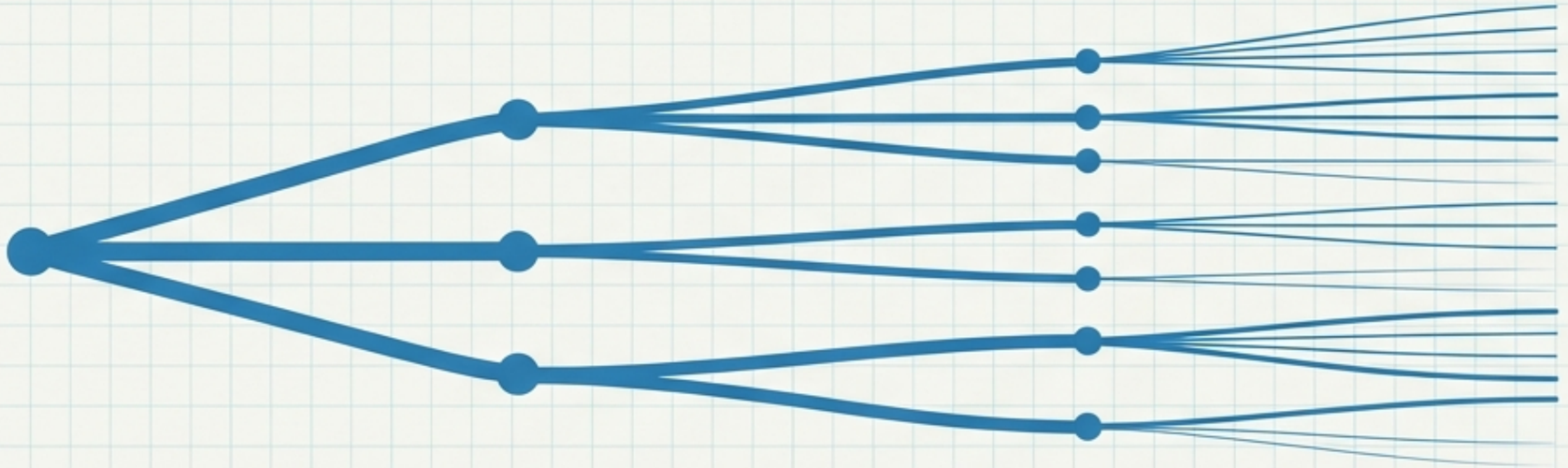
N-gram Language Models: Optimization & Evaluation

Handling uncertainty, smoothing probability distributions, and measuring model performance.

