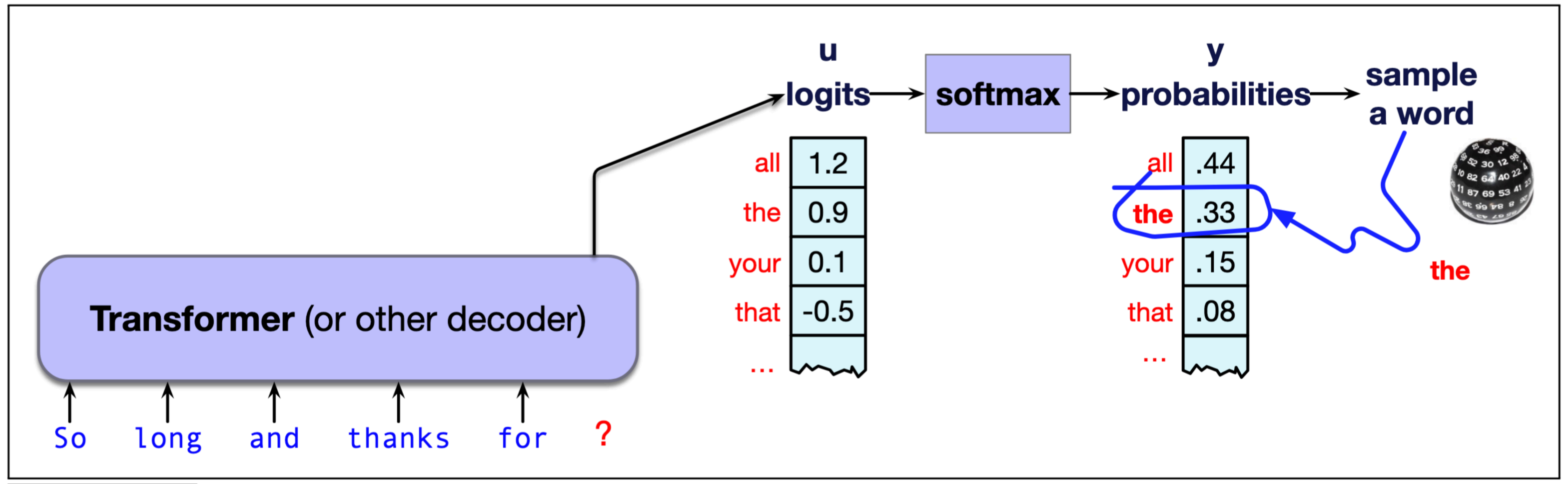


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
&\mathrm{i} \leftarrow 1 \\
&w_i \sim \mathrm{p}(w) \\
&\textbf{while } w_i \mathrel{!=} \mathrm{EOS} \\
&\quad \mathrm{i} \leftarrow \mathrm{i} + 1 \\
&\quad w_i \sim \mathrm{p}(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**

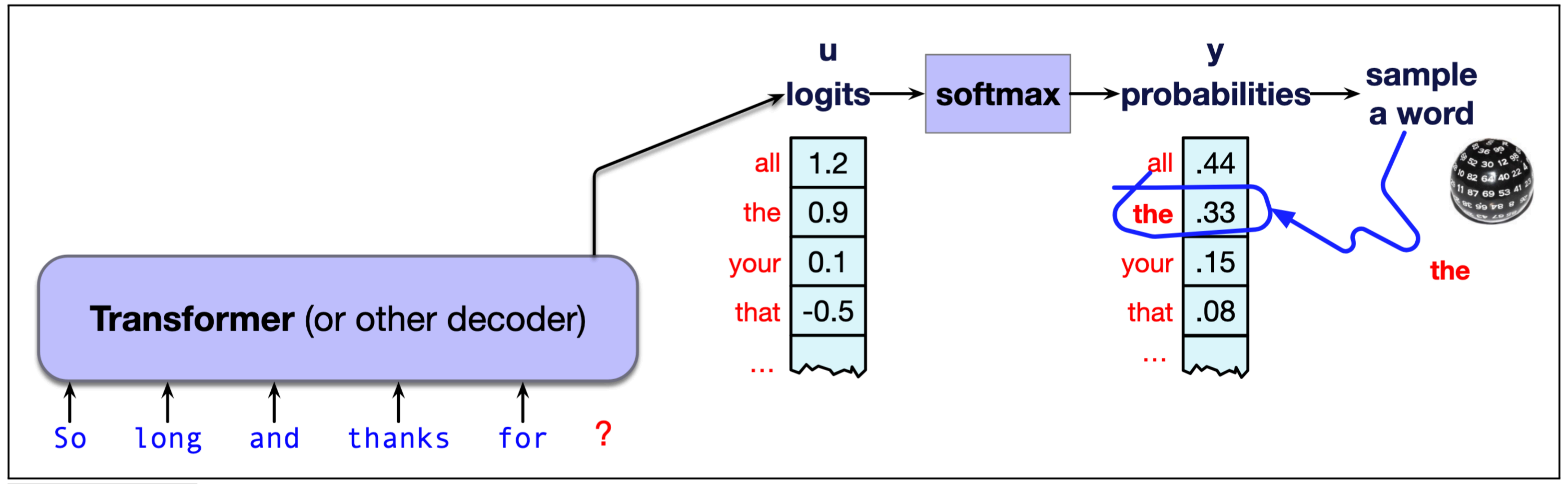


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
& \mathrm{i} \leftarrow 1 \\
& w_i \sim \mathrm{p}(w) \\
& \textbf{while } w_i \mathrel{!=} \mathrm{EOS} \\
& \quad \mathrm{i} \leftarrow \mathrm{i} + 1 \\
& \quad w_i \sim \mathrm{p}(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**
  - For LMs: sample from **distribution over possible words**

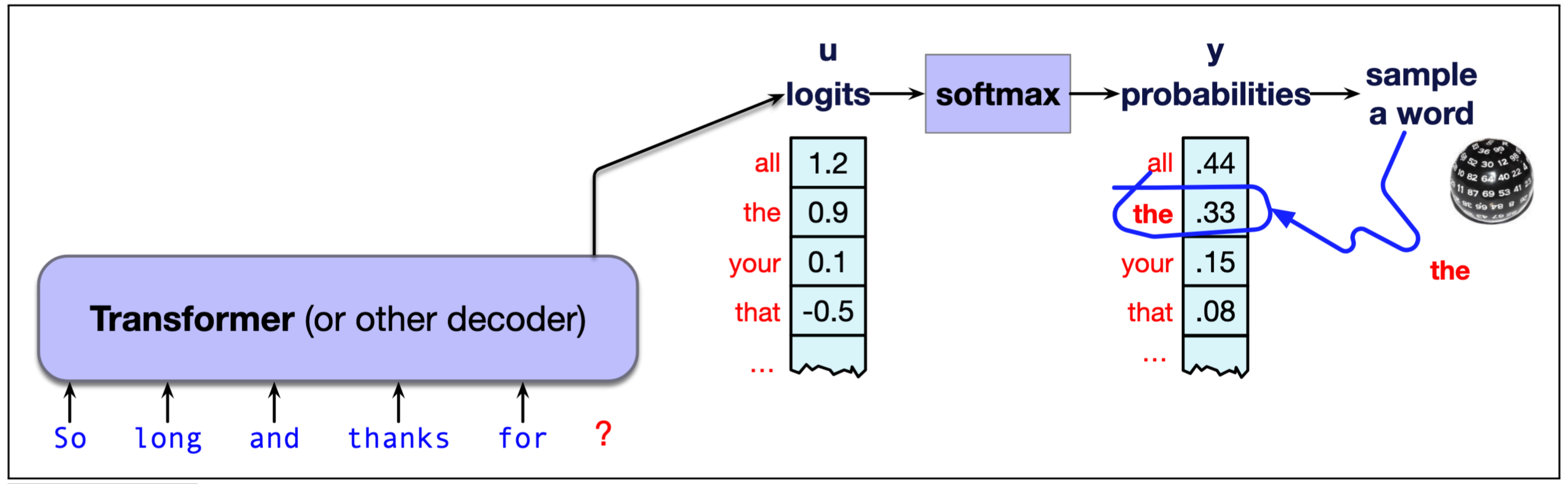


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
&\mathrm{i} \leftarrow 1 \\
&w_i \sim \mathrm{p}(w) \\
&\textbf{while } w_i \mathrel{!=} \mathrm{EOS} \\
&\quad \mathrm{i} \leftarrow \mathrm{i} + 1 \\
&\quad w_i \sim \mathrm{p}(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**
  - For LMs: sample from **distribution over possible words**
  - Stop when the **End-Of-Sequence** token is reached, or define a **maximum sequence length**

