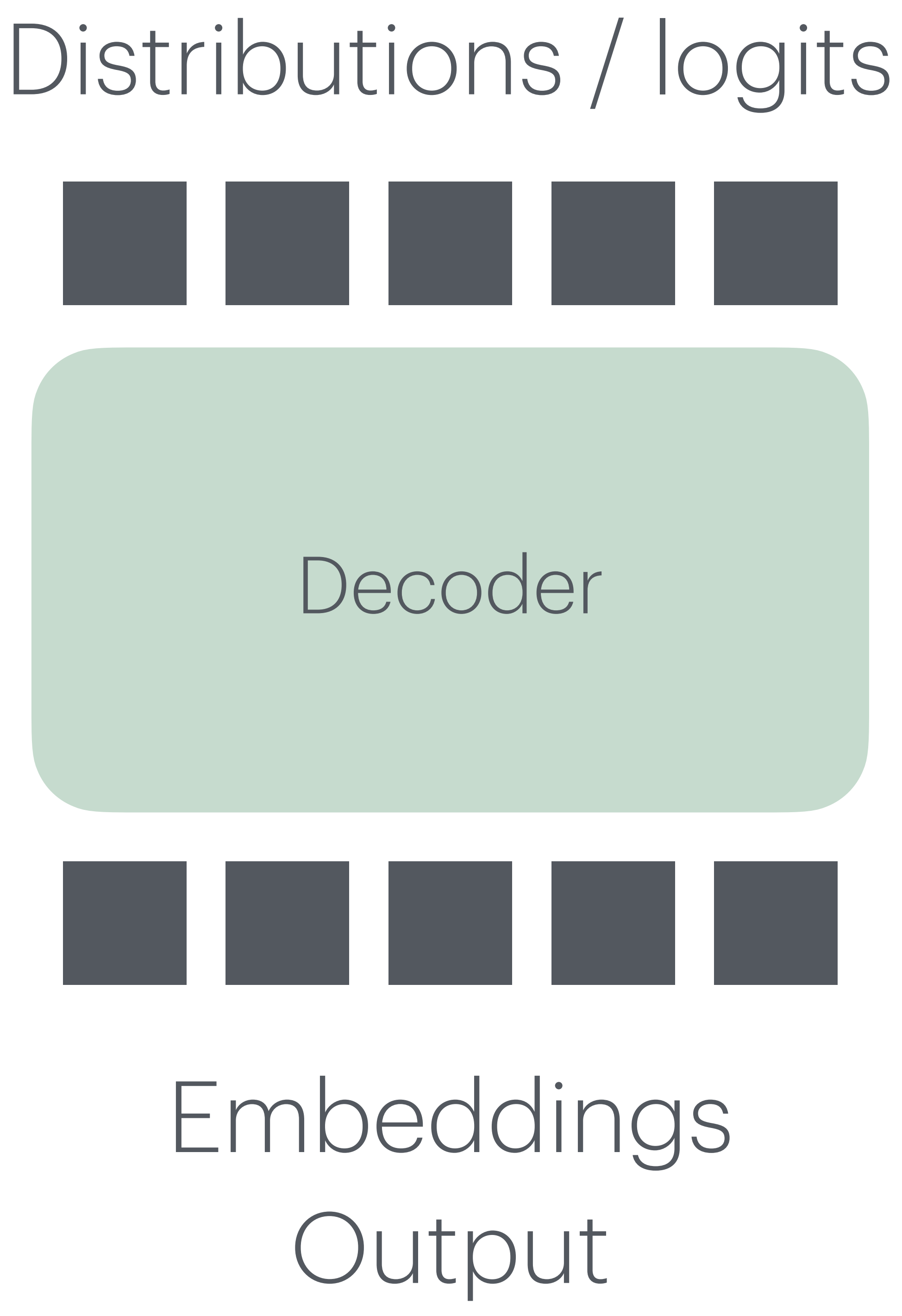
Generation

Philipp Krähenbühl, UT Austin

Language Models

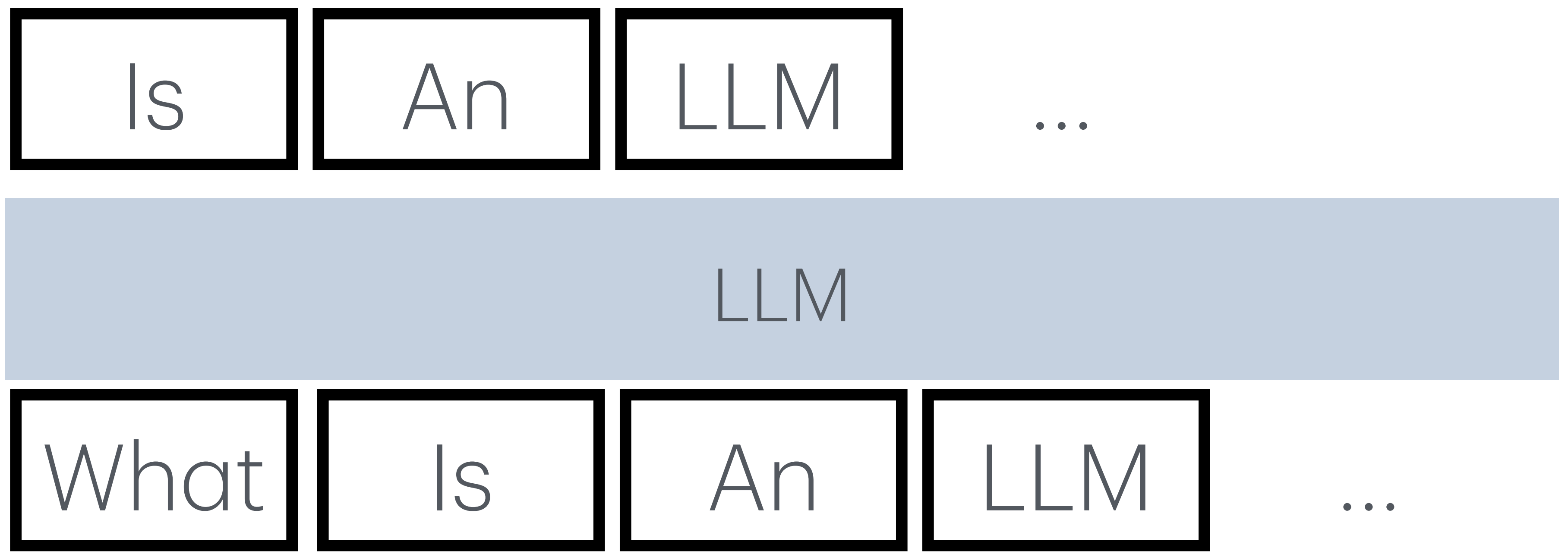
- Decoder-only LLMs
  - Modeling auto-regressive distribution over tokens
  - $P(\mathbf{t}) = P(t_1)P(t_2 \,|\, t_1)P(t_3 \,|\, t_1, t_2)P(t_4 \,|\, t_1 \ldots t_3)\ldots$



Distributions / logits

Decoder

Embeddings Output



Is An LLM ...

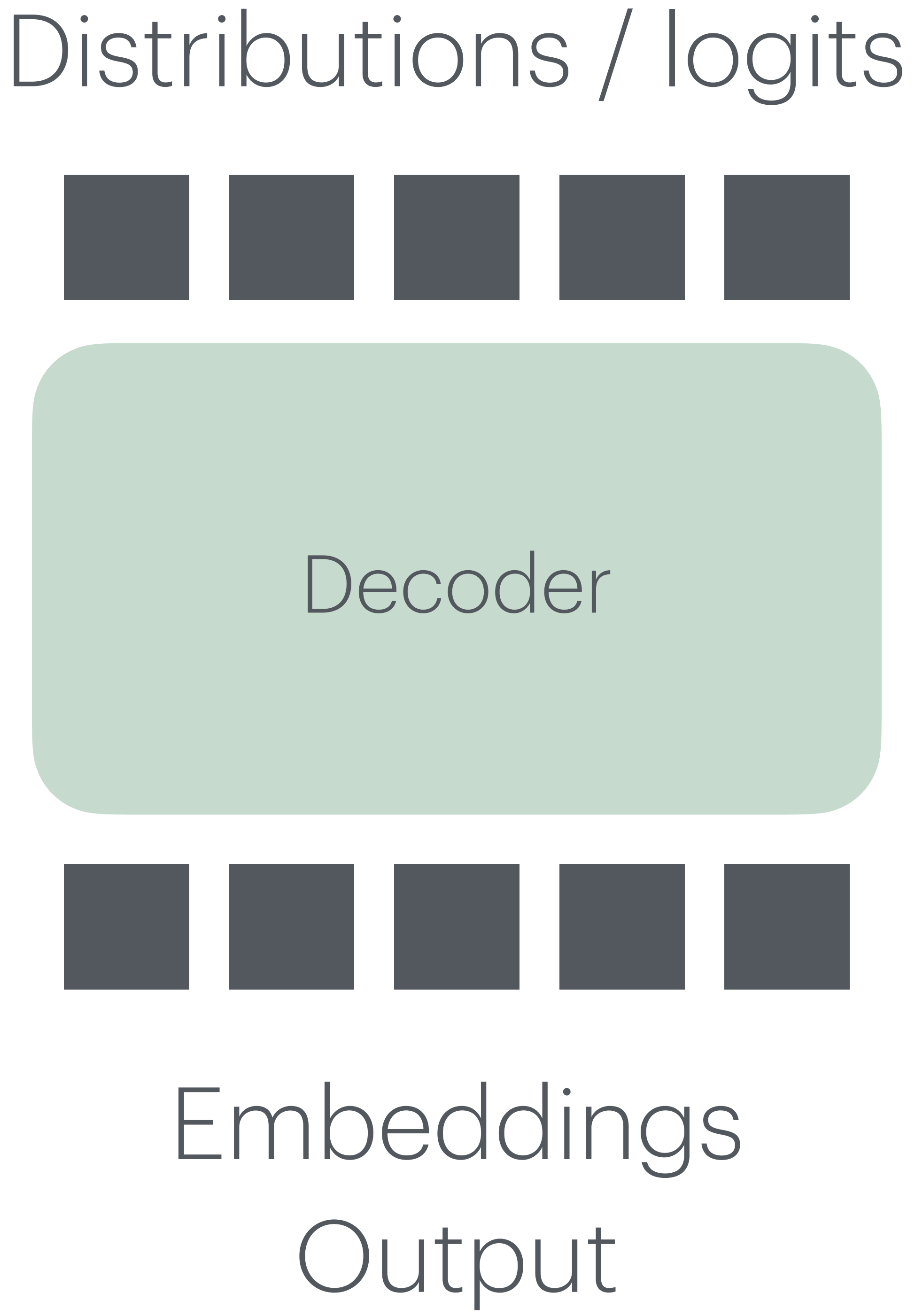
LLM

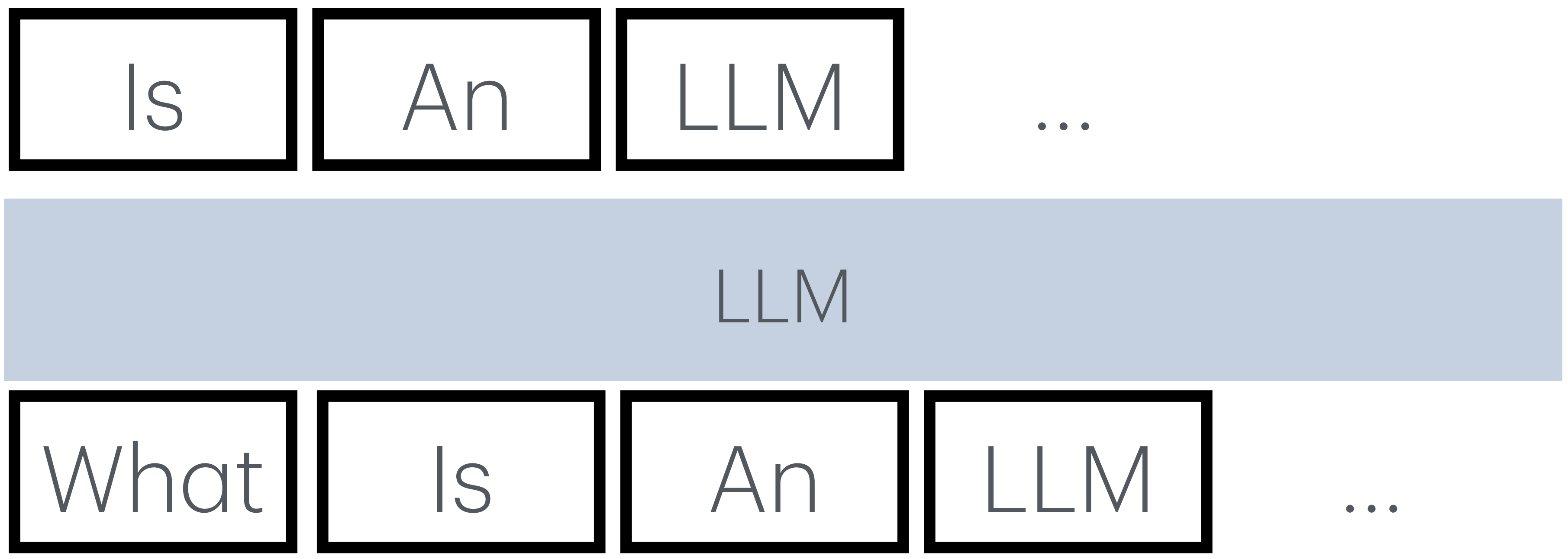
What Is An LLM ...