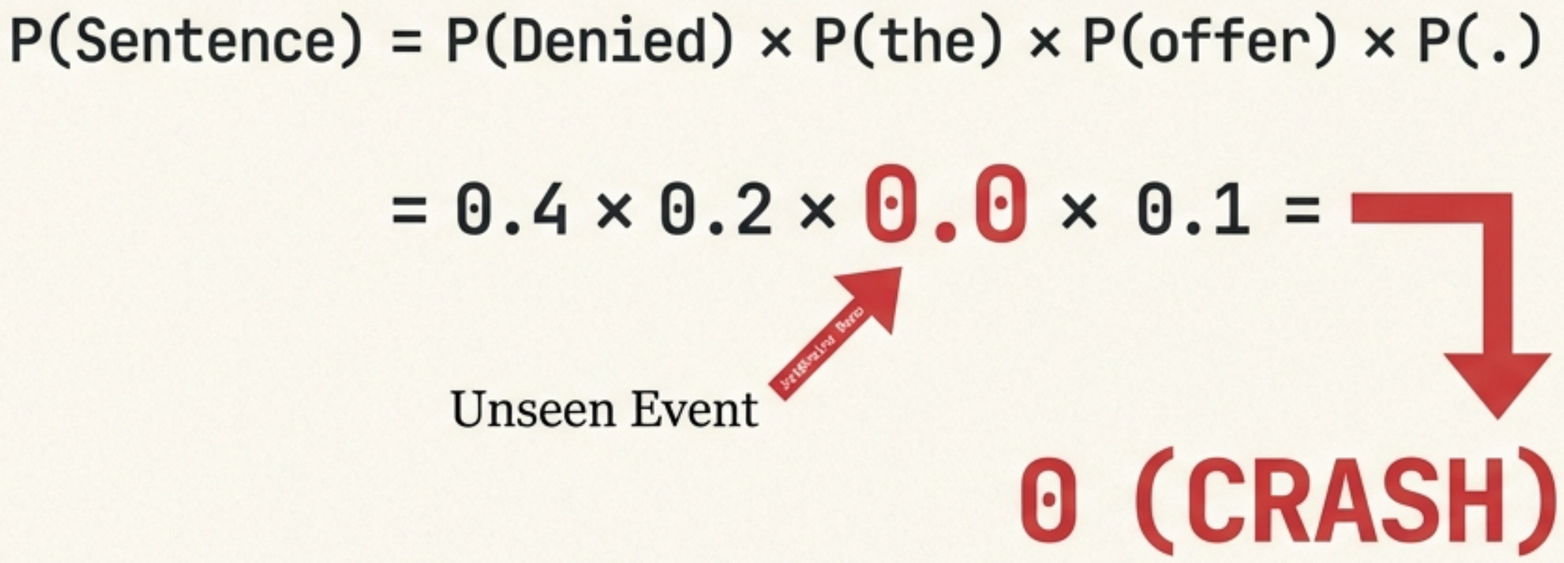


The Tyranny of the Zero

Architectural Paper

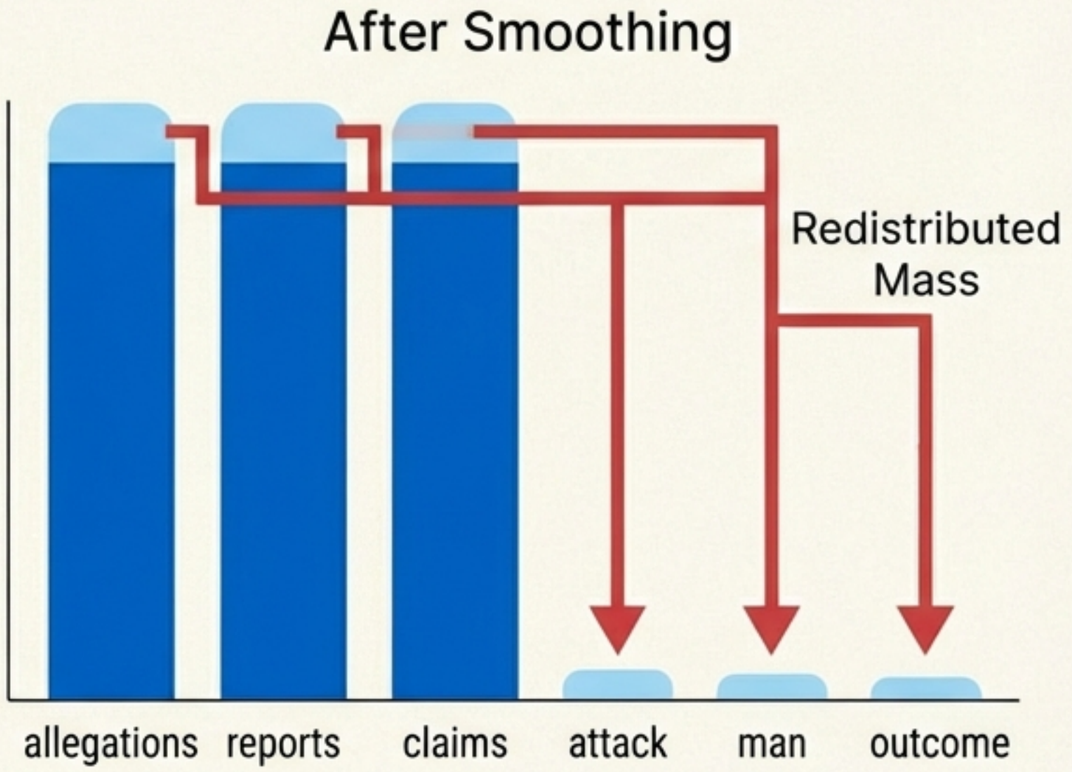
The Core Problem: N-grams rely on a fixed vocabulary. If a phrase like “Denied the offer” appears in testing but was never seen in training, the model assigns it 0% probability.

Because sentence probability is a product chain, a single zero collapses the entire calculation.



Smoothing: Stealing Probability Mass

To fix the zero-probability error, we must account for “unknown unknowns.” We “steal” a small amount of probability mass from frequently seen words and redistribute it to the unseen words.



