Kneser–Ney on the Integers

v2: The Ring Structure

Claude and MJC

February 2026

Abstract

We rewrite the KN-as-quotient paper using the integer framework from the UM arithmetic series: events are integers, event spaces are rings $\mathbb{Z}/N\mathbb{Z}$, and all maps between event spaces are modular reduction. The KN context of order k is an integer $c \in \mathbb{Z}/256^{k-1}\mathbb{Z}$, and the order projection that drops the oldest byte is reduction mod 256^{k-2} —literal integer division by 256. The discount D is subtraction in $\mathbb{Z}$. The GCD of the count row is the GCD in $\mathbb{Z}$. The continuation count is the fiber cardinality of the column projection. The interpolation recursion is a tower of ring surjections $\mathbb{Z}/256^K\mathbb{Z} \twoheadrightarrow \cdots \twoheadrightarrow \mathbb{Z}/256\mathbb{Z}$. None of these are analogies; with exact arithmetic they are the definitions.

This restatement makes several things visible that the v1 paper left obscure: the discount–GCD gap is a gap between subtraction and division in $\mathbb{Z}$; the continuation count is the image size of a ring homomorphism restricted to a fiber; and the unified pattern table is a function on the product ring $\mathbb{Z}/256^{k-1}\mathbb{Z} \times \mathbb{Z}/256\mathbb{Z} \cong \mathbb{Z}/256^k\mathbb{Z}$.

This is v2. v1 introduced the quotient interpretation; v2 restates it on the integers using the framework of the UM arithmetic papers (v1–v4).

1 The Ring of Contexts

1.1 Contexts Are Integers

A byte-level order-k context is a sequence of $k-1$ bytes: $c = (b_1, b_2, \ldots, b_{k-1})$ where $b_i \in \{0, \ldots, 255\}$. Encode this as a base-256 integer:

$$c = b_1 \cdot 256^{k-2} + b_2 \cdot 256^{k-3} + \cdots + b_{k-1} \in \{0, \ldots, 256^{k-1} - 1\}. \tag{1}$$

The context space $I_k = \{0, \ldots, 256^{k-1} - 1\}$ is the ring $\mathbb{Z}/256^{k-1}\mathbb{Z}$.

The output byte $o \in \{0, \ldots, 255\}$ is an integer in $\mathbb{Z}/256\mathbb{Z}$.

The joint event "context c followed by output o" is the integer:

$$e = 256 \cdot c + o = b_1 \cdot 256^{k-1} + b_2 \cdot 256^{k-2} + \cdots + b_{k-1} \cdot 256 + o \in \mathbb{Z}/256^k\mathbb{Z}. \tag{2}$$

This is the context-output pair encoded as a single base-256 number of k digits. The event space $E_k = I_k \times O$ is the ring $\mathbb{Z}/256^k\mathbb{Z}$.

Observation 1 (Product Ring)**.** *The Chinese Remainder Theorem does not apply directly since* $\gcd(256^{k-1}, 256) = 256 \neq 1$*. The decomposition* $E_k = I_k \times O$ *is the mixed-radix decomposition of* $\mathbb{Z}/256^k\mathbb{Z}$*, not a CRT decomposition. This reflects the fact that the context and output share the same alphabet—they are not independent event spaces but successive positions in the same stream.*

1.2 The Count Table Lives on the Ring

The n-gram count table is a function on the ring:

$$c_k : \mathbb{Z}/256^k\mathbb{Z} \to \mathbb{N}, \qquad c_k(e) = |\{t : (b_{t-k+1}, \dots, b_t) = e\}|. \tag{3}$$

It counts how many times the k-byte pattern e appears in the data. The function c_k lives on the ring and inherits the ring's structure.

The row indexed by context c is the restriction of c_k to the coset $\{256c + o : o \in \{0, \dots, 255\}\}$. This coset is an ideal translate in the ring: it contains all events with the same context, varying only the output.

The row total is $c_k(c, \cdot) = \sum_{o=0}^{255} c_k(256c + o)$ —the sum over the coset.