Language Models

Sampling

- How to generate text $\mathbf{t}$ from P

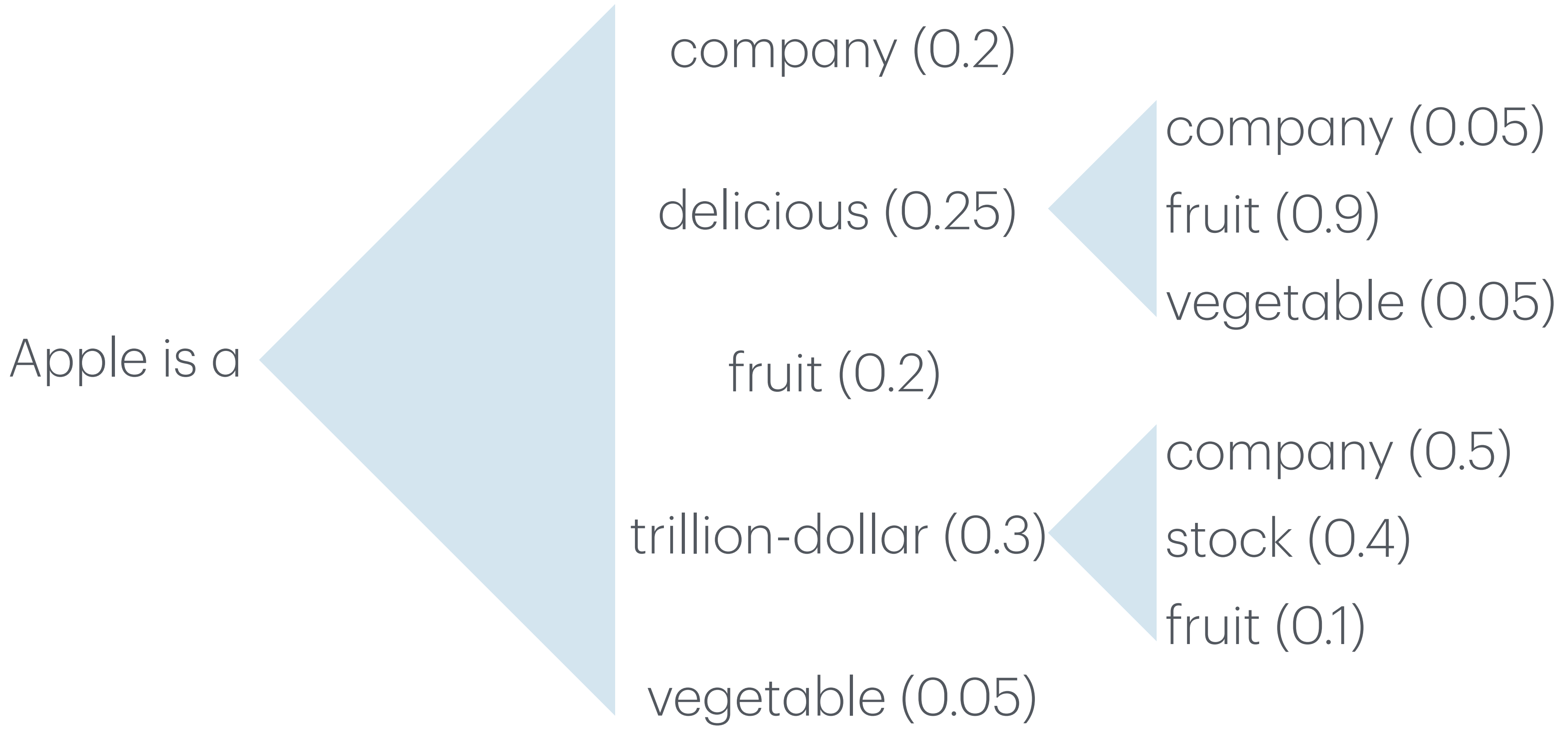




$$P(\mathbf{t}) = P(t_1)P(t_2 \mid t_1)P(t_3 \mid t_1, t_2)P(t_4 \mid t_1 \ldots t_3)\ldots$$

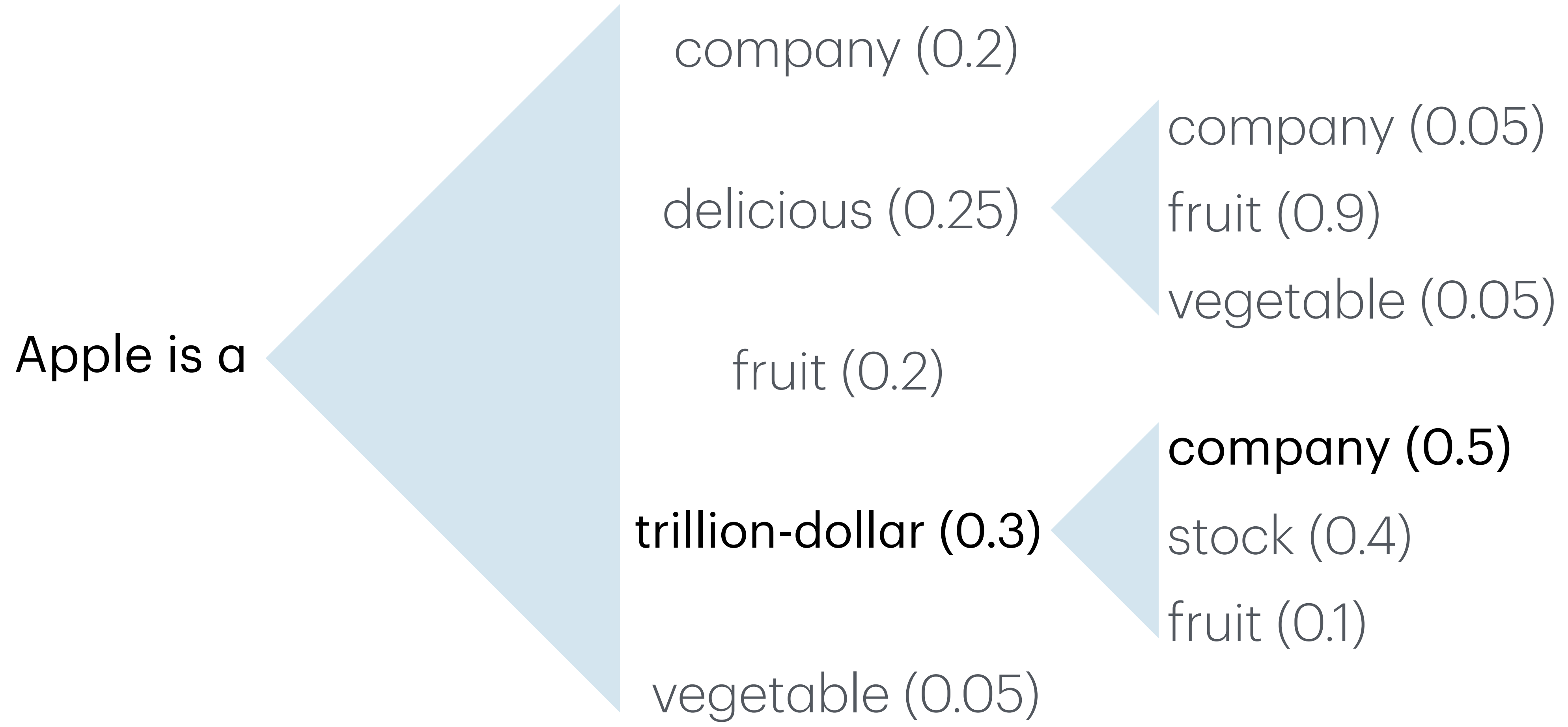
Generation

- LLM produces distribution over tokens
  - Exponentially large output space
- Tension between
  - Generalization (not assigning prob=0)
  - Fidelity (odd low-probability outputs)



Sampling - Greedy search

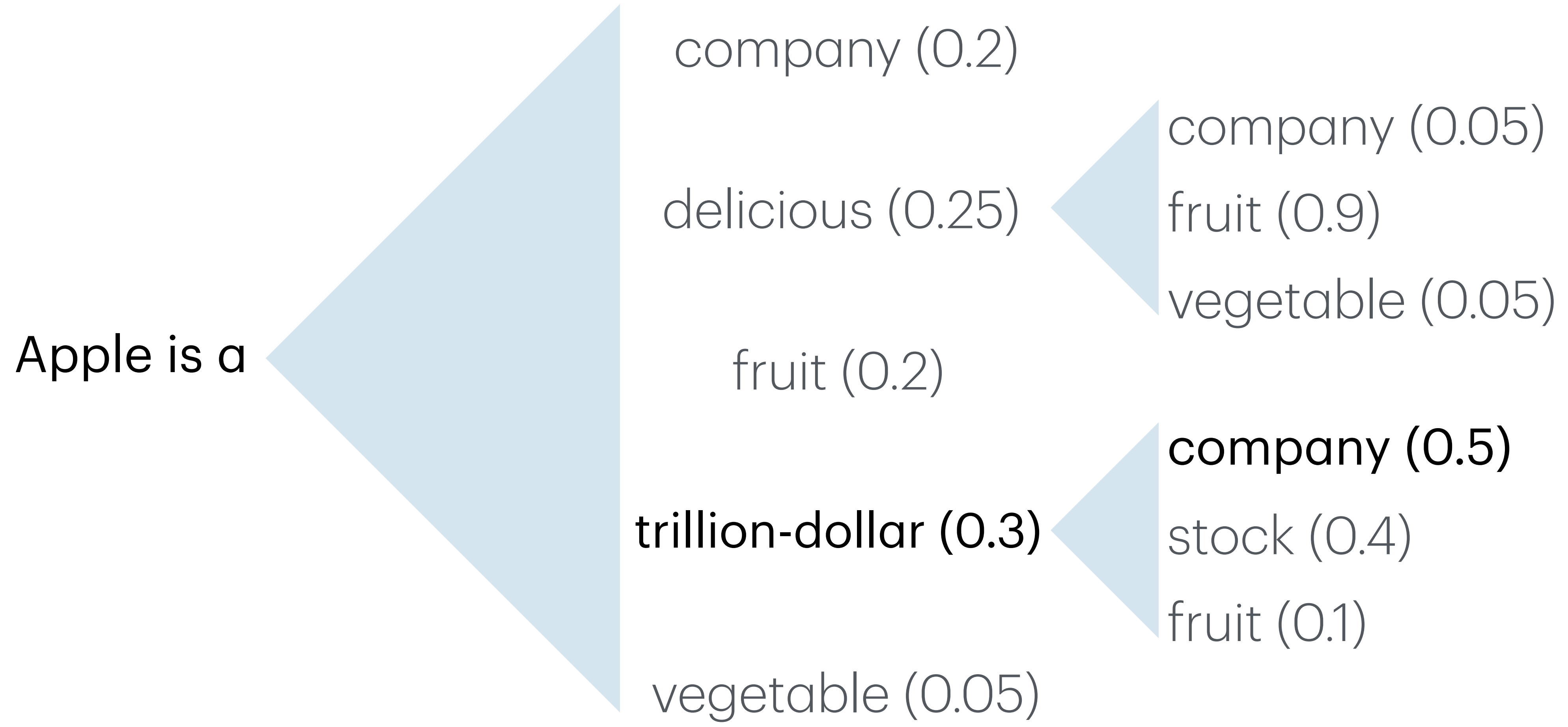
- Pick highest probability token next



Sampling - Greedy search

- A demo
  ollama run llama3.1:greedy

```
FROM llama3.1:8b-text-q4_0
PARAMETER temperature 0
PARAMETER top_k 1000
PARAMETER top_p 1.0
```