Add-One (Laplace) Smoothing

$$P(w_i \mid w_{i-1}) = \frac{\text{Count}(w_{i-1}, w_i) + 1}{\text{Count}(w_{i-1}) + V}$$

Pretend we saw it one more time

Normalize by Vocabulary Size

Example Scenario:

Training Data:
Count('Question Paper') = 180,
Count('Question') = 600

Vocabulary Size (V) = 1210

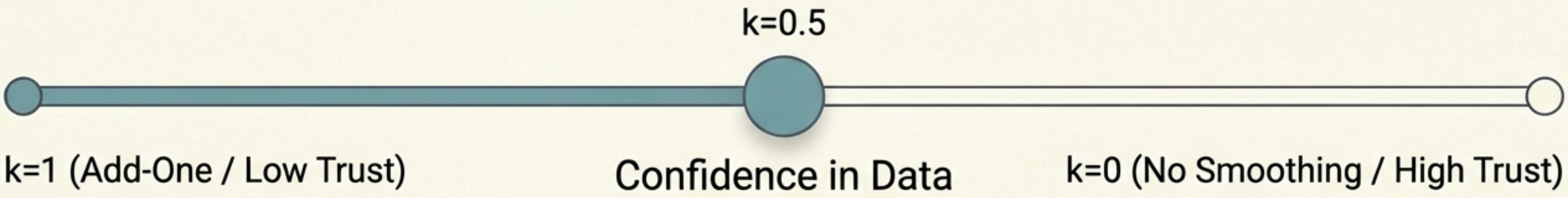
Original	Smoothed
$\frac{180}{600} = 0.30$	$\frac{180 + 1}{600 + 1210} = \frac{181}{1810} \approx 0.10$

Note how the probability of the seen event drops to make room for unseen events.

Refining the Solution: Add-k Smoothing

Add-One is a blunt instrument that moves too much mass. Add-k allows us to tune the smoothing by adding a fractional count (k) instead of 1.

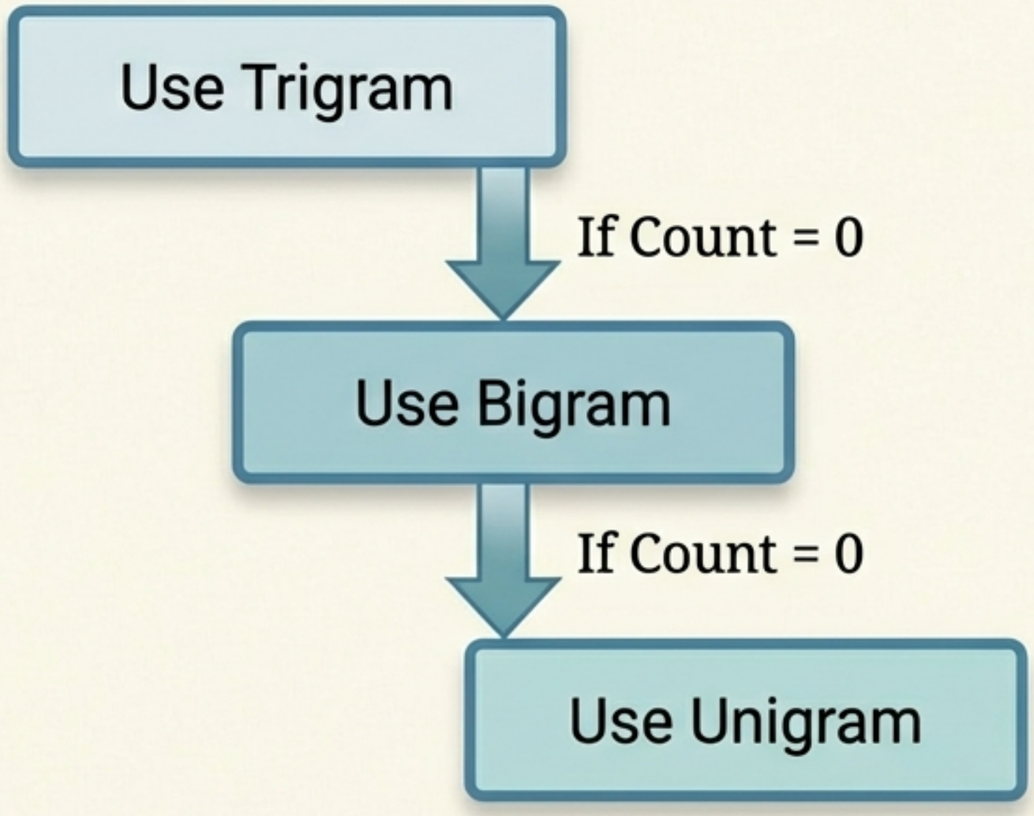
$$P_{\text{add-k}}(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i) + k}{C(w_{i-1}) + k*V}$$