2 The Order Projection Is Modular Reduction

2.1 Dropping the Oldest Byte

The order projection $\pi_k : I_k \to I_{k-1}$ drops the oldest byte from the context. In integer terms:

$$\pi_k(c) = c \bmod 256^{k-2}. \tag{4}$$

This is modular reduction: divide by 256^{k-2} and keep the remainder. The quotient $\lfloor c/256^{k-2} \rfloor = b_1$ is the dropped byte.

Proposition 2 (Order Projection = Reduction mod 256^{k-2})**.** *The map $\pi_k : \mathbb{Z}/256^{k-1}\mathbb{Z} \to \mathbb{Z}/256^{k-2}\mathbb{Z}$ defined by $\pi_k(c) = c \bmod 256^{k-2}$ is a ring surjection. Its kernel is $256^{k-2}\mathbb{Z}/256^{k-1}\mathbb{Z} \cong \mathbb{Z}/256\mathbb{Z}$, which is the "oldest byte" event space—the information that the projection discards.*

Proof. Modular reduction is a ring homomorphism (it preserves addition and multiplication). It is surjective since every element of $\mathbb{Z}/256^{k-2}\mathbb{Z}$ has a preimage. The kernel consists of elements c with $c \bmod 256^{k-2} = 0$, i.e., $c = b_1 \cdot 256^{k-2}$ for $b_1 \in \{0, \dots, 255\}$. □

2.2 The Equivalence Class Is a Residue Class

Two contexts c, c' project to the same shorter context iff $c \equiv c' \pmod{256^{k-2}}$—iff they are in the same residue class. The equivalence class:

$$[c]_{k-1} = \{c' : c' \equiv c \pmod{256^{k-2}}\} = \{c + b \cdot 256^{k-2} : b \in \{0, \dots, 255\}\} \tag{5}$$

has exactly 256 elements (one per value of the oldest byte b). This is a coset of the kernel.

The lower-order count is the sum over the residue class:

$$c_{k-1}(\pi_k(c), o) = \sum_{c' \in [c]_{k-1}} c_k(c', o) = \sum_{b=0}^{255} c_k(c \bmod 256^{k-2} + b \cdot 256^{k-2}, o). \tag{6}$$

Marginalization over the oldest byte = summation over the coset.

2.3 The Full Tower

The tower of projections

$$\mathbb{Z}/256^{K-1}\mathbb{Z} \xrightarrow{\text{mod } 256^{K-2}} \mathbb{Z}/256^{K-2}\mathbb{Z} \xrightarrow{\text{mod } 256^{K-3}} \cdots \xrightarrow{\text{mod } 256} \mathbb{Z}/256\mathbb{Z} \xrightarrow{\text{mod } 1} \{0\} \tag{7}$$

is a chain of ring surjections. Each step reduces mod 256^{k-2}, discarding one base-256 digit (one byte of context). The full chain forgets bytes oldest-first, matching the KN backoff order.

On the joint event space, the tower extends by one position:

$$\underbrace{\mathbb{Z}/256^{K}\mathbb{Z}}_{E_K} \twoheadrightarrow \underbrace{\mathbb{Z}/256^{K-1}\mathbb{Z}}_{E_{K-1}} \twoheadrightarrow \cdots \twoheadrightarrow \underbrace{\mathbb{Z}/256\mathbb{Z}}_{E_1} \tag{8}$$

where each E_k is the ring of k-byte events (($k-1$)-byte context + 1-byte output).

3 The Discount Is Subtraction

3.1 Subtraction in $\mathbb{Z}$

The KN discount subtracts D from each nonzero count:

$$\tilde{c}_k(c, o) = \max(c_k(c, o) - D,\ 0). \tag{9}$$

This is subtraction in $\mathbb{Z}$ (the count ring), clamped at zero. The counts live in $\mathbb{N} \subset \mathbb{Z}$; the discount pushes them toward zero, and the clamp prevents them from going negative.

3.2 The GCD Is the GCD

The per-row GCD from the Bayes-from-counting framework is the GCD in $\mathbb{Z}$:

$$g(c) = \gcd\{c_k(c, o) : c_k(c, o) > 0\}. \tag{10}$$

The "common evidence" is $g(c)$; the "reduced counts" are $r(c, o) = c_k(c, o)/g(c)$. Since $g(c)$ divides every nonzero count in the row, division is exact (no remainder). The reduced counts are coprime: $\gcd\{r(c, o) : r(c, o) > 0\} = 1$.