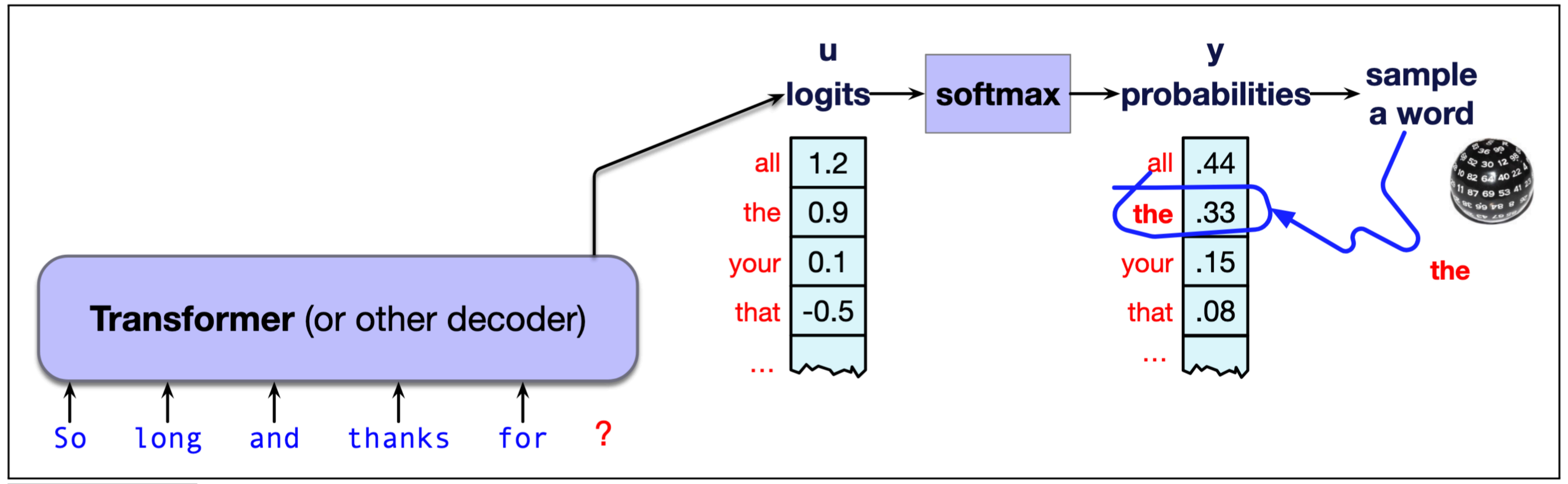


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
&\mathrm{i} \leftarrow 1 \\
&w_i \sim \mathrm{p}(w) \\
&\textbf{while } w_i \mathrel{!}= \mathrm{EOS} \\
&\quad \mathrm{i} \leftarrow \mathrm{i} + 1 \\
&\quad w_i \sim \mathrm{p}(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**
  - For LMs: sample from **distribution over possible words**
  - Stop when the **End-Of-Sequence** token is reached, or define a **maximum sequence length**
- Each word has the **probability assigned by the LM** of being generated

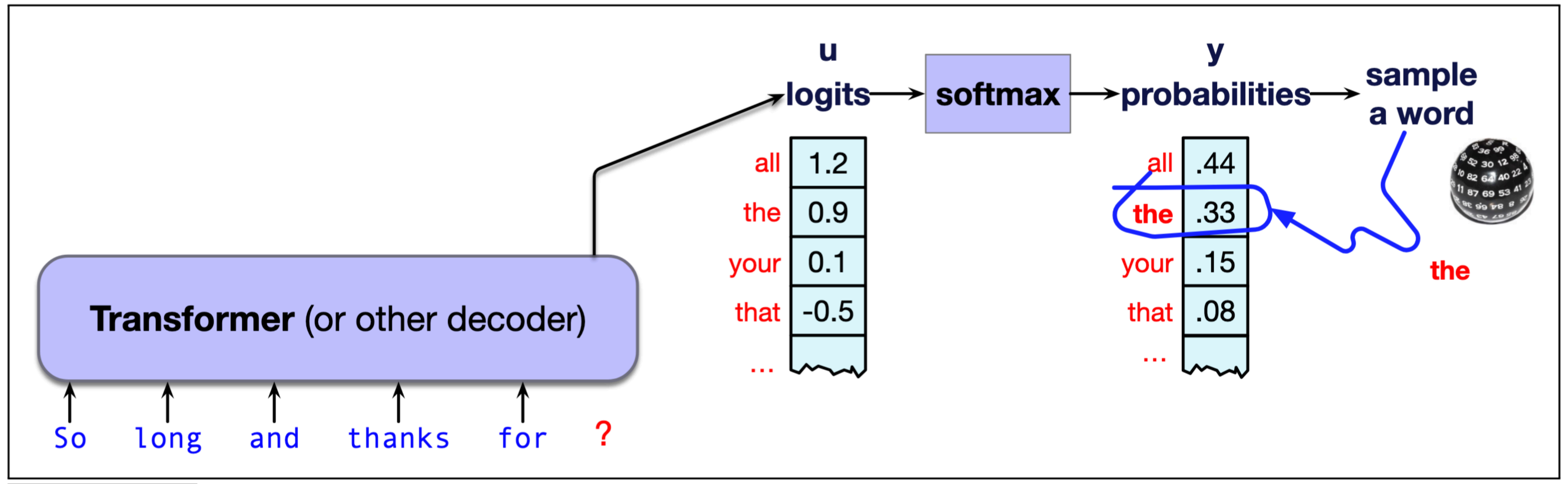


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
&\mathrm{i} \leftarrow 1 \\
&w_i \sim \mathrm{p}(w) \\
&\textbf{while } w_i \mathrel{!}= \mathrm{EOS} \\
&\quad \mathrm{i} \leftarrow \mathrm{i} + 1 \\
&\quad w_i \sim \mathrm{p}(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**
  - For LMs: sample from **distribution over possible words**
  - Stop when the **End-Of-Sequence** token is reached, or define a **maximum sequence length**
- Each word has the **probability assigned by the LM** of being generated
  - Pros: relatively **"interesting"**, novel text generation

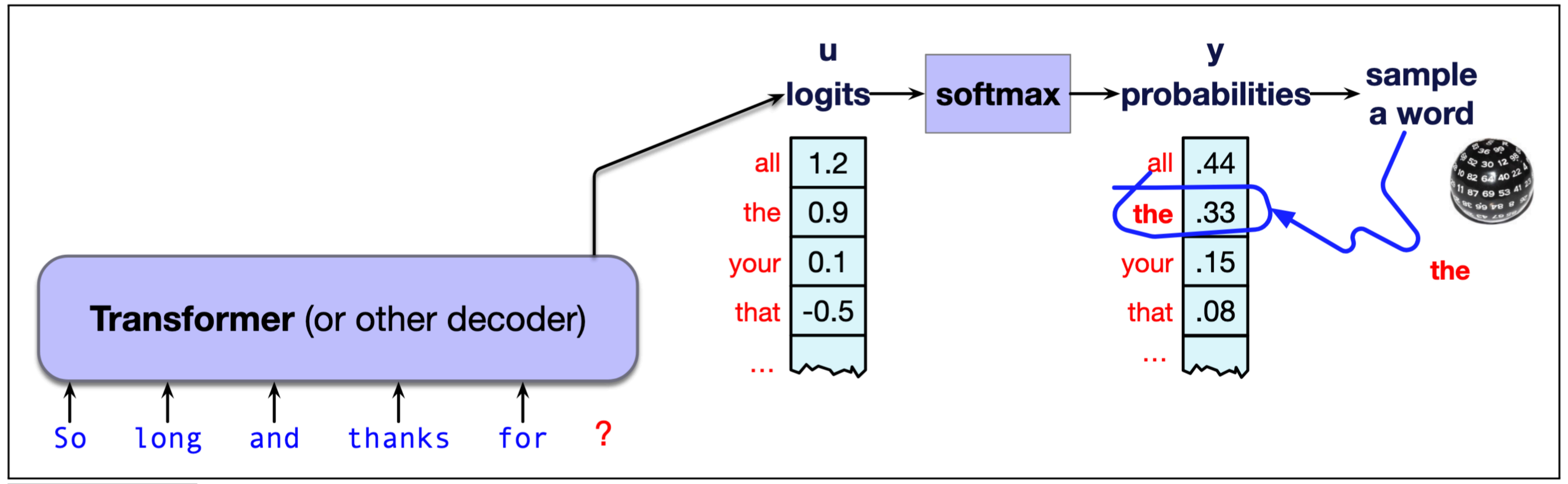


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
& \mathrm{i} \leftarrow 1 \\
& w_i \sim \mathrm{p}(w) \\
& \textbf{while } w_i \mathrel{!=} \mathrm{EOS} \\
& \quad \mathrm{i} \leftarrow \mathrm{i} + 1 \\
& \quad w_i \sim \mathrm{p}(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**
  - For LMs: sample from **distribution over possible words**
  - Stop when the **End-Of-Sequence** token is reached, or define a **maximum sequence length**
- Each word has the **probability assigned by the LM** of being generated
  - Pros: relatively **"interesting"**, novel text generation
  - Cons: high chance of **generating nonsense** (because of the **long tail** of **low-probability choices**)

