The Architecture of Prediction

Understanding N-Gram Language Models: The Mathematical Ancestors of Modern AI

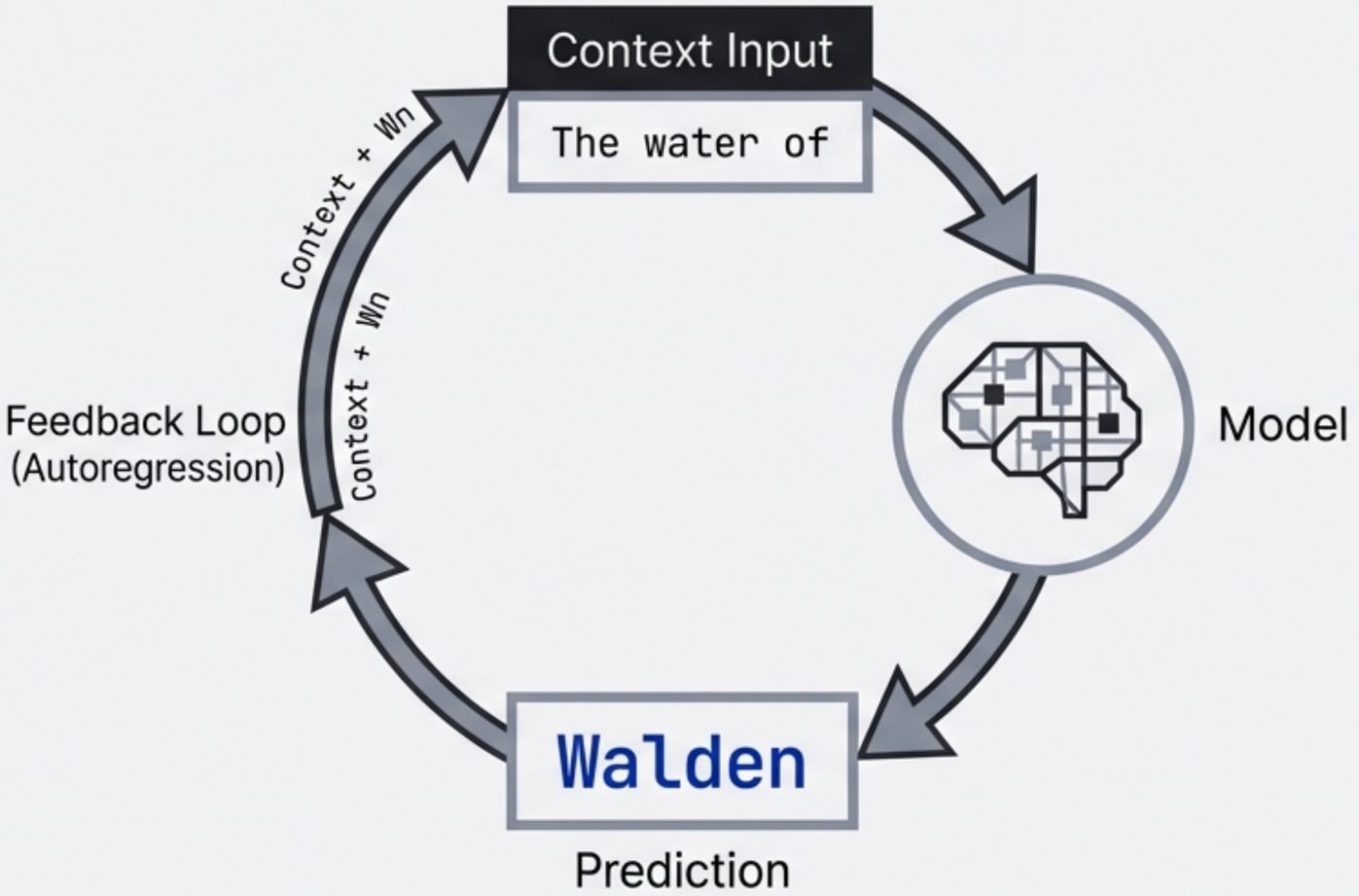
The water of Walden Pond

Based on Lecture 03: N-gram Language Models (Part 1)

At the Heart of Every LLM is a Simple Question: ‘What Comes Next?’