Apple is a

company (0.2)

delicious (0.25)

company (0.05)

fruit (0.9)

vegetable (0.05)

fruit (0.2)

trillion-dollar (0.3)

company (0.5)

stock (0.4)

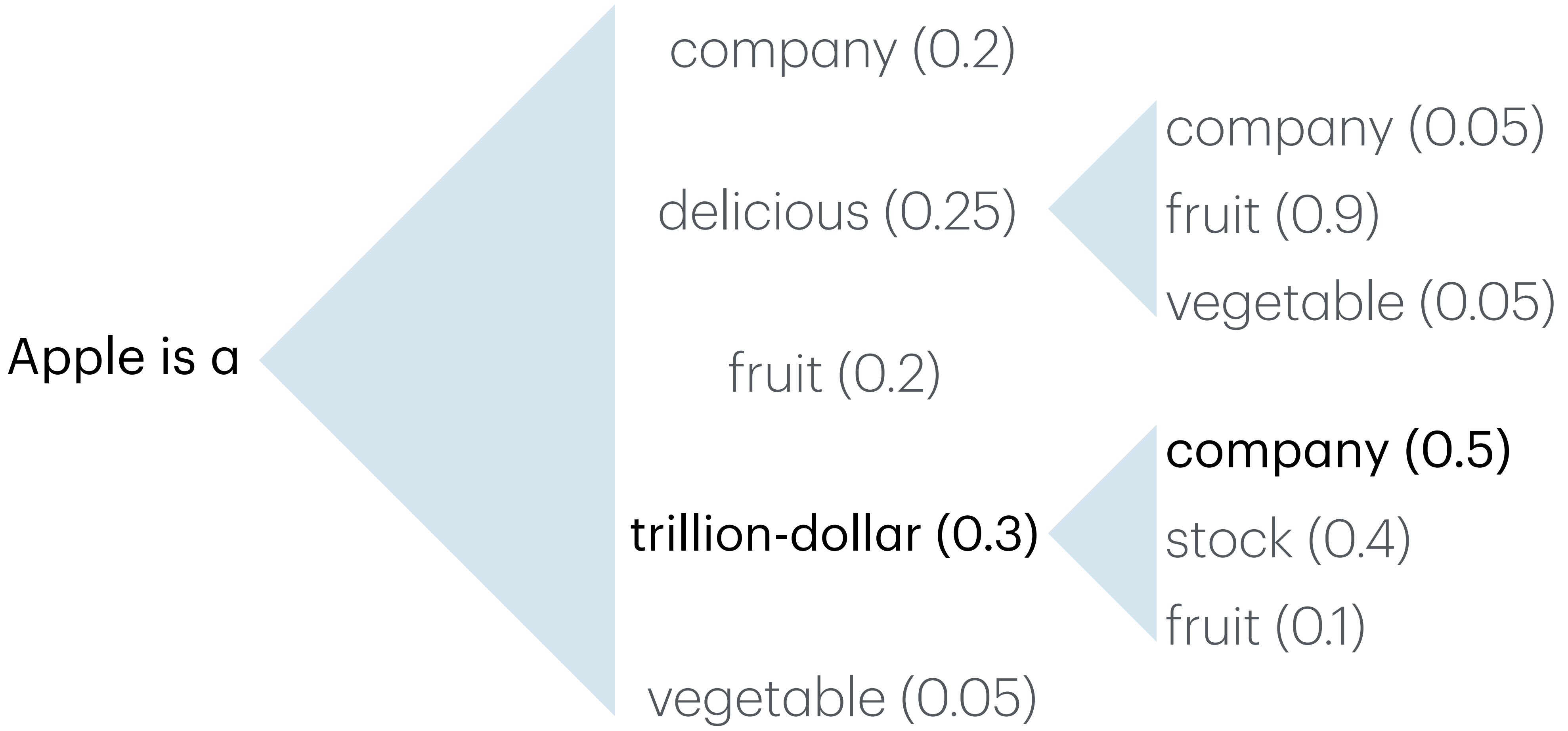
fruit (0.1)

vegetable (0.05)

Sampling - Greedy search

- Pick highest probability token next

- 😄 Super simple
- 😄 Computationally efficient
- ☹️ Single sequence
- ☹️ Bad local decisions



Apple is a

company (0.2)

delicious (0.25)

company (0.05)

fruit (0.9)

vegetable (0.05)

fruit (0.2)

trillion-dollar (0.3)

company (0.5)

stock (0.4)

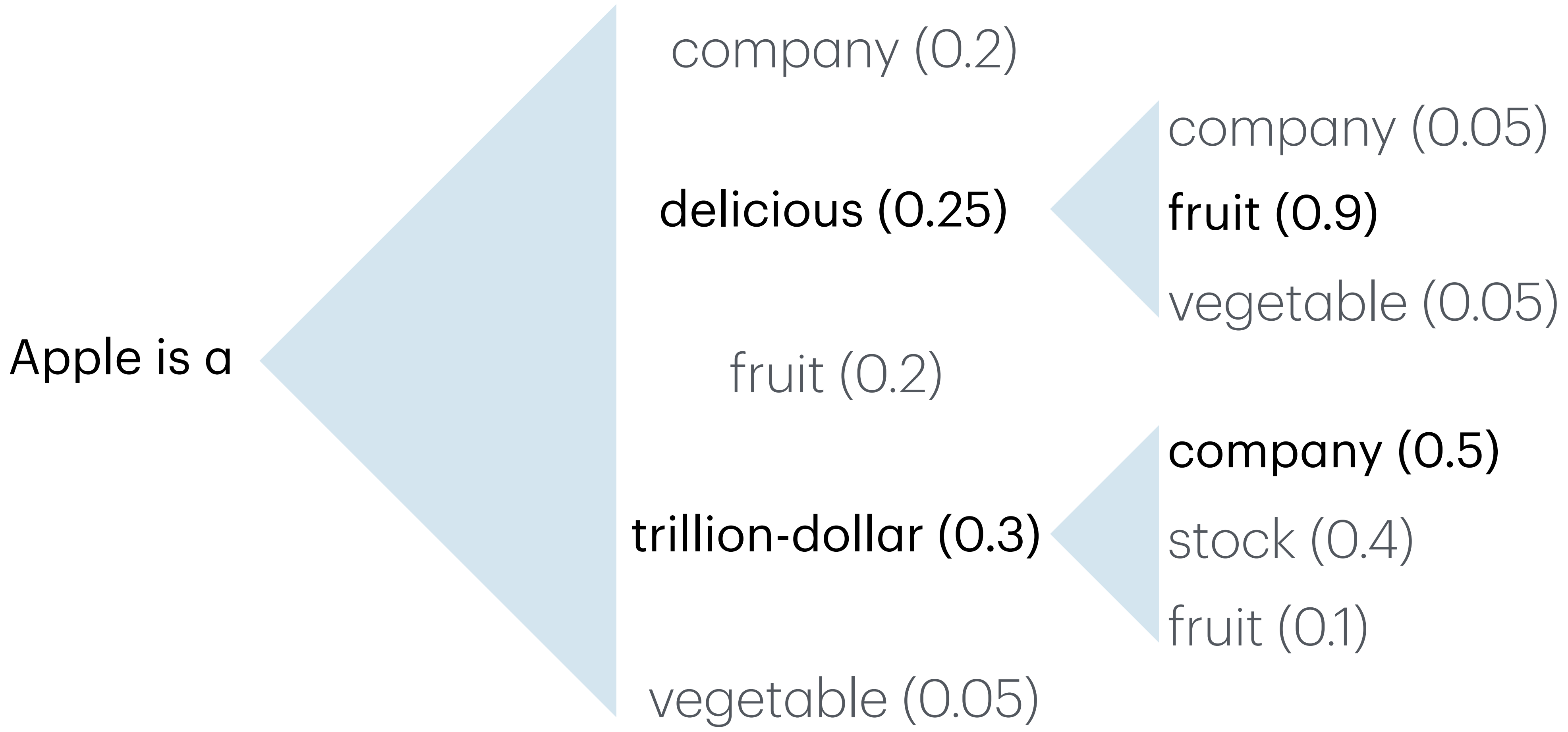
fruit (0.1)

vegetable (0.05)

Sampling - Beam search

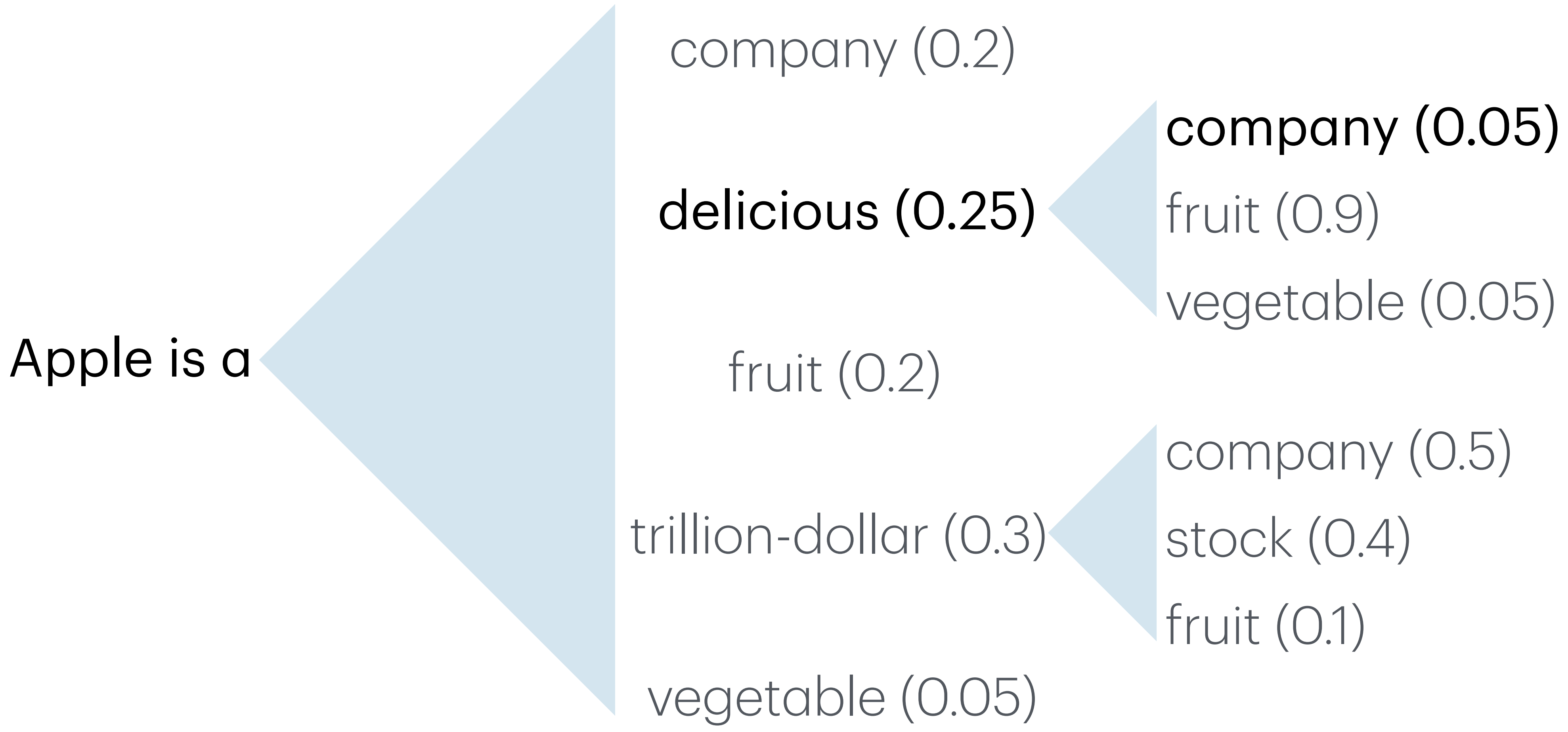
- Keep k-best samples around
  - Expand all, filter according to prob