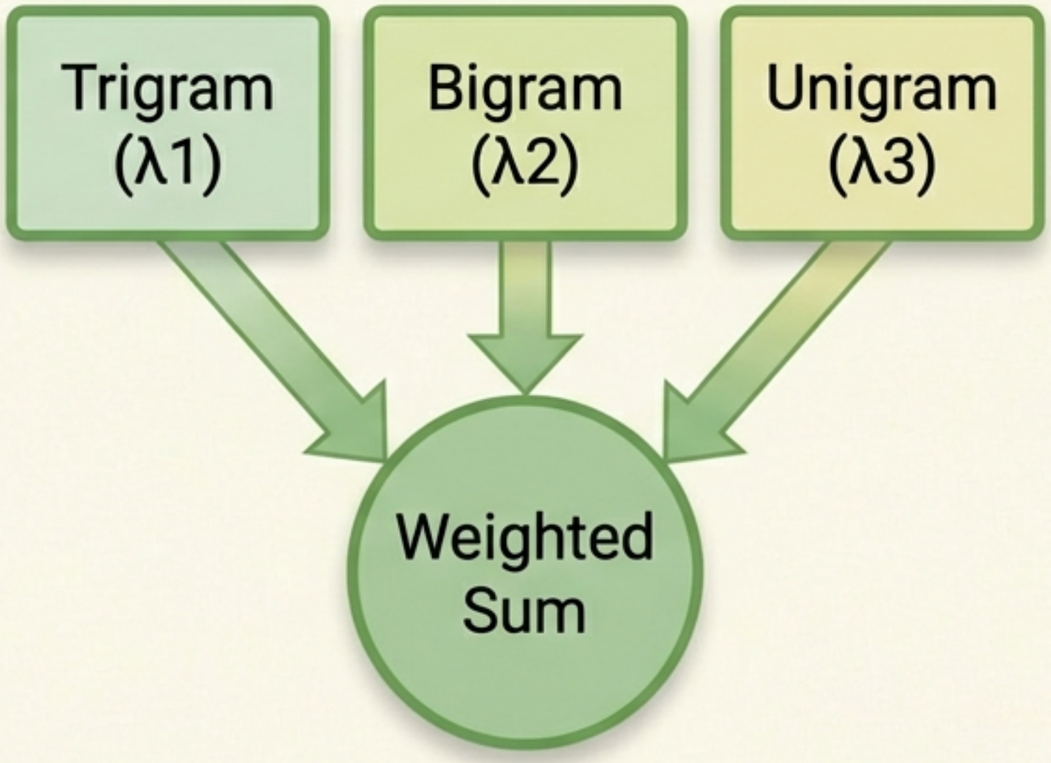
Interpret kV as a parameter M. We are interpolating the observed counts with a uniform distribution (1/V).

Advanced Strategy: Backoff vs. Interpolation

Backoff (The Safety Net)



Interpolation (The Mixer)



Sum of Lambdas (λ) must equal 1

Evaluating Success

How do we choose the best model?

Extrinsic Evaluation

Real-world Task (e.g., Translation, Speech Rec).

✓ Pros

Accurate to specific utility.

✗ Cons

Expensive, Slow, Hard to generalize.

Intrinsic Evaluation

Probability Check (Perplexity).

✓ Pros

Fast, Standard metric, Task-independent.