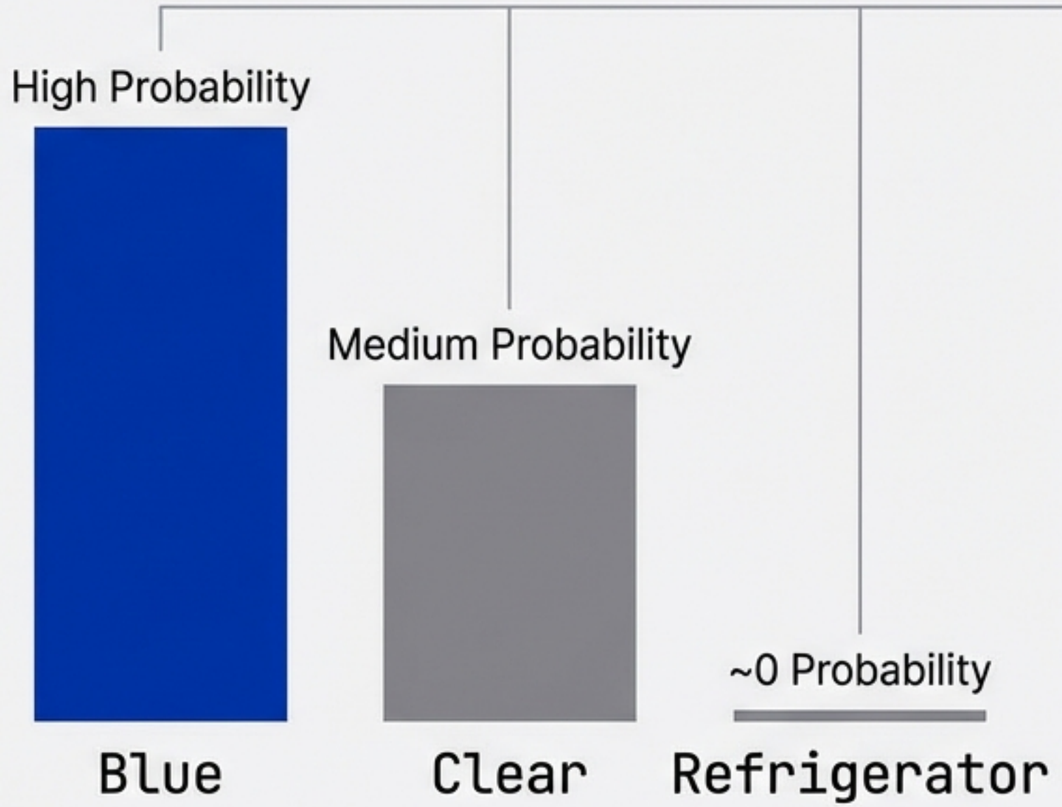
Modern Large Language Models (LLMs) are fundamentally autoregressive. Whether it is a simple N-gram model or a massive Transformer, the training objective remains consistent: predict the next word based on the context of previous words.

- **Pre-training:** Hiding a word to learn context.
- **Generation:** Predicting Wn, appending it, and repeating.



Language is Probabilistic, Not Random

The water of Walden Pond is beautifully [_____]



A Language Model (LM) assigns a probability to every word in the vocabulary (V). The most contextually appropriate word receives the highest mass.

Probability Resolves Ambiguity

Context-Sensitive Spelling

~~Their is no plan.~~ ✗ Low P(Sequence)

There is no plan. ✓ High P(Sequence)

Speech Recognition (ASR)

I'll be back soonish. ✓

~~I'll be back moonish.~~ ✗

Language models act as statistical referees for noisy input.

Formalizing the Intuition

1. The Sequence

Let W be a sequence of words: W1, W2, … Wn

2. The Dual Objectives

1. Next-Word Prediction

$$P(W4 \mid W1, W2, W3)$$

Mathematically Connected →

2. Sequence Probability

$$P(W) = P(W1, W2, \ldots, Wn)$$

The Chain Rule of Probability

The joint probability is the product of conditional probabilities.

$$P(\text{The water of Walden}) = P(\text{The}) \times P(\text{water} \mid \text{The}) \times P(\text{of} \mid \text{The water}) \times P(\text{Walden} \mid \text{The water of})$$

This is the “Gold Standard” calculation. **It requires no loss of information.**

The Data Bottleneck: Sparsity

Observed in Corpus

`The water of Walden Pond`

20 Count

Target Sequence

`The water of Walden Pond is so beautifully blue`

0 Count

Zero Probability Problem.
If the model hasn't seen this exact sequence, the **Chain Rule** fails to estimate it.