- 😄 Good optimization
- ☹️ Computationally more expensive
- ☹️ Hard to define sampling objective



Sampling - Random sampling

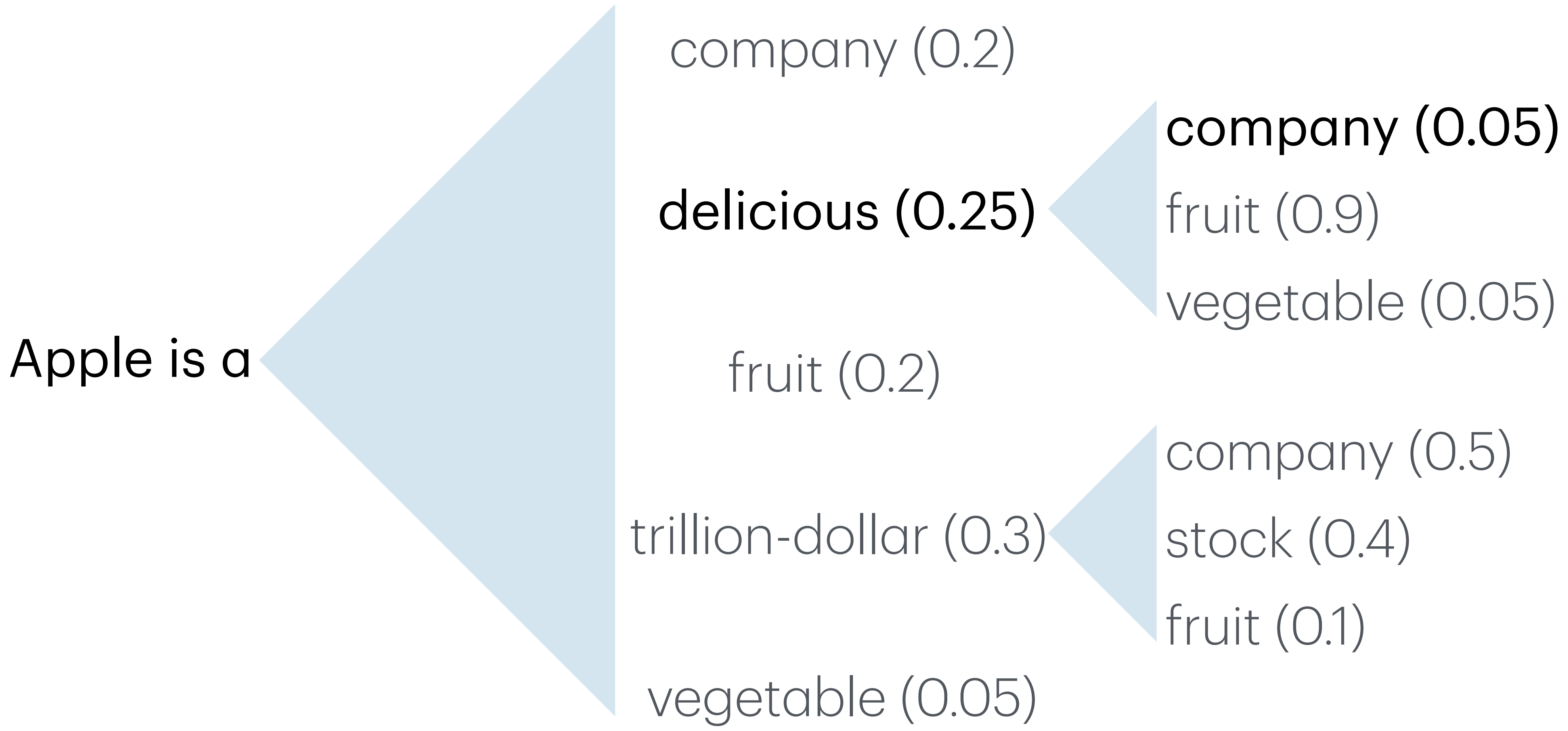
- Sample next word/token according to model distribution
  - Samples follow exponentially large model distribution



Sampling - Random sampling

- A demo
  ollama run llama3.1:temp1_random

```
FROM llama3.1:8b-text-q4_0
PARAMETER temperature 1
PARAMETER top_k 1000
PARAMETER top_p 1.0
```



Apple is a

company (0.2)

delicious (0.25)

company (0.05)
fruit (0.9)
vegetable (0.05)

fruit (0.2)

trillion-dollar (0.3)

company (0.5)
stock (0.4)
fruit (0.1)

vegetable (0.05)

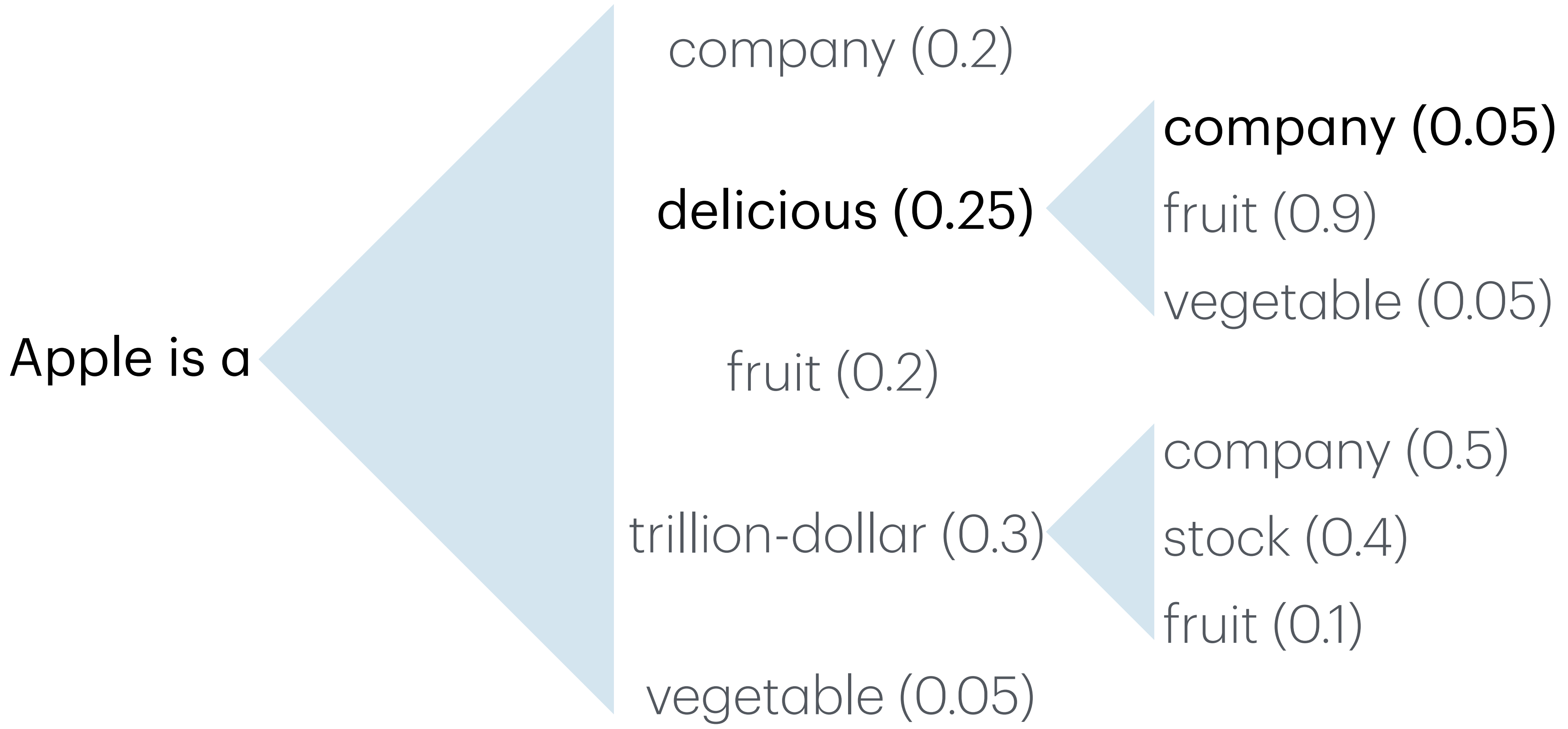
Sampling - Random sampling

- Sample next word/token according to model distribution
  - Samples follow exponentially large model distribution

- 😄 Samples sound human-like

- 😄 Computationally efficient

- ☹️ Sampling low-prob transitions



Apple is a

company (0.2)

delicious (0.25)

company (0.05)

fruit (0.9)

vegetable (0.05)

fruit (0.2)

trillion-dollar (0.3)

company (0.5)

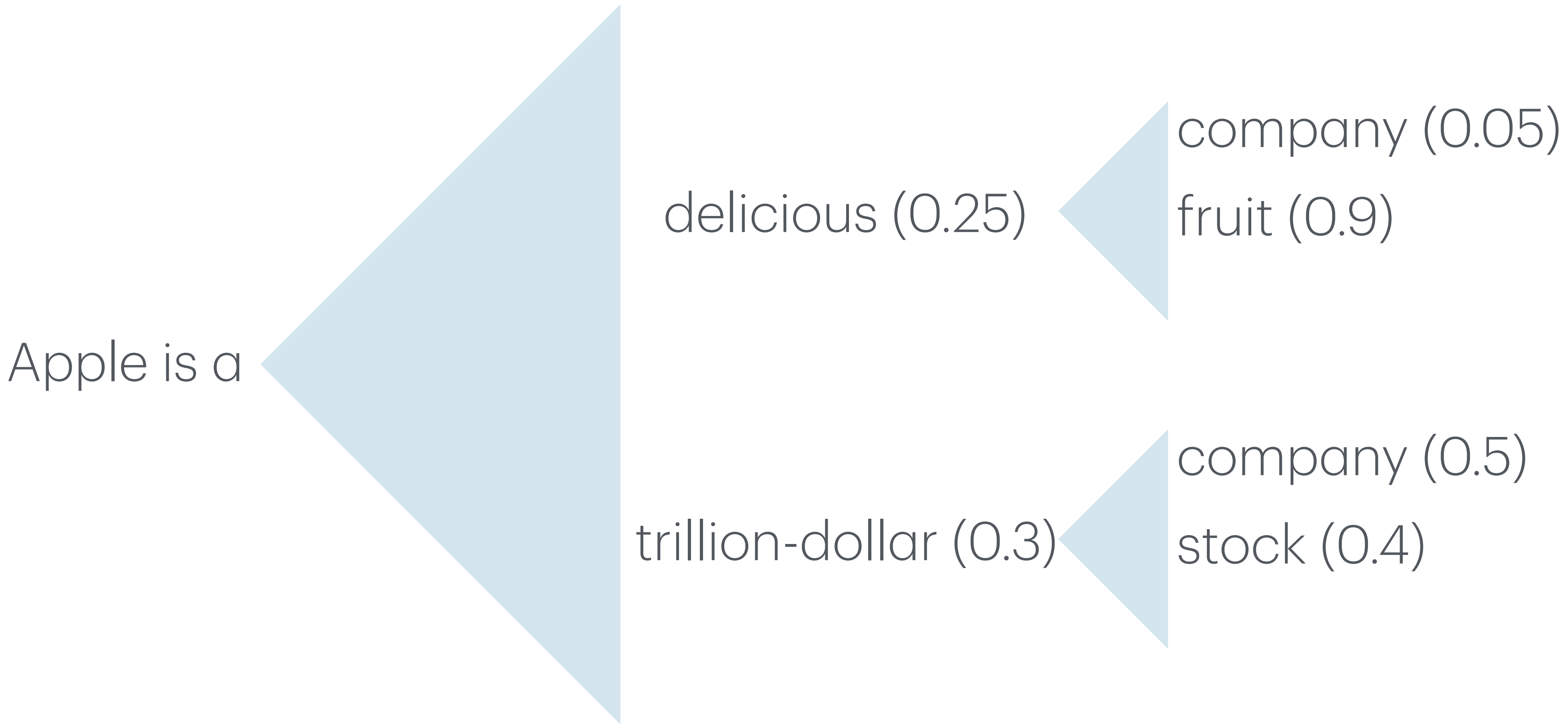
stock (0.4)

fruit (0.1)

vegetable (0.05)