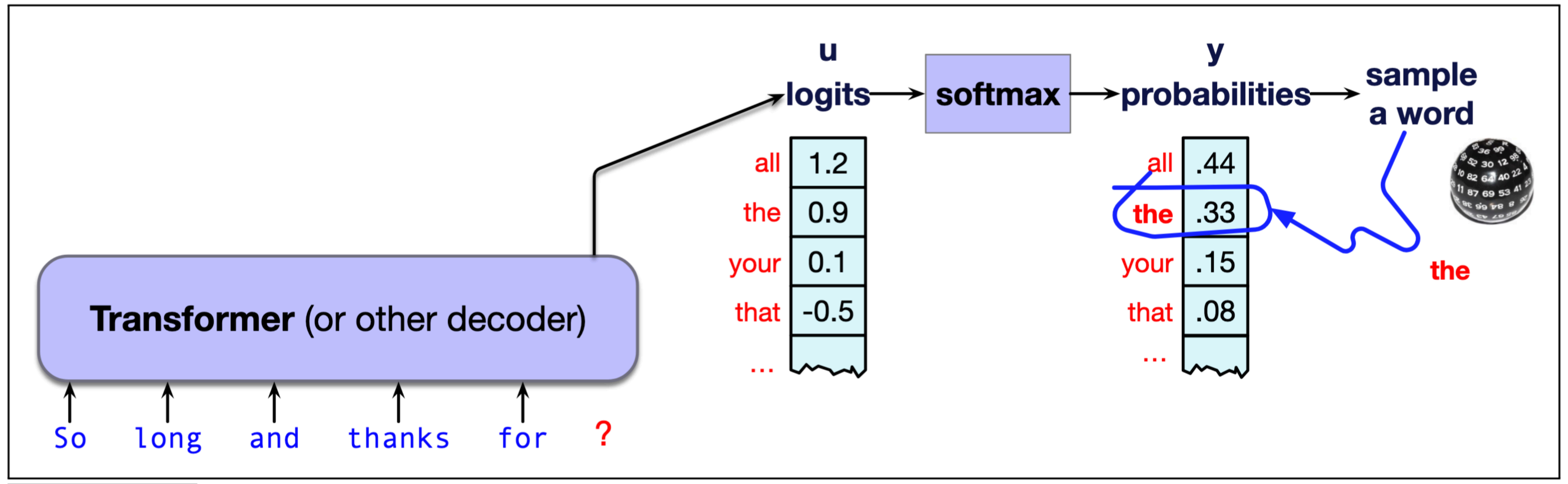


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
& i \leftarrow 1 \\
& w_i \sim \ p(w) \\
& \textbf{while } w_i \ != \text{EOS} \\
& \quad i \leftarrow i+1 \\
& \quad w_i \sim \ p(w_i \mid w_{<i})
\end{aligned}
$$

Random Sampling

- **Sampling:** taking **random draws** from a **probability distribution**
  - For LMs: sample from **distribution over possible words**
  - Stop when the **End-Of-Sequence** token is reached, or define a **maximum sequence length**
- Each word has the **probability assigned by the LM** of being generated
  - Pros: relatively **"interesting"**, novel text generation
  - Cons: high chance of **generating nonsense** (because of the **long tail** of **low-probability choices**)
- Is there something in-between this and greedy?

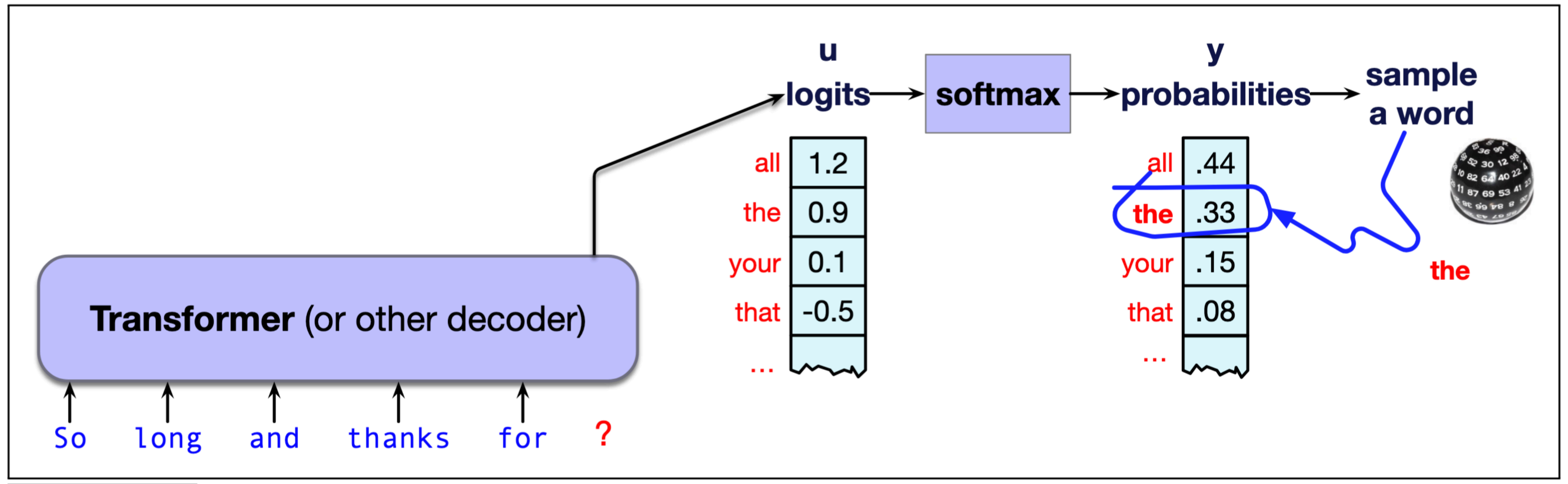
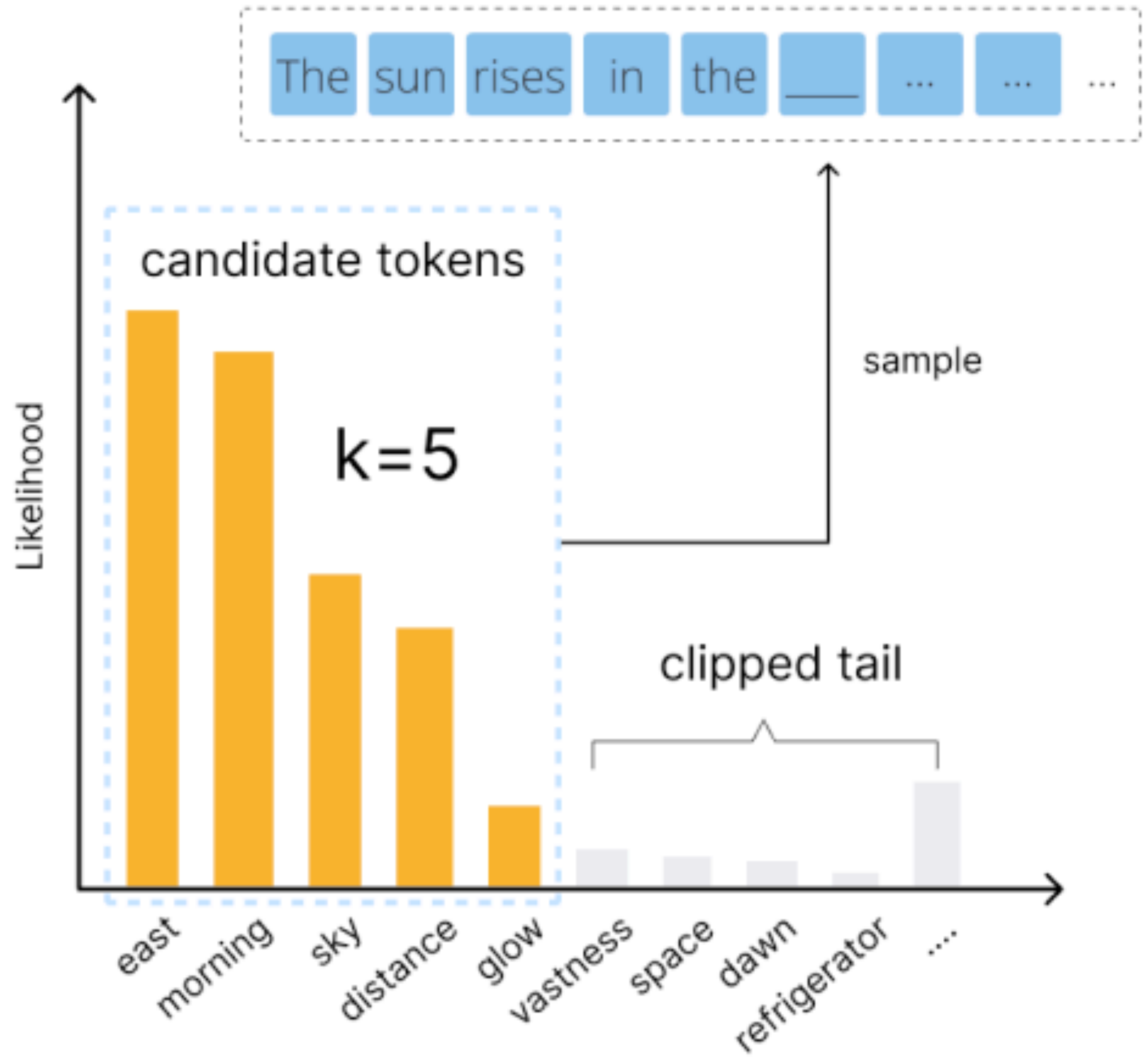