The Solution: The Markov Assumption

Approximating the future by looking only at the immediate past

$$P(W_n \mid W_1, W_2, \ldots W_{n-1})$$

$$\approx P(W_n \mid W_{n-1})$$

We trade perfect accuracy for calculability. We assume the next word depends only on the previous few words.

Defining the N-Gram Architecture

Unigram (N=1)

$$P(W_n)$$

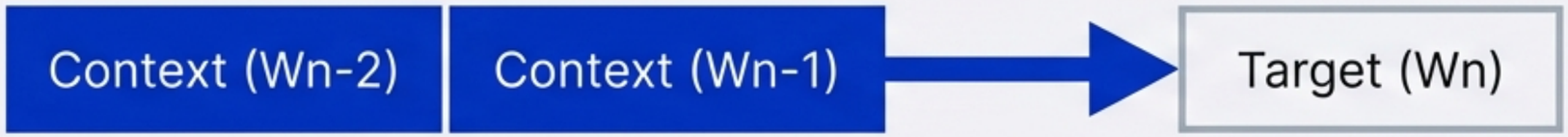
Word

Bigram (N=2)

$$P(W_n \mid W_{n-1})$$

Trigram (N=3)

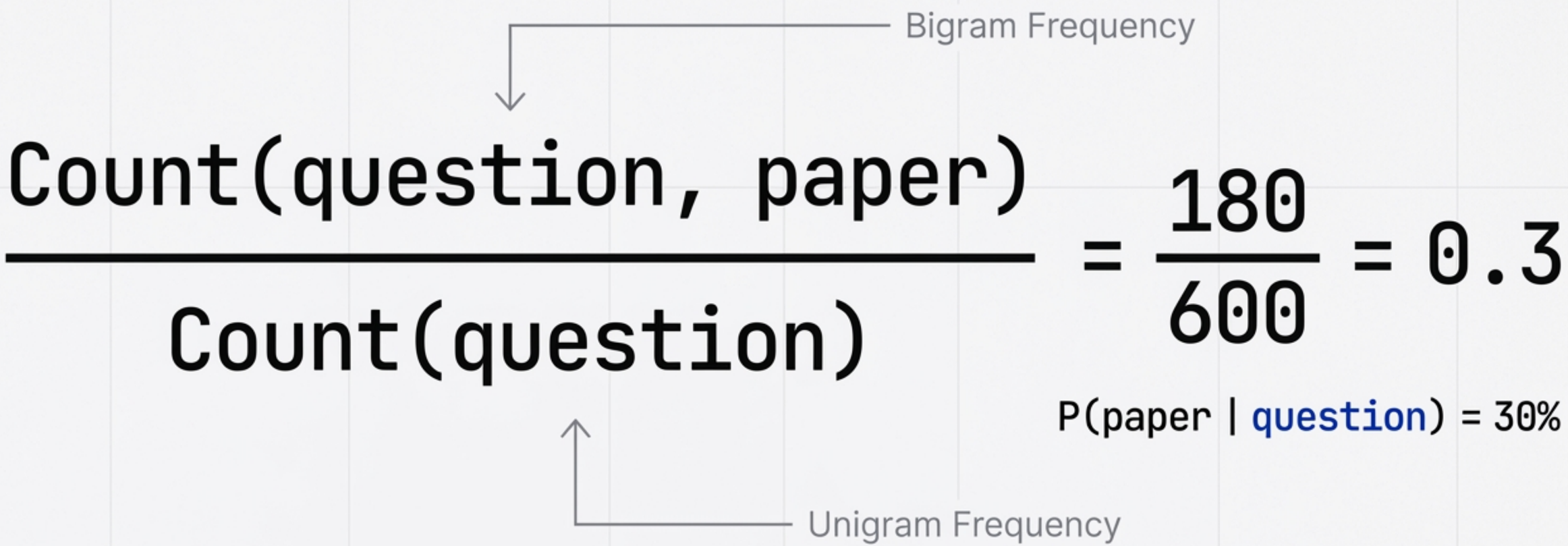
$$P(W_n \mid W_{n-2}, W_{n-1})$$



General Rule: An N-gram looks at N-1 words of past context.