Sampling - Top-K sampling

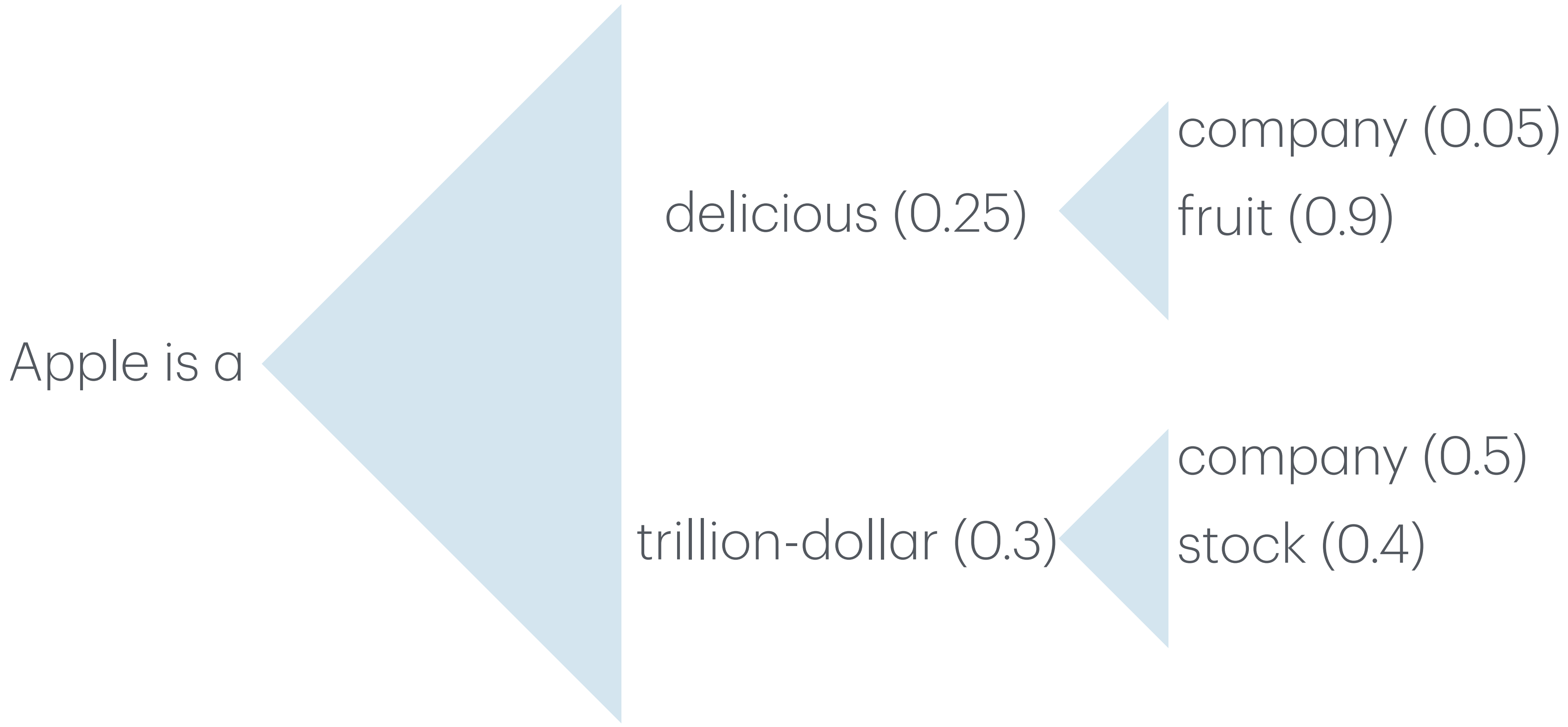
- Random sampling
  - Only consider k-most likely options



Improving Language Understanding by Generative Pre-Training. Radford et al. 2018.

Sampling - Top-K sampling

- A demo
  ollama run llama3.1:top10



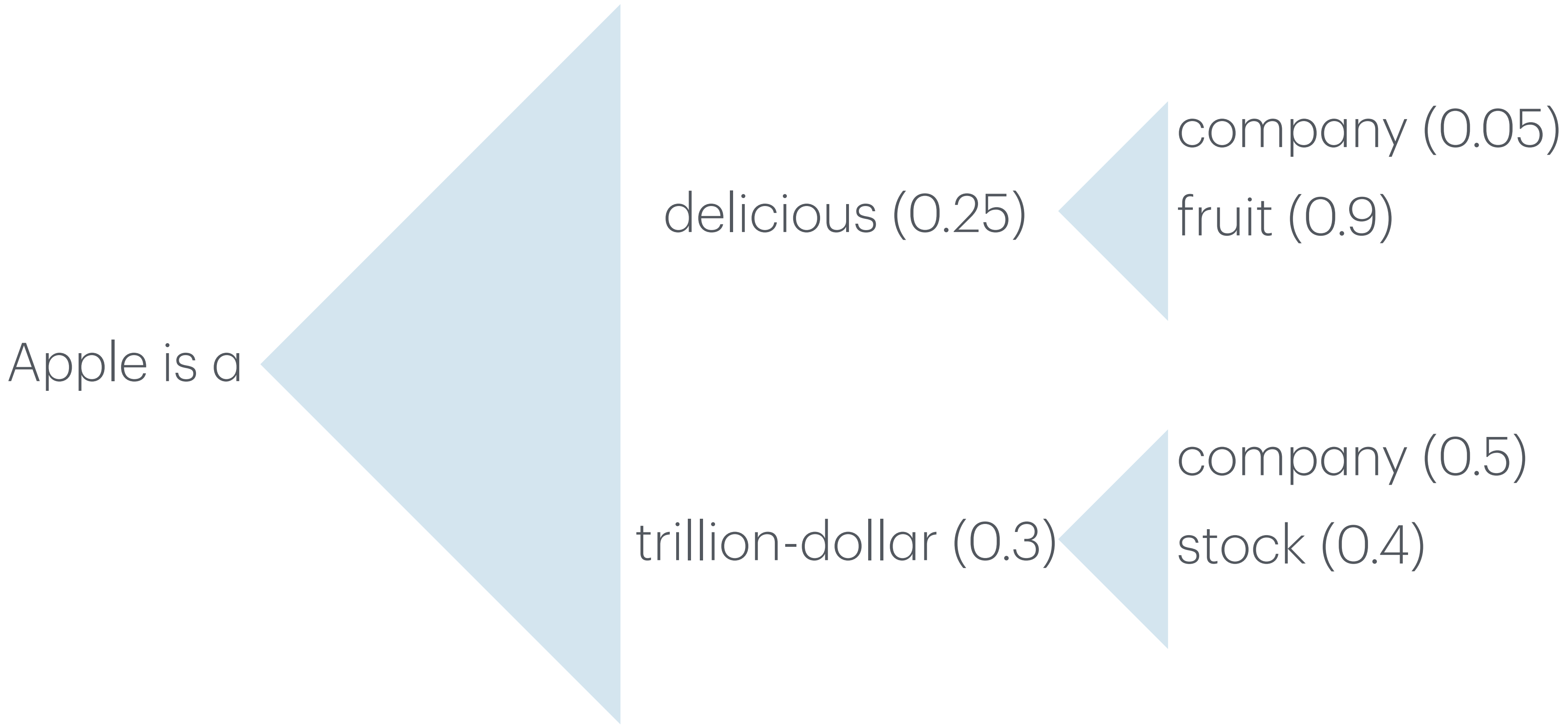
```
FROM llama3.1:8b-text-q4_0
PARAMETER temperature 1
PARAMETER top_k 10
PARAMETER top_p 1.0
```

Improving Language Understanding by Generative Pre-Training. Radford et al. 2018.

Sampling - Top-K sampling

- Random sampling
  - Only consider k-most likely options

- 😄 Samples sound human-like
- 😄 Sampling fewer low-prob transitions
- ☹️ k is hard to set (context dependent)

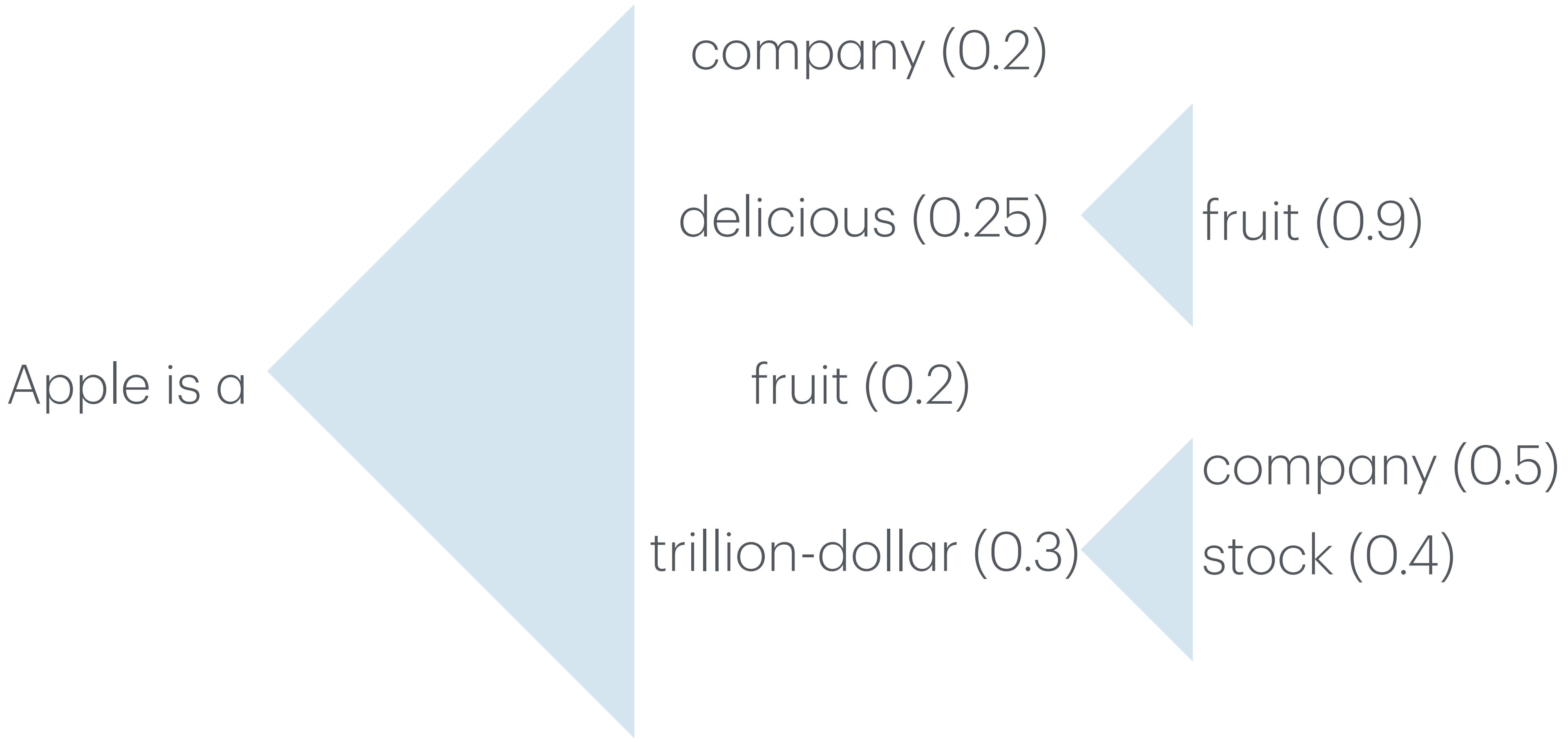


Improving Language Understanding by Generative Pre-Training. Radford et al. 2018.