Figure 7.9 Random multinomial sampling: we randomly chose a word according to its probability.

$$
\begin{aligned}
& i \leftarrow 1 \\
& w_i \sim \; p(w) \\
& \textbf{while } w_i \; != \text{EOS} \\
& \quad i \leftarrow i + 1 \\
& \quad w_i \sim \; p(w_i \mid w_{<i})
\end{aligned}
$$

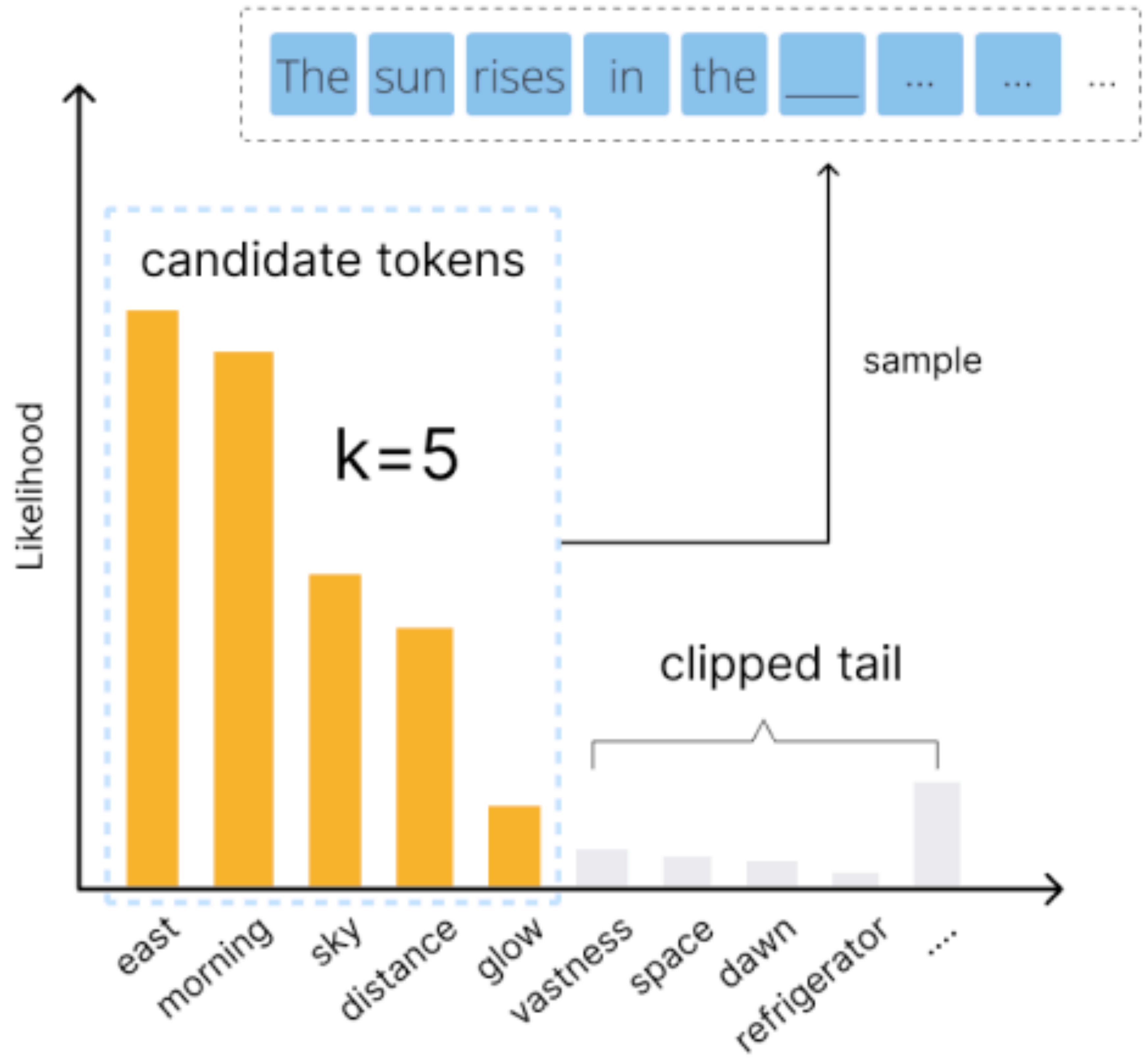
Top-k Sampling



source

Top-k Sampling

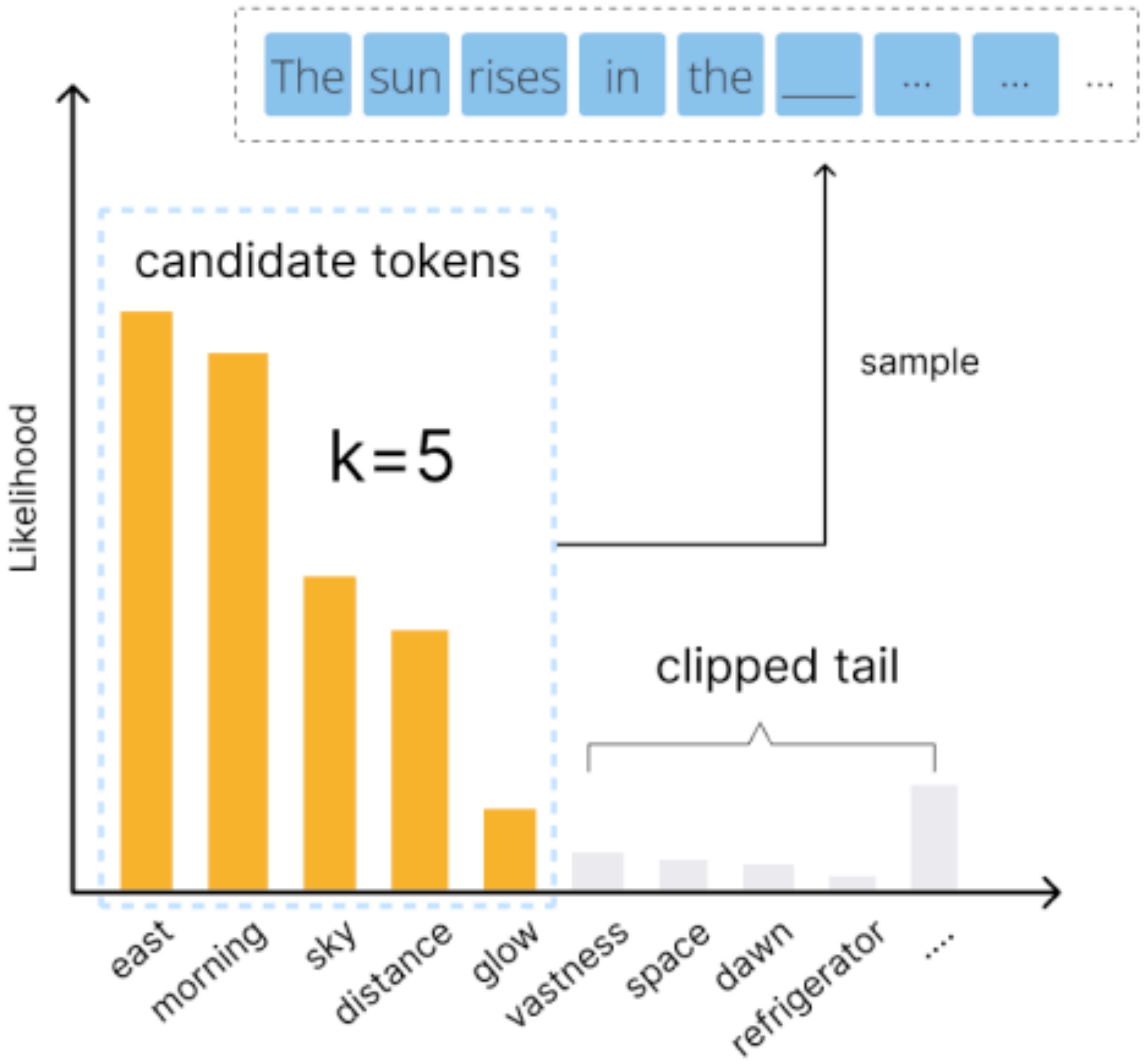
- Instead of considering the whole distribution, what about the **top few words?**



source

Top-k Sampling

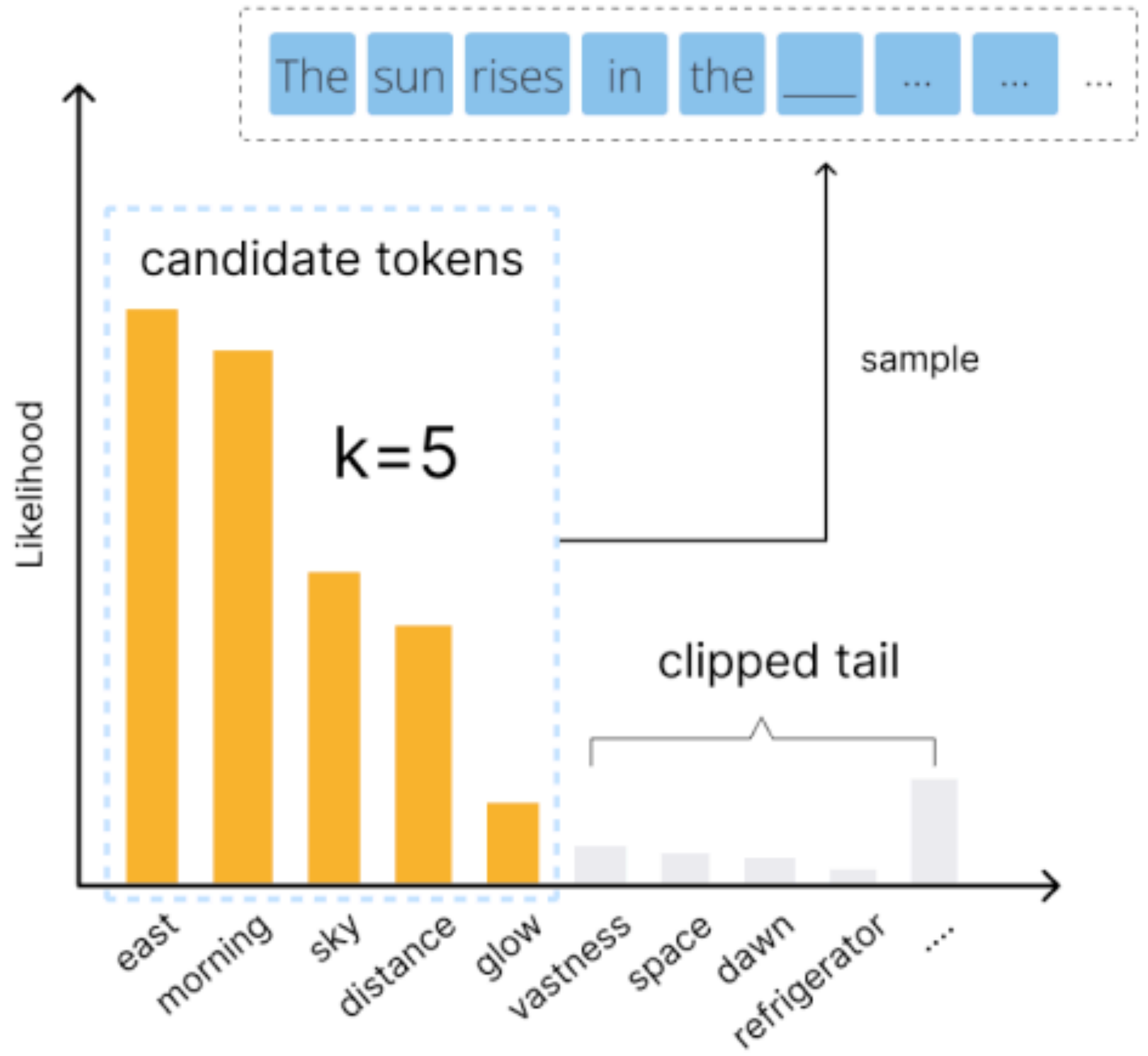
- Instead of considering the whole distribution, what about the **top few words?**