✗ Cons

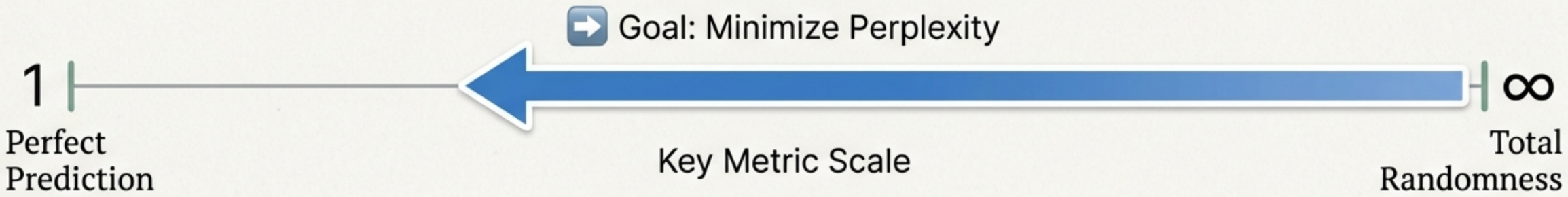
Approximation of quality.

We use Intrinsic Evaluation (Perplexity) for rapid optimization.

Perplexity (PP): A Measure of Surprise

Perplexity measures how “surprised” a model is by the test set. A lower score means the model is less surprised and assigns higher probability to the text.

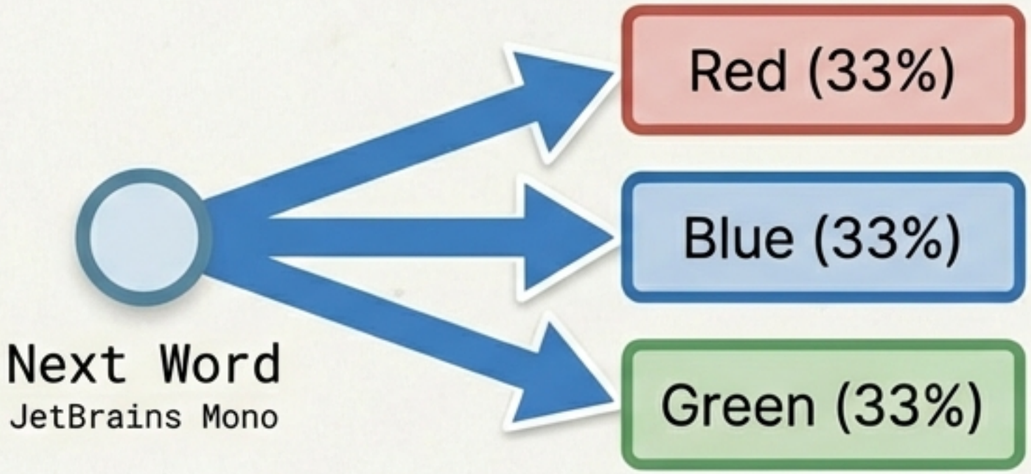
$$PP(W) = P(w_1 \ldots w_n)^{-1/N} = \sqrt[N\text{-th Root}]{\left(\frac{1}{\text{Probability of Sequence}}\right)}$$



Intuition: Perplexity as Branching Factor

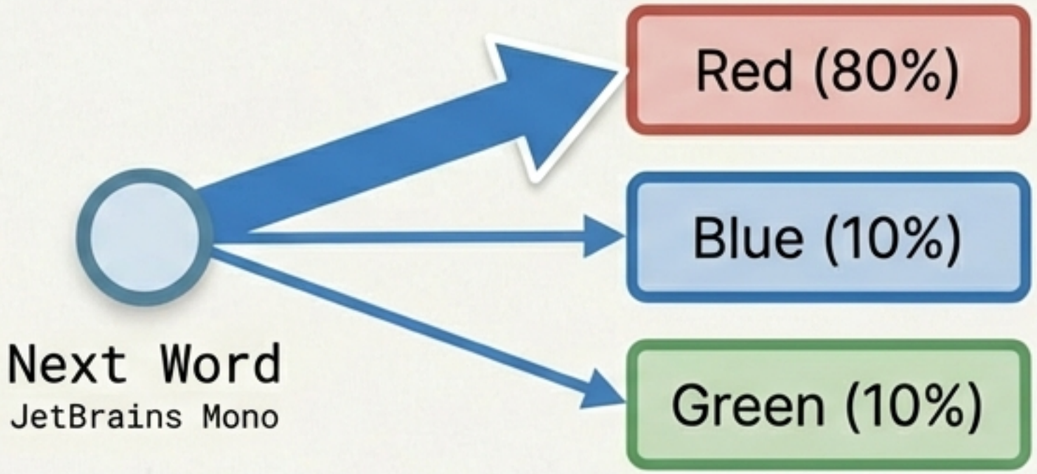
Think of Perplexity as the number of equally probable words the model has to choose from.