Sampling - Nucleus sampling

Top-p

- Random sampling
  - Ignore p least likely percentile

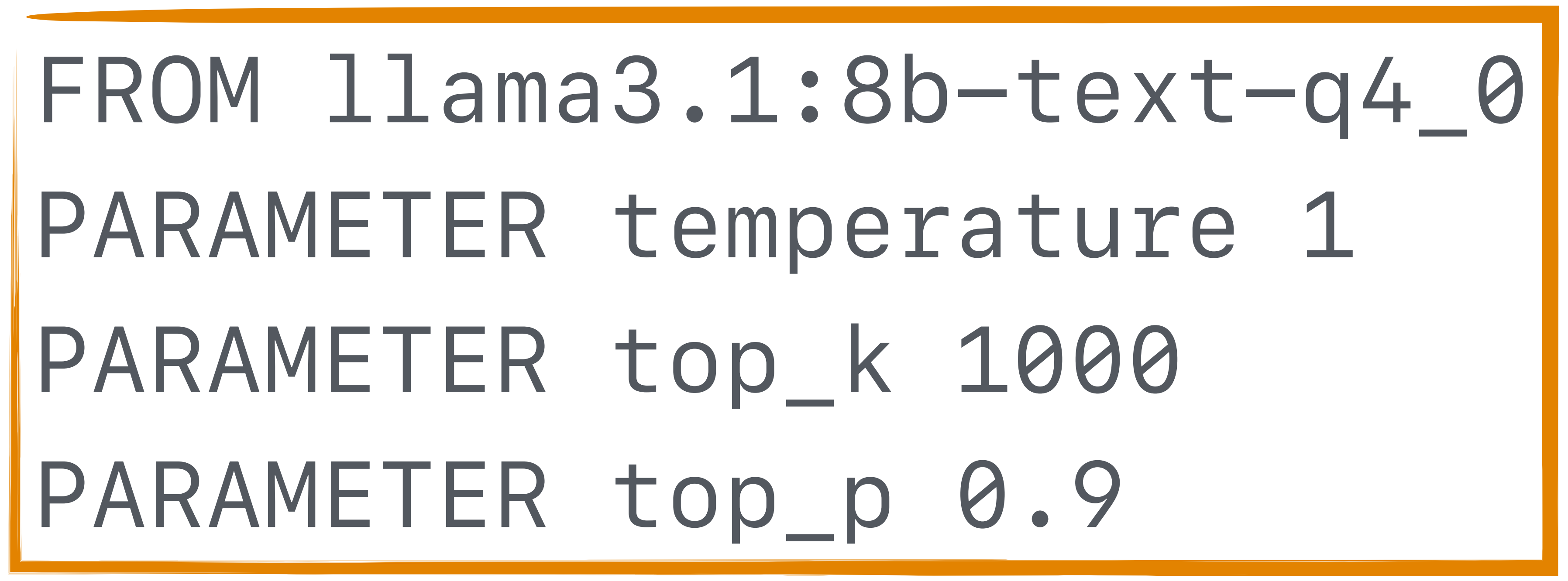


The Curious Case of Neural Text Degeneration. Holtzman et al. 2019.

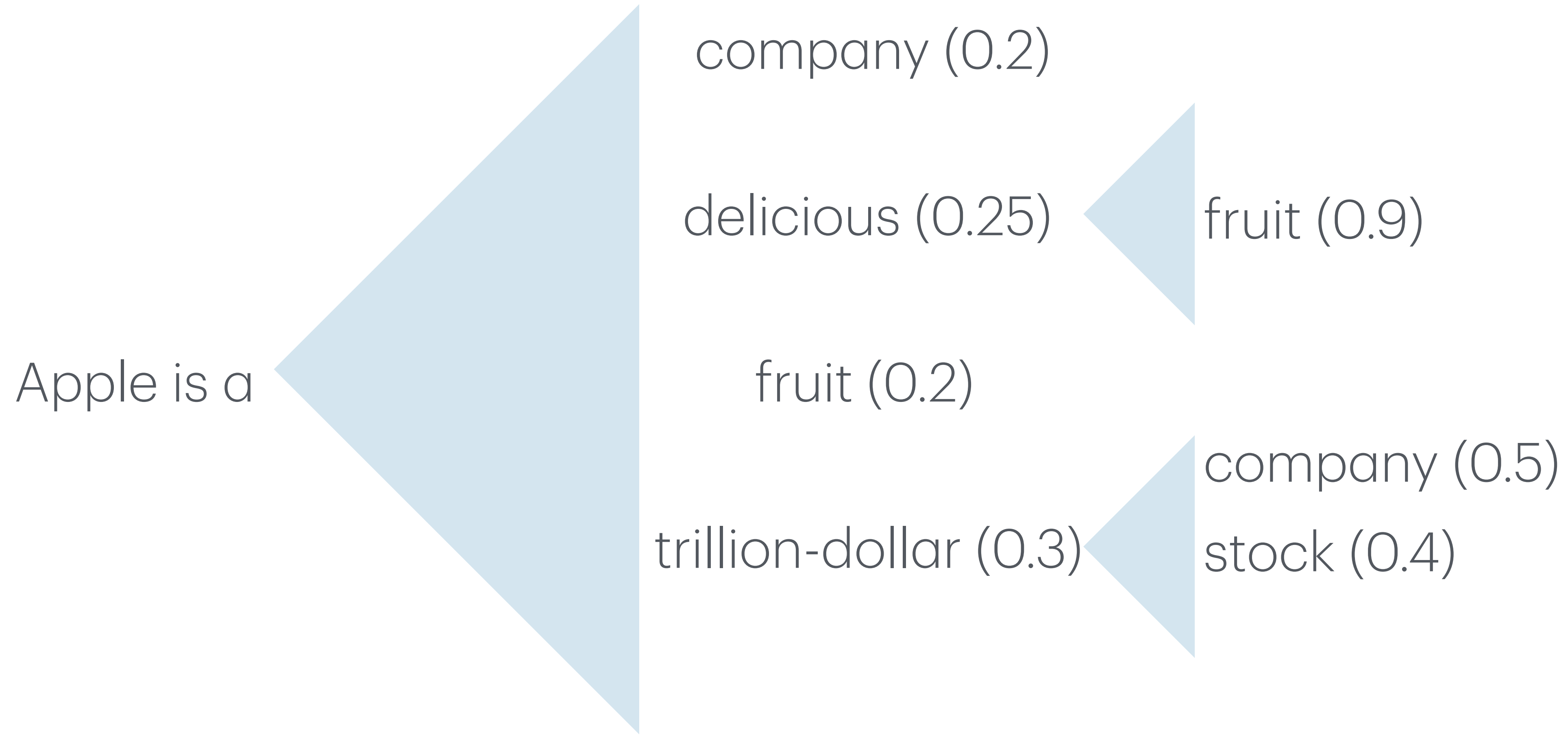
Sampling - Nucleus sampling

Top-p

- A demo
  ollama run llama3.1:top-p



```
FROM llama3.1:8b-text-q4_0
PARAMETER temperature 1
PARAMETER top_k 1000
PARAMETER top_p 0.9
```



The Curious Case of Neural Text Degeneration. Holtzman et al. 2019.

Sampling - Nucleus sampling

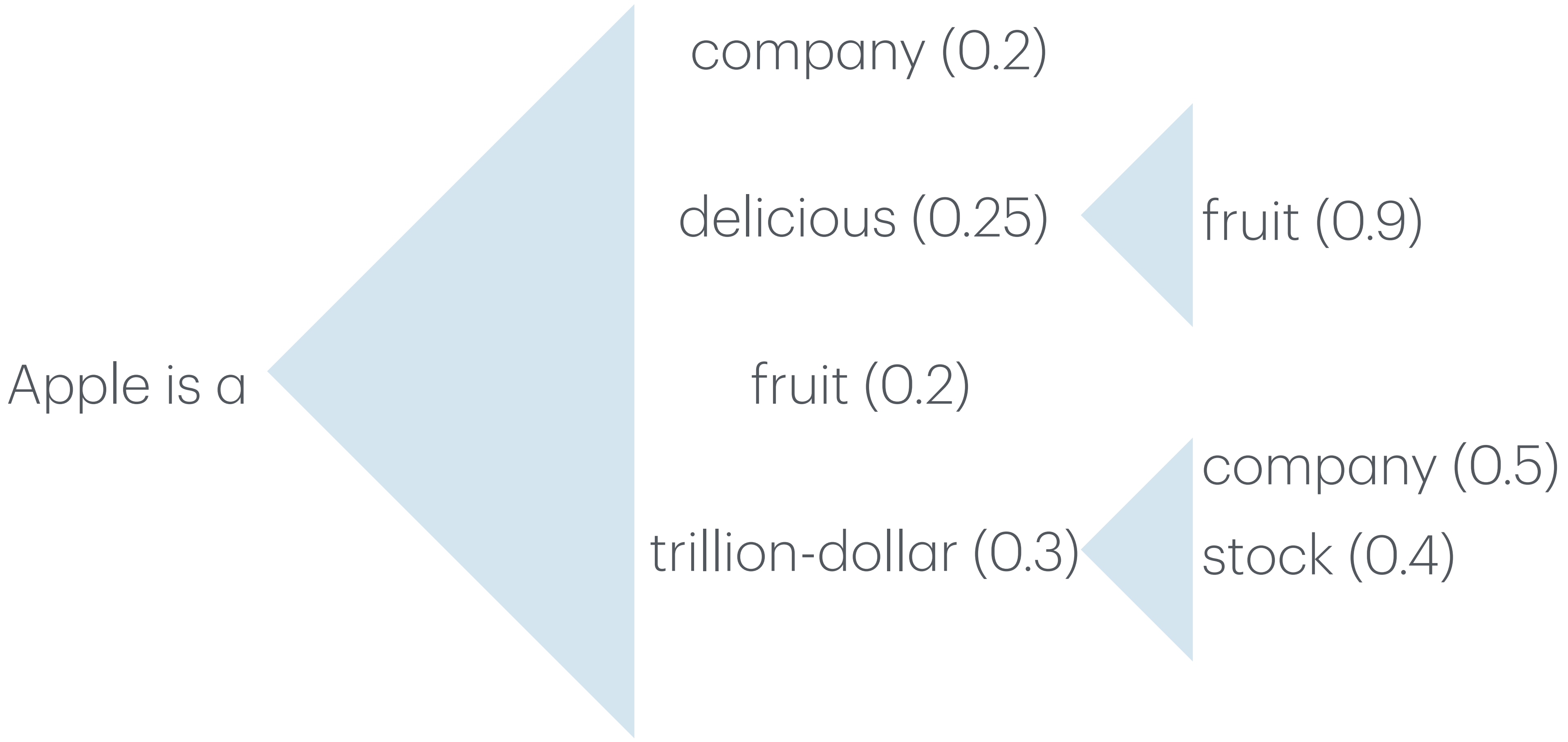
Top-p

- Random sampling
  - Ignore p least likely percentile

- 😄 Samples sound human-like

- 😄 Sampling fewer low-prob transitions

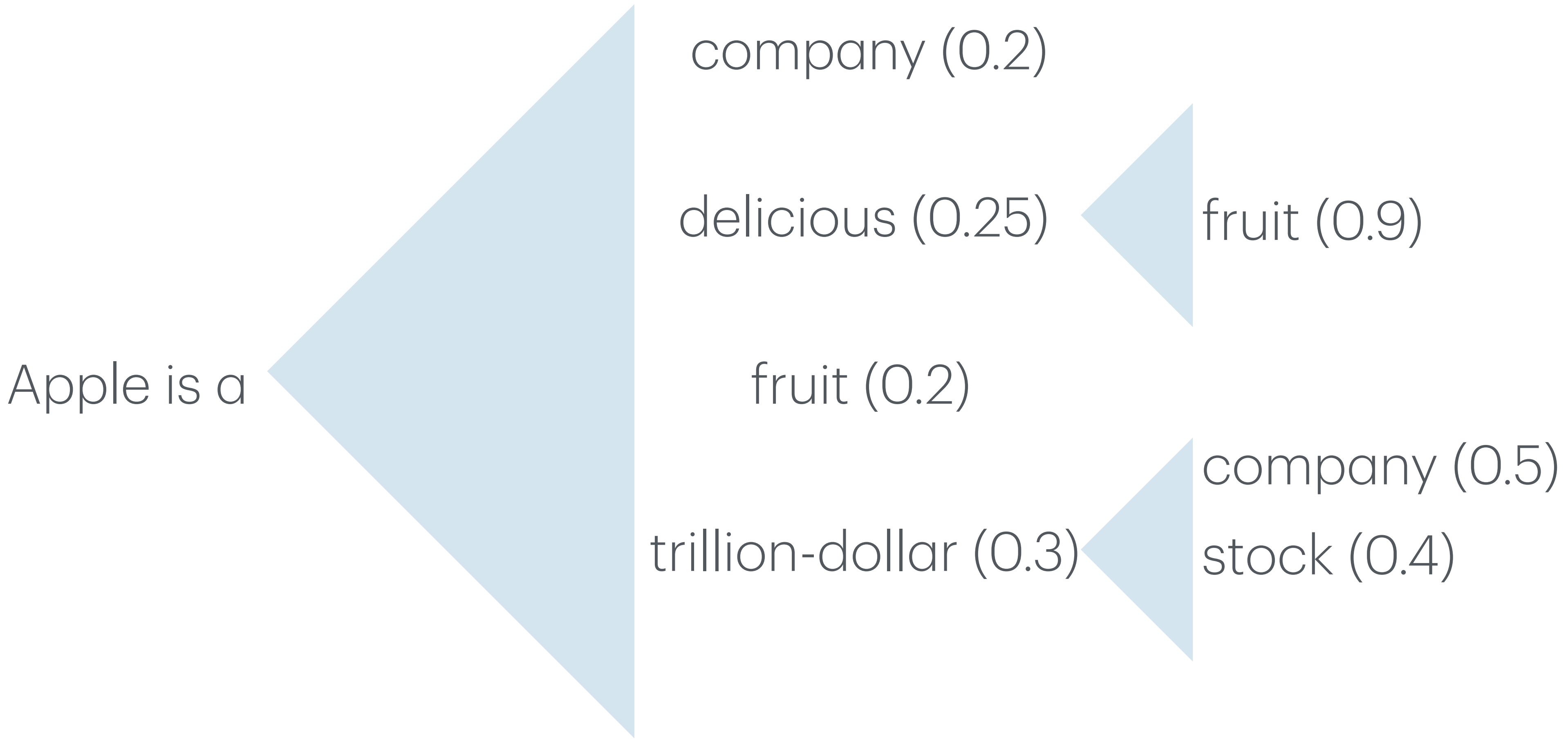
- Used almost everywhere



The Curious Case of Neural Text Degeneration. Holtzman et al. 2019.