- Top-k Sampling:
  - Take the ***k* highest-probability** words
  - Sample **among these words** according to their probability



source

Top-k Sampling

- Instead of considering the whole distribution, what about the **top few words?**
- Top-k Sampling:
  - Take the ***k* highest-probability** words
  - Sample **among these words** according to their probability
- **Cuts off** the long tail of the distribution



source

Top-p Sampling

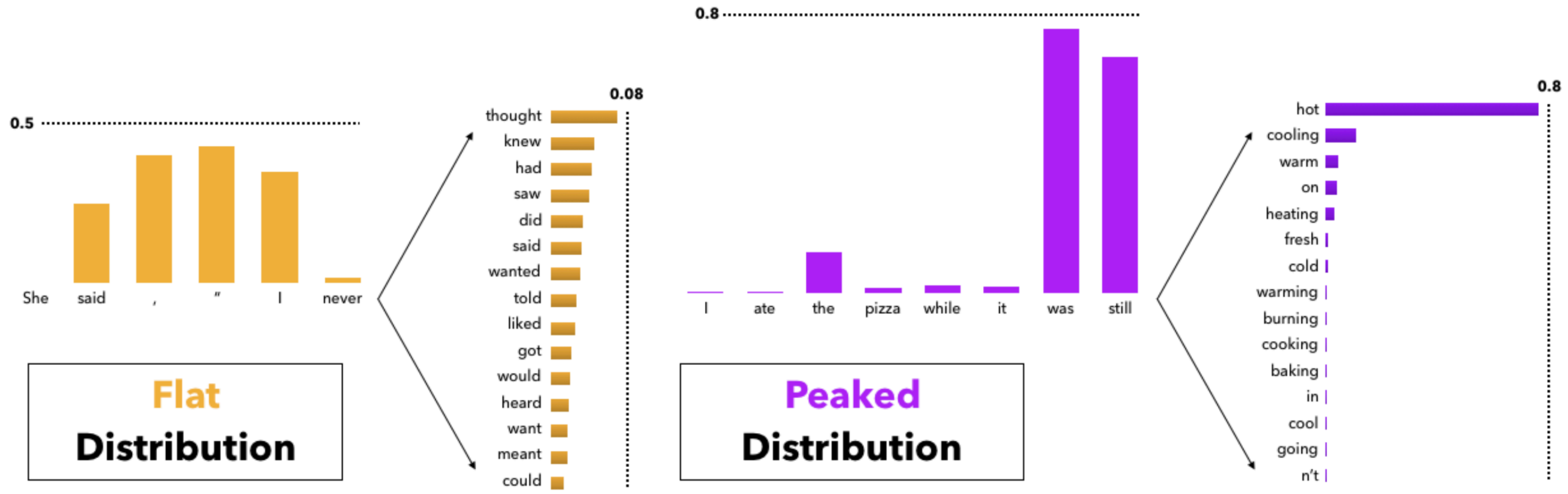


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

[Holtzman et al (2020)]