Maximum Likelihood Estimation (MLE)

Converting corpus counts into probabilities.



$$P(\text{paper} \mid \text{question}) = 30\%$$

The Evolution of Meaning: Generating Shakespeare

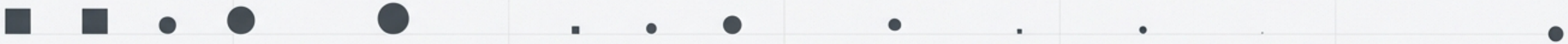
Unigram (Random)	Bigram (Local Sense)	Quadrigram (Near-Sentences)
To him swallowed confess here both which.	I confess all sorts... He is... Would seem... Last December through the way	It cannot be but so indeed the short and the long.
Chaos	Local Coherence	Structural Coherence

Engineering Reality: The Log Space

The Problem

Multiplying Probabilities leads to Underflow.

$$0.003 \times 0.0004 \times 0.00002 \approx 0$$

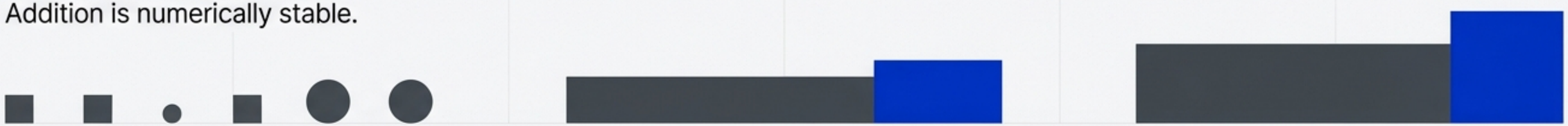


The Solution

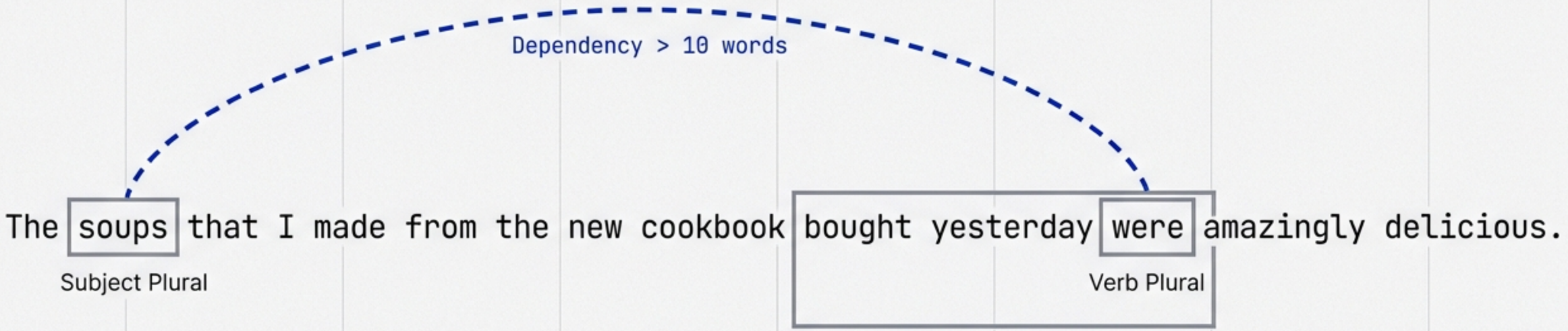
Logarithms convert multiplication to addition.

$$\log(P1) + \log(P2) + \log(P3)$$

Addition is numerically stable.