Sampling - Min-P

- Random sampling
  - Ignore $p < \alpha p_{\max}$

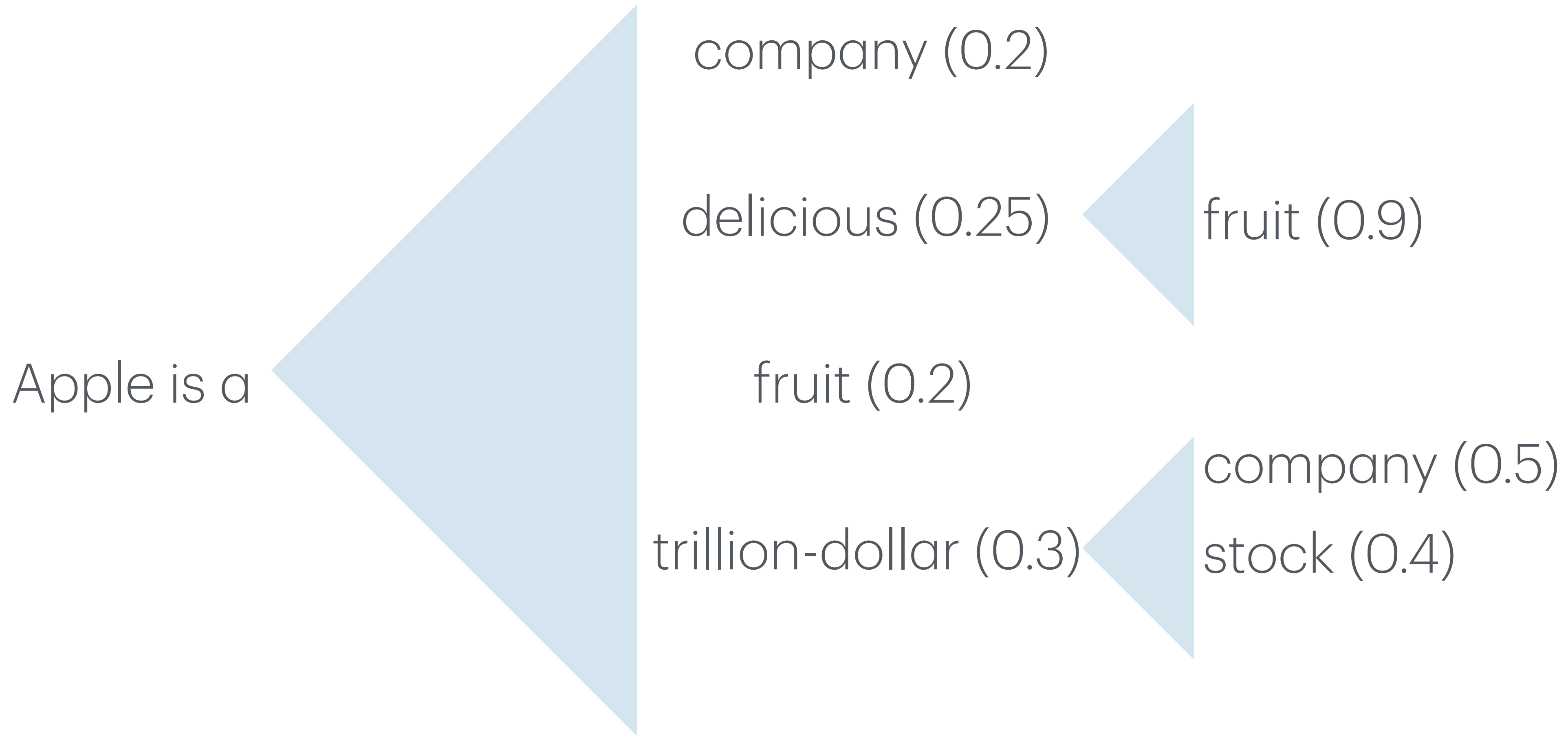


Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs. Nguyen et al. 2024.

Sampling - Min-P

- A demo
  ollama run llama3.1:min-p

```
FROM llama3.1:8b-text-q4_0
PARAMETER temperature 1
PARAMETER top_k 1000
PARAMETER top_p 1.0
PARAMETER min_p 0.1
```



Apple is a

company (0.2)

delicious (0.25)

fruit (0.9)

fruit (0.2)

company (0.5)

trillion-dollar (0.3)

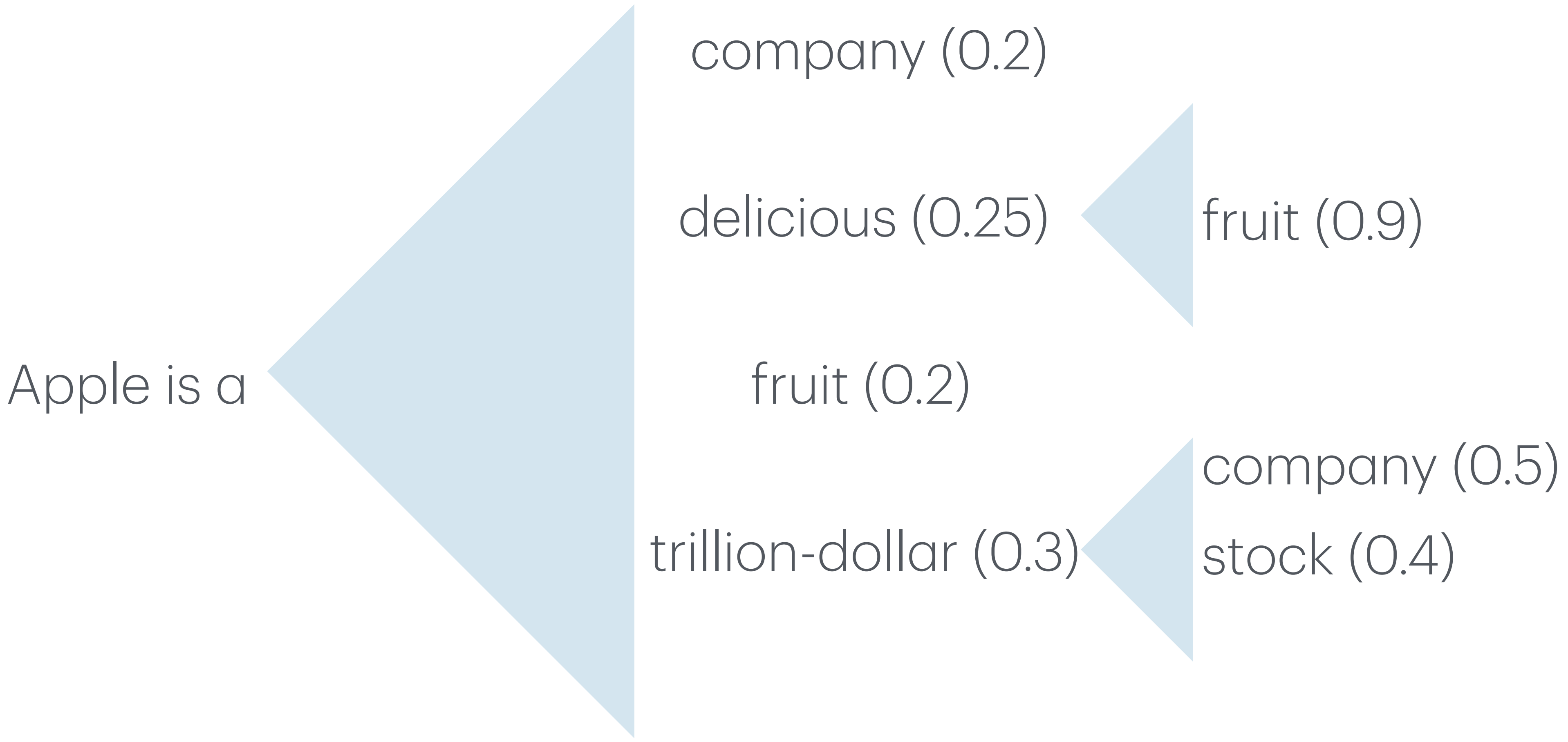
stock (0.4)

Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs. Nguyen et al. 2024.

Sampling - Min-P

- Random sampling
  - Ignore $p < \alpha p_{\max}$

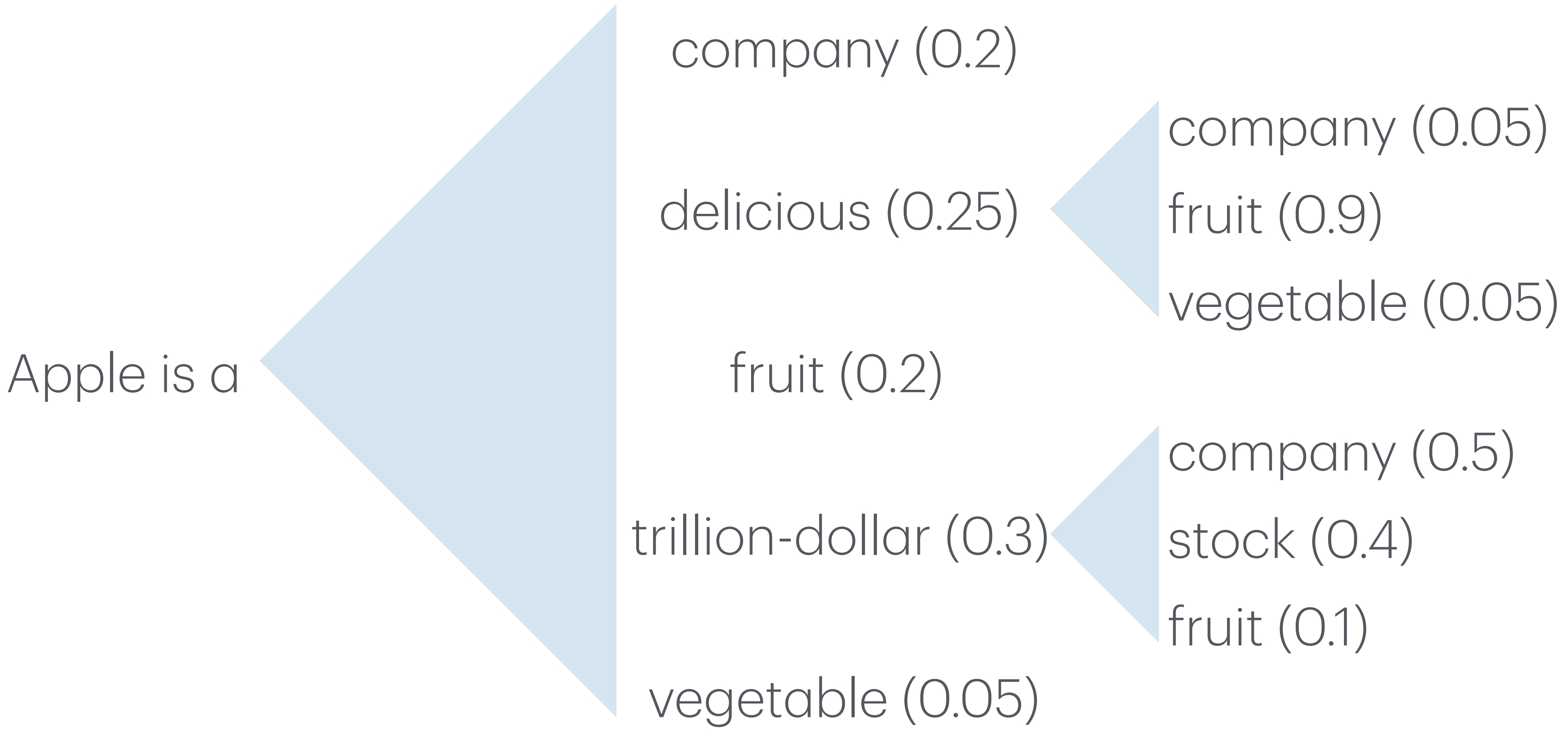
- 😄 Samples sound human-like
- 😄 Sampling fewer low-prob transitions
- Top-p better understood



Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs. Nguyen et al. 2024.