Top-p Sampling

- Problem with top-k: probability distributions can look **very different**

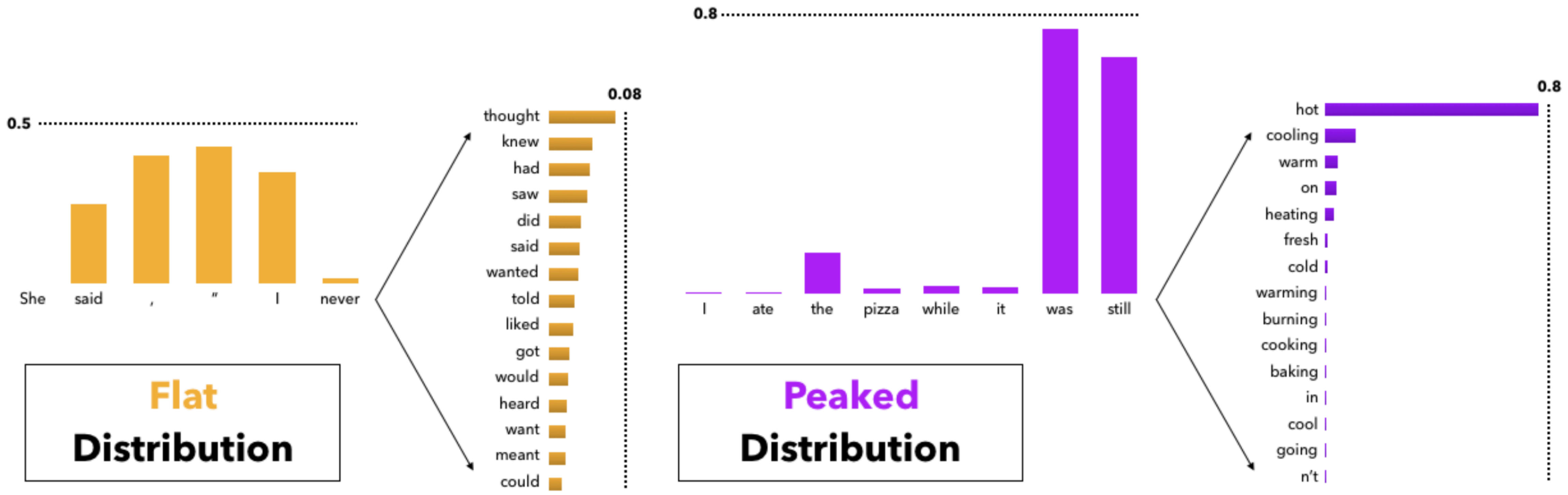


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

Holtzman et al (2020)

Top-p Sampling

- Problem with top-k: probability distributions can look **very different**
  - Sometimes the top k will make up the **majority** of the **probability mass** (peaked)

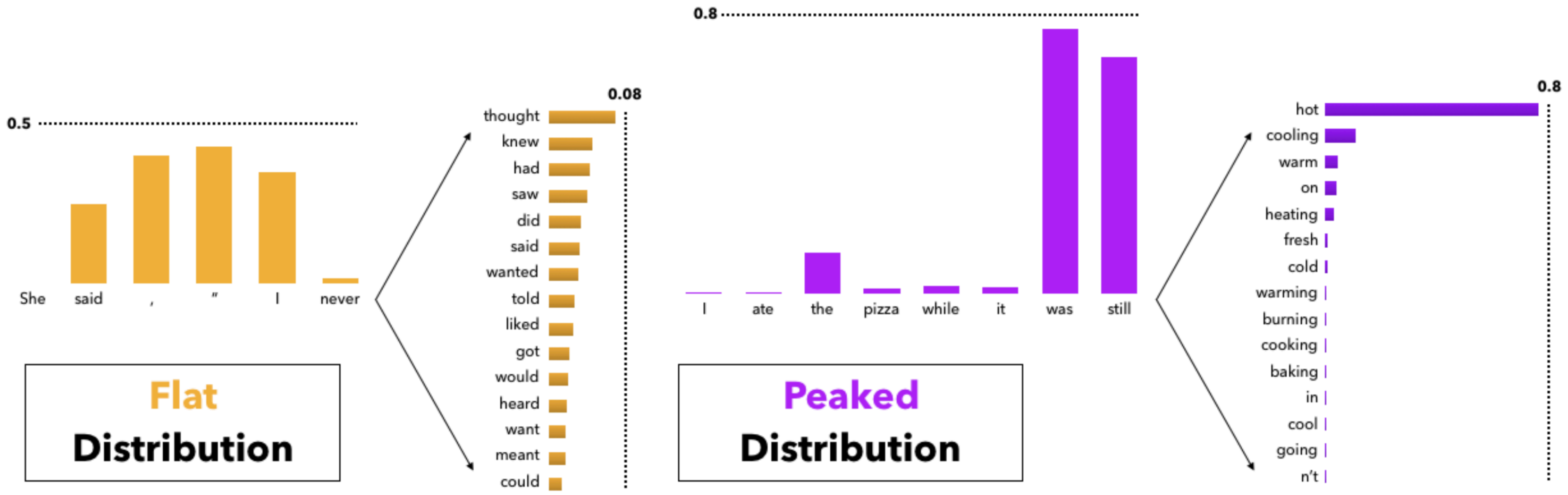


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

[Holtzman et al (2020)]

Top-p Sampling

- Problem with top-k: probability distributions can look **very different**
  - Sometimes the top k will make up the **majority** of the **probability mass** (peaked)
  - Other times the distribution is **spread out** (flat)

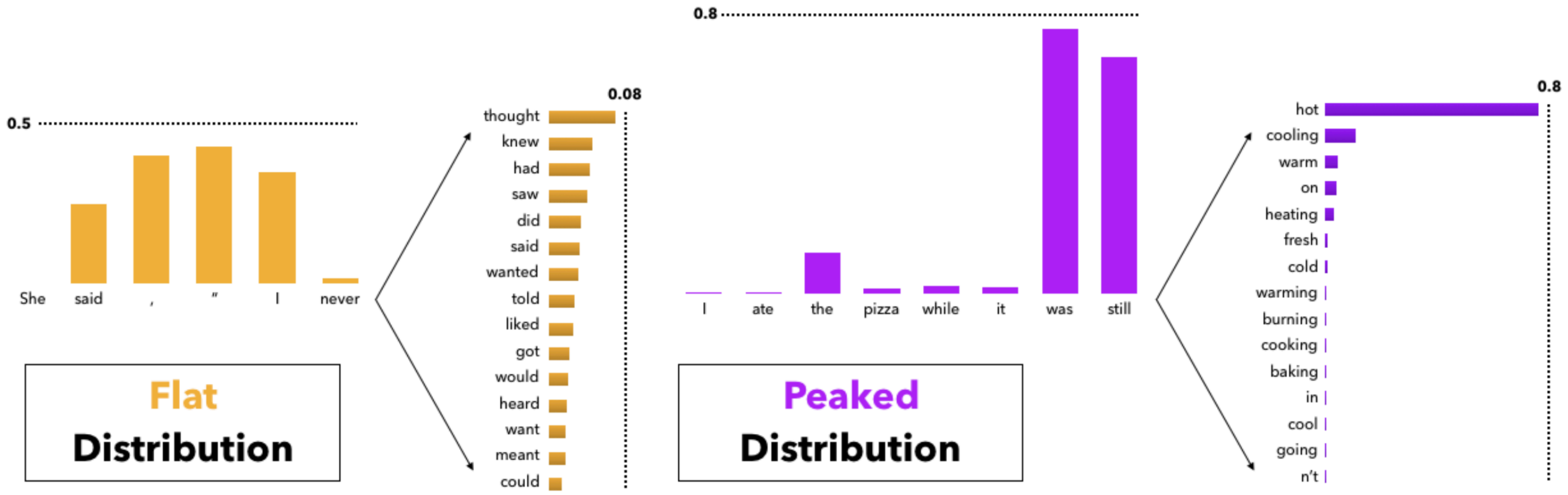


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

Holtzman et al (2020)