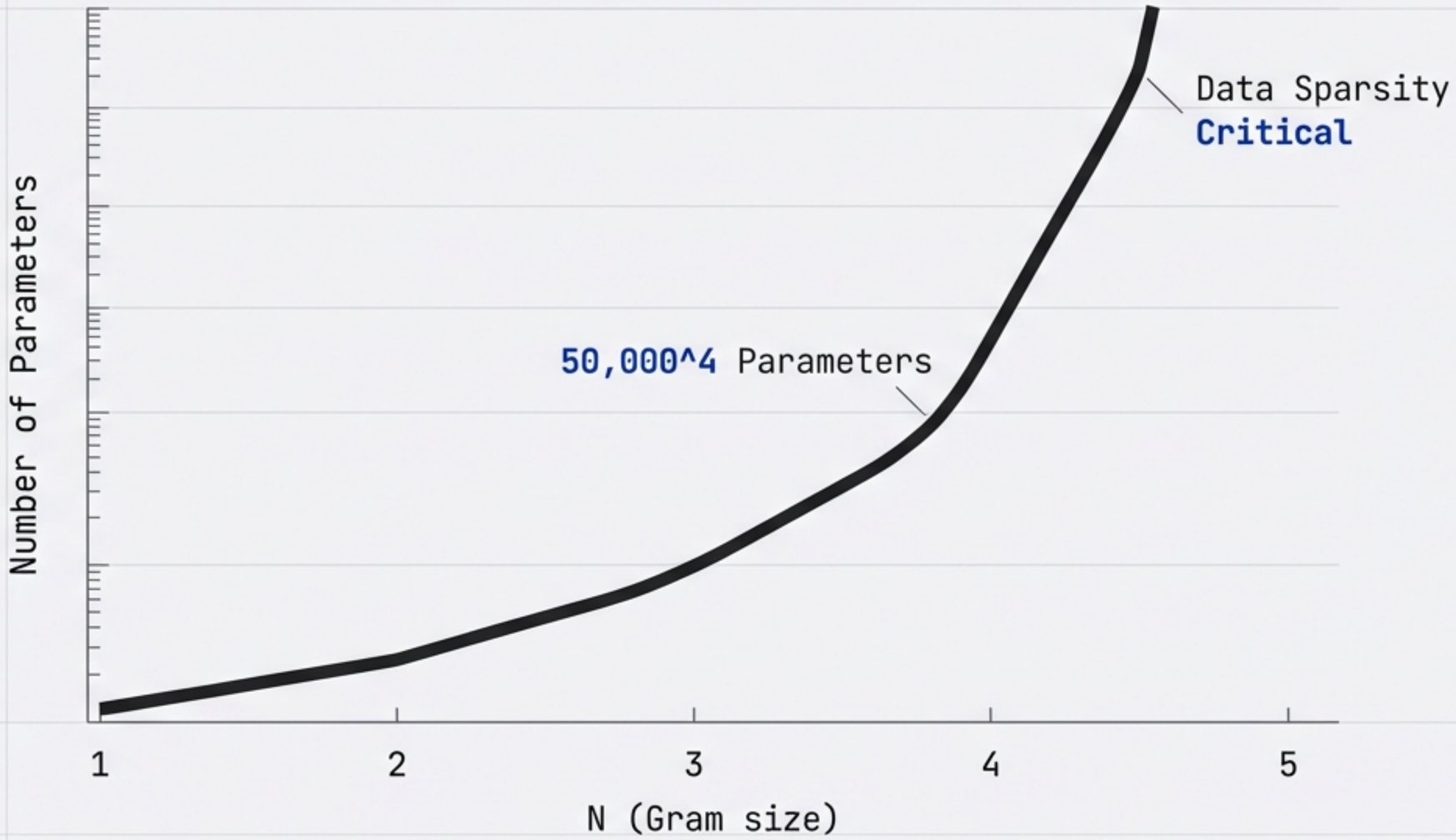
Limitation: Long-Distance Dependencies



The N-Gram model cannot see **'soups'**.
It only sees **'cookbook'** (Singular)
and incorrectly predicts **'was'**.

The Trade-off: The Curse of Dimensionality

Why we can’t just increase N indefinitely.

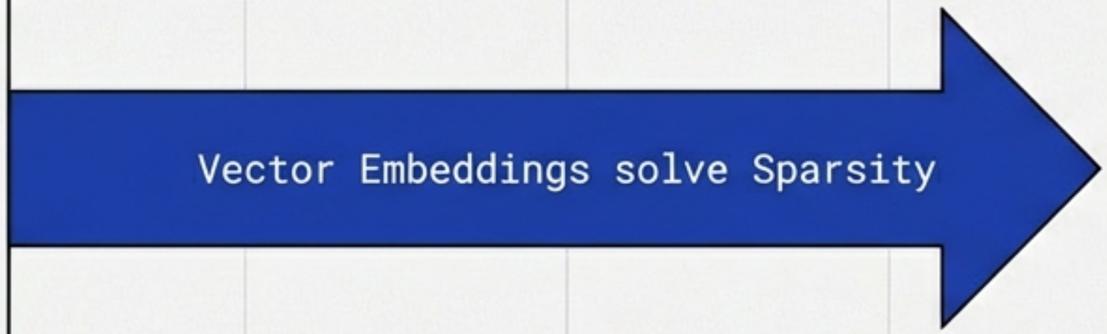


As N grows, the parameter space **V^N** explodes. We run out of data before we can learn the relationships.

The Legacy of N-Grams

Statistical N-Grams

- Evaluation Metrics (Perplexity)
- Training/Test Splits
- Generalization Trade-offs



Modern LLMs

- Massive Context Windows
- Semantic Similarity
- Transformer Architecture

N-grams taught us how to frame the problem.
Deep Learning gave us the capacity to solve it.