Confused Model (PP = 3)



Rolling a 3-sided die.

Learned Model (PP = 1.89)

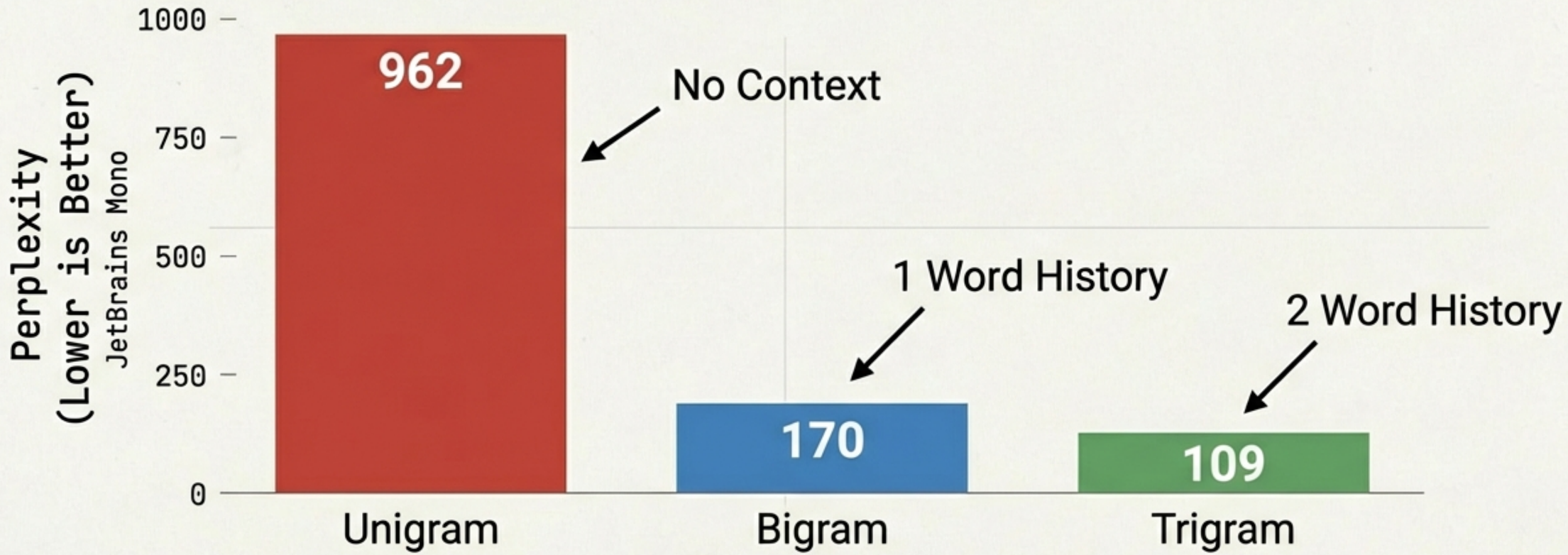


Mostly sure it's Red.

Test Sequence: "Red Red Red Red Blue". The Learned Model is less surprised.

Context Reduces Confusion

Wall Street Journal Benchmarks (38M words training)



Adding just one word of history dramatically drops uncertainty from 962 choices to 170.

Data Hygiene: The Protocol

Never train on the test set. Doing so is cheating (overfitting).

Training Set

Estimate Parameters (Counts)

JetBrains Mono

Dev Set (Held Out)

Tune Hyperparameters (Find best k)