Top-p Sampling

- Problem with top-k: probability distributions can look **very different**
  - Sometimes the top k will make up the **majority** of the **probability mass** (peaked)
  - Other times the distribution is **spread out** (flat)
  - (Hard to find a *k* that always works well)

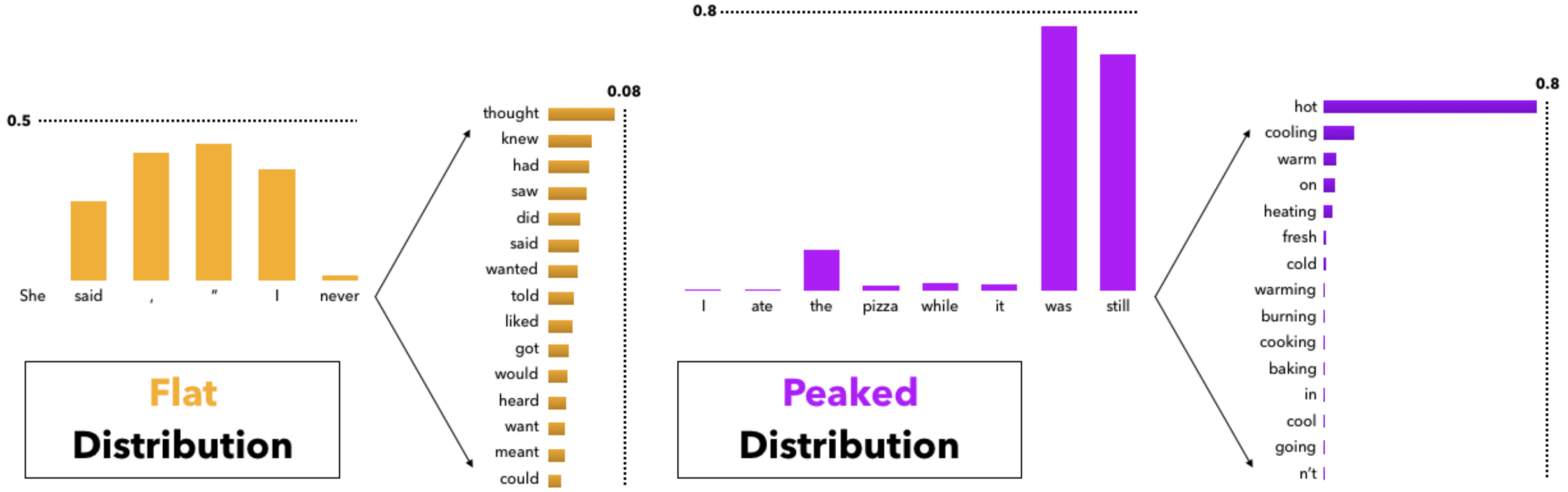


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

[Holtzman et al (2020)]

Top-p Sampling

- Problem with top-k: probability distributions can look **very different**
  - Sometimes the top k will make up the **majority** of the **probability mass** (peaked)
  - Other times the distribution is **spread out** (flat)
  - (Hard to find a *k* that always works well)
- Top-p (AKA **nucleus**) **sampling:** truncate the distribution to the **top probability mass** (e.g. 0.2 / 20%)

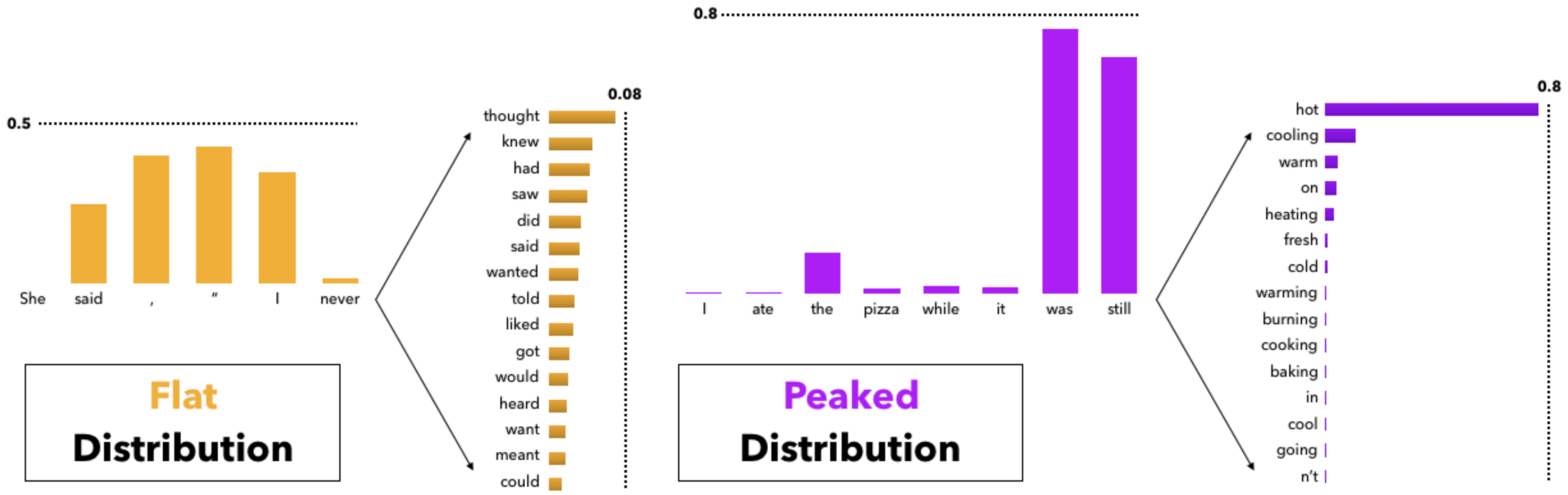


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

Holtzman et al (2020)

Top-p Sampling

- Problem with top-k: probability distributions can look **very different**
  - Sometimes the top k will make up the **majority** of the **probability mass** (peaked)
  - Other times the distribution is **spread out** (flat)
  - (Hard to find a *k* that always works well)
- Top-p (AKA **nucleus**) **sampling:** truncate the distribution to the **top probability mass** (e.g. 0.2 / 20%)
  - Adaptable to different distribution shapes

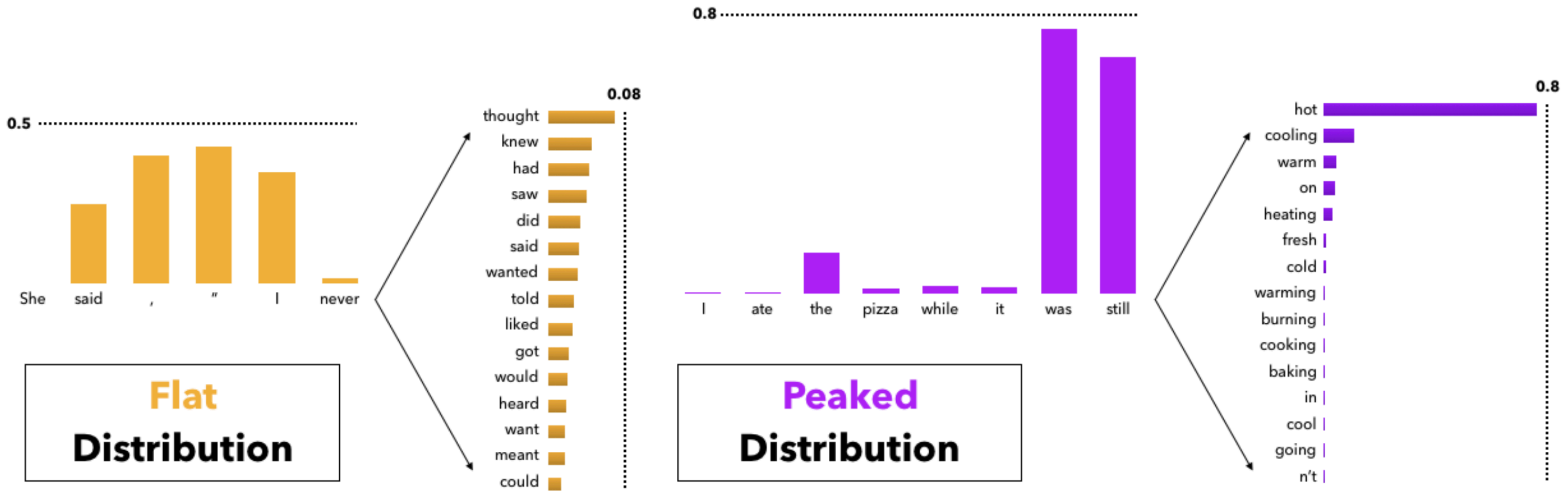


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

[Holtzman et al (2020)]