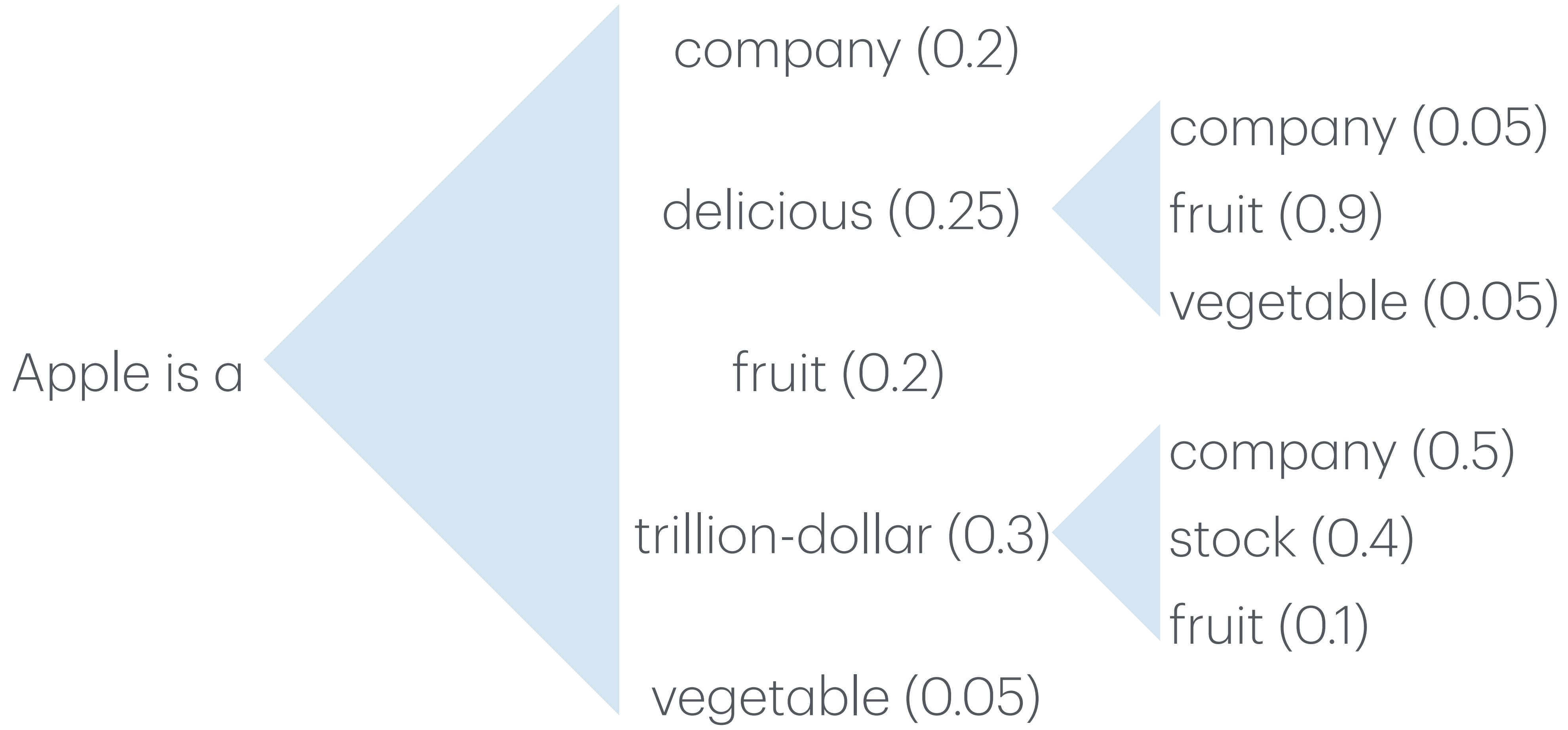
Sampling - Temperature

- More or less creative (random) writing by raising model prob to power $\frac{1}{T}$
  - Temperature T
  - Equivalent to multiplying logits with $\frac{1}{T}$
- $T = 0$: Greedy sampling
- $T \to \infty$: Uniform generation

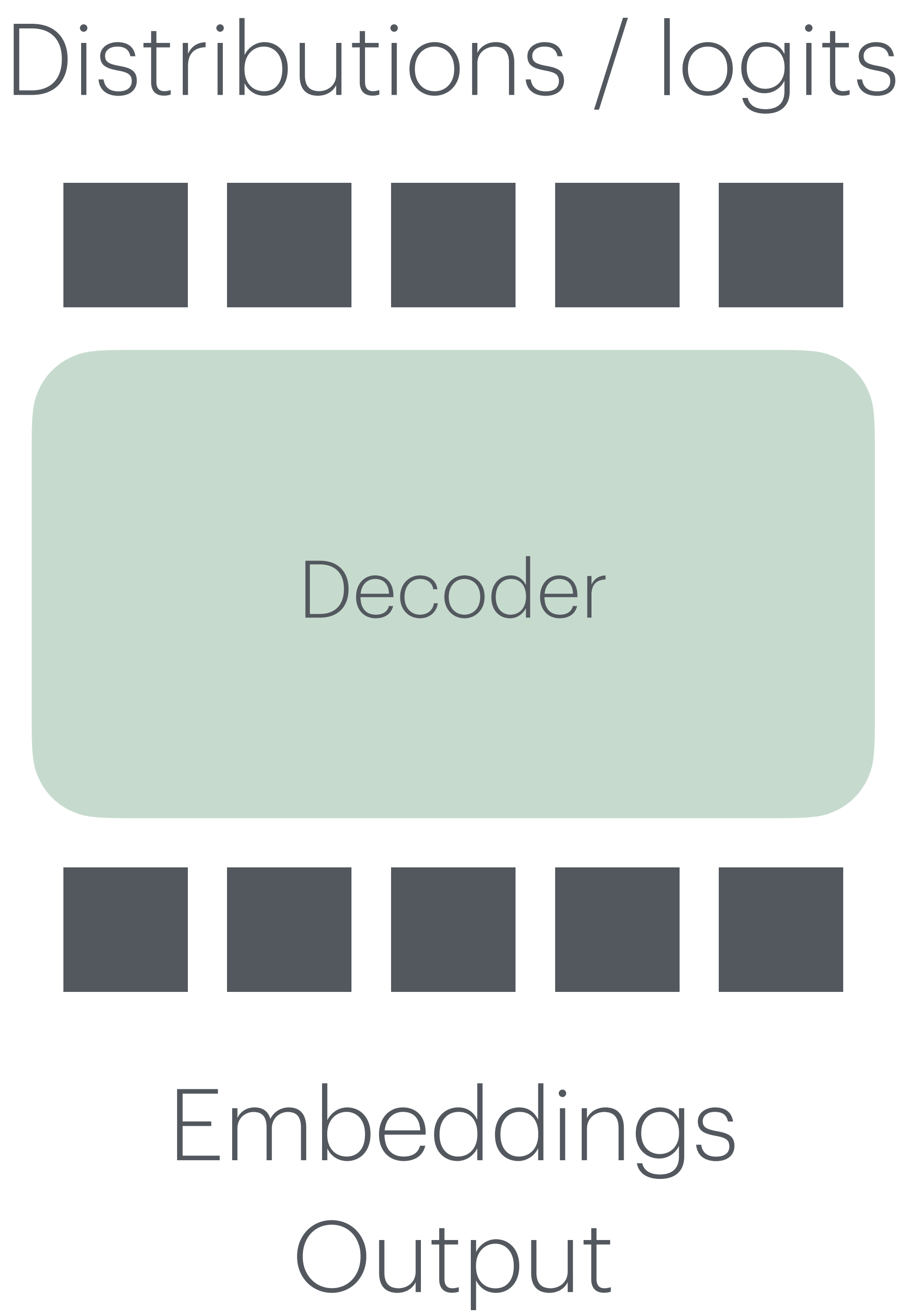


Sampling - When do we stop?

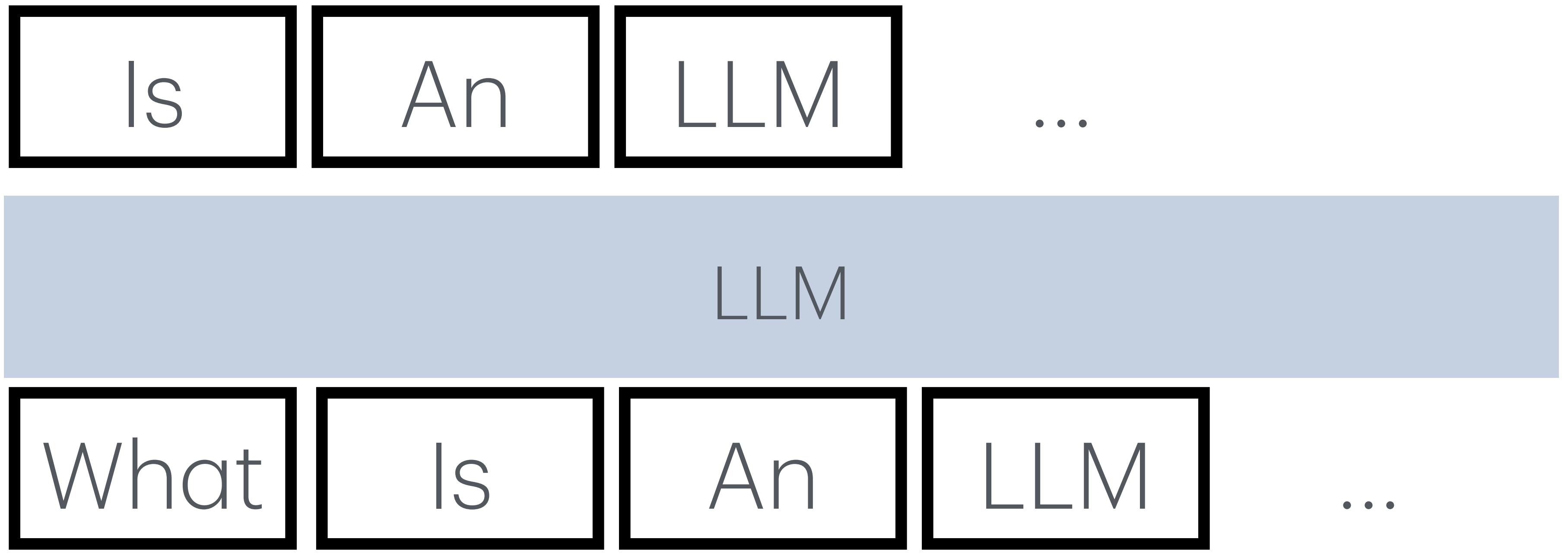
- LLMs have special tokens [bos], [eos]



Language Models



- Decoder-only LLMs
    - Modeling auto-regressive distribution over tokens
    - $P(\mathbf{t}) = P(t_1)P(t_2\,|\,t_1)P(t_3\,|\,t_1, t_2)P(t_4\,|\,t_1\ldots t_3)\ldots$
- Generation / Sampling: $\mathbf{t} \sim P$



References

- [1] Improving Language Understanding by Generative Pre-Training. Radford et al. 2018.
- [2] The Curious Case of Neural Text Degeneration. Holtzman et al. 2019.
- [3] Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs. Nguyen et al. 2024.