JetBrains Mono

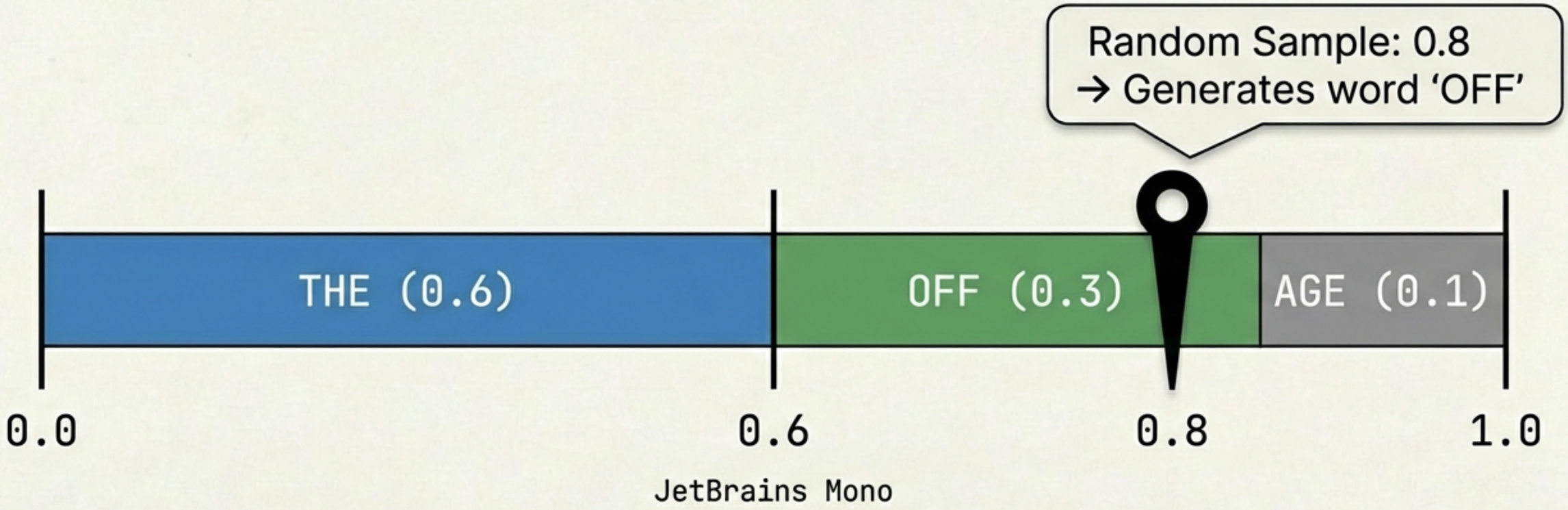
Test Set

Final Evaluation ONLY

JetBrains Mono

Generation: Mapping Probability to Text

How N-grams write. We map the probability distribution to a number line and sample a random number.

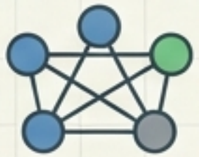


JetBrains Mono

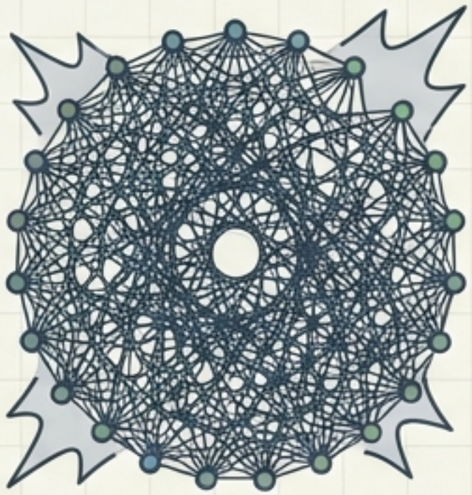
Shannon’s Method: Words are generated proportionally to their likelihood.

Scaling Up: The Storage Challenge

As N increases, parameter count grows exponentially. Storing 5-grams for the entire web is computationally massive.



N=3
(Trigrams)



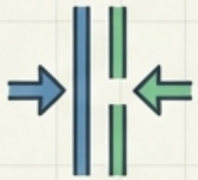
N=5
(5-grams)

Combinatorial Explosion

Solutions

Large Corpora

Google Web 1T



Quantization

Storing probabilities as 4-bit integers instead of 64-bit floats.

JetBrains Mono

Pruning

Removing singletons (count < 2).

JetBrains Mono

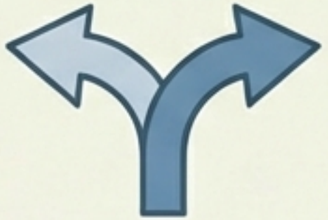
The Cutting Edge: Infinigram (2024)

Instead of a fixed N, Infinigram looks back as far as possible until it finds a match in the training data using Suffix Arrays.

Unlimited context window allows for exact retrieval of specific entities.

Key Takeaways

Zeros break models. Smoothing (Add-k) is required to handle the unknown.



Perplexity is the standard metric. Lower perplexity = less confusion.

Hygiene is critical. Tune on Dev sets, never on Test sets.

Context matters. From Bigrams to Infinigrams, history improves prediction.

Source: Jurafsky & Martin (Chapter 3) / NPTEL Lecture 04