Softmax Temperature

$$\text{probs} = \text{softmax}(\mathbf{x}/\tau)$$

close to greedy

normal softmax

close to uniform

	logits	τ=0.1	τ=0.5	τ=1	τ=10	τ=100
all	1.2	.95	.59	.44	.27	.25
the	0.9	.05	.32	.33	.26	.25
your	0.1	0	.07	.15	.24	.25
that	-0.5	0	.02	.08	.23	.25

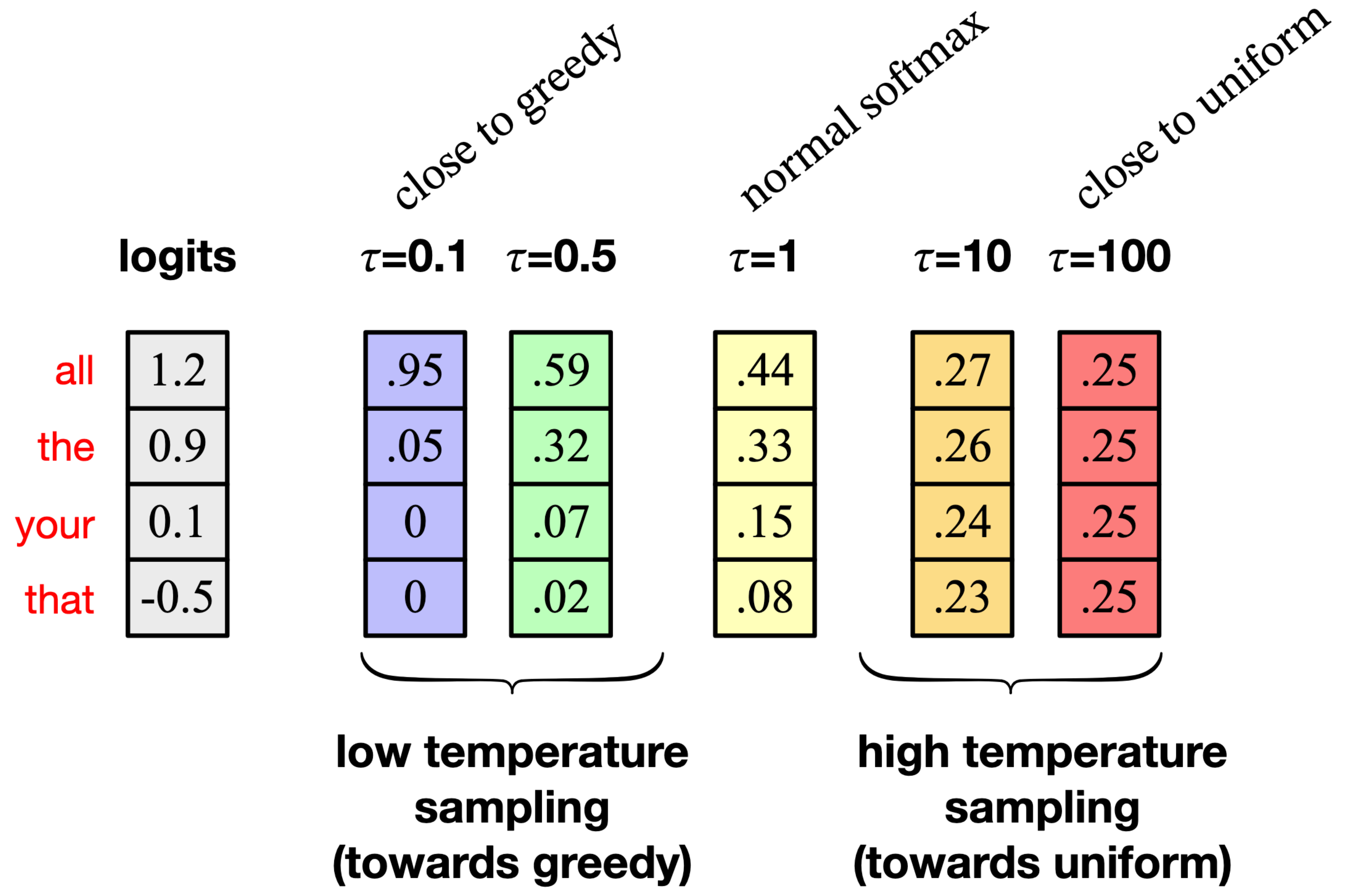
low temperature sampling (towards greedy)

high temperature sampling (towards uniform)

Softmax Temperature

- The "peakiness" of a distribution can be adjusted with parameter called **temperature** (τ)

$$\text{probs} = \text{softmax}(\mathbf{x}/\tau)$$



Softmax Temperature

- The "peakiness" of a distribution can be adjusted with parameter called **temperature** (τ)
  - **Low** temperature → **more peaky** / close to greedy sampling

$$\text{probs} = \text{softmax}(\mathbf{x}/\tau)$$

		close to greedy		normal softmax	close to uniform	
	logits	**τ=0.1**	**τ=0.5**	**τ=1**	**τ=10**	**τ=100**
all	1.2	.95	.59	.44	.27	.25
the	0.9	.05	.32	.33	.26	.25
your	0.1	0	.07	.15	.24	.25
that	-0.5	0	.02	.08	.23	.25

low temperature sampling (towards greedy)

high temperature sampling (towards uniform)

Softmax Temperature

- The "peakiness" of a distribution can be adjusted with parameter called **temperature** (τ)
  - **Low** temperature → **more peaky** / close to greedy sampling
  - **High** temperature → **more flat** / close to uniform distribution

$$\text{probs} = \text{softmax}(\mathbf{x}/\tau)$$

	logits	τ=0.1	τ=0.5	τ=1	τ=10	τ=100
		close to greedy		normal softmax		close to uniform
all	1.2	.95	.59	.44	.27	.25
the	0.9	.05	.32	.33	.26	.25
your	0.1	0	.07	.15	.24	.25
that	-0.5	0	.02	.08	.23	.25

low temperature sampling (towards greedy) (τ=0.1, τ=0.5)

high temperature sampling (towards uniform) (τ=10, τ=100)

Softmax Temperature

- The "peakiness" of a distribution can be adjusted with parameter called **temperature** (τ)
  - **Low** temperature $\rightarrow$ **more peaky** / close to greedy sampling
  - **High** temperature $\rightarrow$ **more flat** / close to uniform distribution
  - $\tau = 1.0 \rightarrow$ **regular softmax**

$$\text{probs} = \text{softmax}(\mathbf{x}/\tau)$$

		close to greedy		normal softmax		close to uniform
	logits	**τ=0.1**	**τ=0.5**	**τ=1**	**τ=10**	**τ=100**
all	1.2	.95	.59	.44	.27	.25
the	0.9	.05	.32	.33	.26	.25
your	0.1	0	.07	.15	.24	.25
that	-0.5	0	.02	.08	.23	.25

low temperature sampling (towards greedy) (τ=0.1, τ=0.5)

high temperature sampling (towards uniform) (τ=10, τ=100)

Softmax Temperature

- The "peakiness" of a distribution can be adjusted with parameter called **temperature** (τ)
  - **Low** temperature $\rightarrow$ **more peaky** / close to greedy sampling
  - **High** temperature $\rightarrow$ **more flat** / close to uniform distribution
  - $\tau = 1.0$ $\rightarrow$ **regular softmax**
- Can be **tuned** to give more/less deterministic outputs

$$\text{probs} = \text{softmax}(\mathbf{x}/\tau)$$

close to greedy — normal softmax — close to uniform

	logits	τ**=0.1**	τ**=0.5**	τ**=1**	τ**=10**	τ**=100**
all	1.2	.95	.59	.44	.27	.25
the	0.9	.05	.32	.33	.26	.25
your	0.1	0	.07	.15	.24	.25
that	-0.5	0	.02	.08	.23	.25

$\underbrace{\tau=0.1,\ \tau=0.5}$ **low temperature sampling (towards greedy)**

$\underbrace{\tau=10,\ \tau=100}$ **high temperature sampling (towards uniform)**