3.3 The Discount–GCD Gap Is Subtraction vs. Division

The v1 paper identified the discount–GCD gap as an open question. With the integer framework, the gap has a precise characterization:

Proposition 3 (The Gap)**.** *The KN discount subtracts a constant D from each count: $\tilde{c} = c - D$. The GCD operation divides each count by g: $r = c/g$. These are different operations in $\mathbb{Z}$:*

- ***Subtraction** $c - D$: shifts the count down by D. The result depends on c: small counts lose a larger fraction of their value than large counts.*
- ***Division** c/g: scales the count by $1/g$. The result is proportional to c: all counts lose the same fraction $1 - 1/g$ of their value.*

Subtraction and division agree when $g = 1$ and $D = 1$ (both reduce each count by 1). They disagree when $g > 1$: division removes a factor (preserving ratios), while subtraction removes an additive constant (distorting ratios in favor of large counts).

Remark 4 (Why subtraction "works")**.** *For natural language at the byte level, most rows have* $g(c) = 1$ *(at least one continuation appears exactly once). For these rows, the GCD operation is "subtract 1 from each count," which is close to the KN discount* $D \approx 0.9$*. The rows where* $g > 1$ *are high-frequency contexts ("th", "he", etc.) where division would be more appropriate—but these rows also have large counts, so the distortion from subtraction is small relative to the totals.*

The KN discount works not *because subtraction is the right operation, but because* $g = 1$ *almost everywhere, and where* $g > 1$ *the counts are large enough that the error is negligible.*

3.4 Per-Row GCD as UM-Native Discount

The UM-native discount uses division instead of subtraction:

$$r(c, o) = c_k(c, o) \ / \ g(c). \tag{11}$$

This is exact integer division (no remainder, by definition of GCD). The reduced counts are the "differential evidence"—what remains after the common evidence $g(c)$ is divided out.

The "mass" removed is $c_k(c, o) - r(c, o) = c_k(c, o)(1 - 1/g(c))$, which varies by row. For $g = 1$: nothing removed ($r = c$). For $g = 2$: half removed. For $g = 10$: 90% removed.

The removed mass goes to the backoff distribution, just as in KN. The difference: KN removes a constant D per continuation; the GCD removes a fraction $1 - 1/g(c)$ per row. The GCD version is the ring-native operation.

4 The Continuation Count Is Fiber Cardinality

4.1 Fibers of the Output Projection

Consider the output projection $\rho : E_k \to O$ that extracts the output byte: $\rho(c, o) = o$, i.e., $\rho(e) = e \bmod 256$. This is reduction mod 256 on the joint ring.

The fiber of o under ρ is:

$$\rho^{-1}(o) = \{e \in E_k : e \equiv o \pmod{256}\} = \{256c + o : c \in I_k\}. \tag{12}$$

This is a coset of the kernel $256\mathbb{Z}/256^k\mathbb{Z}$ in the joint ring.

4.2 Continuation Count = Non-Empty Fiber Sections

The continuation count $c_{\mathrm{KN}}(o)$ counts the number of distinct contexts c such that $c_k(c, o) > 0$:

$$c_{\mathrm{KN}}(o) = |\{c \in I_k : c_k(256c + o) > 0\}|. \tag{13}$$

This is the number of "occupied" positions in the fiber $\rho^{-1}(o)$—the fiber cardinality restricted to the support of c_k.

Proposition 5 (Continuation Count as Fiber Size)**.** *Define the support* $S = \{e \in \mathbb{Z}/256^k\mathbb{Z} : c_k(e) > 0\}$ *(the set of observed k-grams). The continuation count of output o is:*

$$c_{\mathrm{KN}}(o) = |S \cap \rho^{-1}(o)| \tag{14}$$

—the size of the support's intersection with the output fiber.

In the ring, $\rho^{-1}(o)$ is a coset with $|I_k| = 256^{k-1}$ elements. The continuation count measures what fraction of this coset is "populated" in the data. High continuation count = the output byte o appears in many contexts (general). Low = it appears in few contexts (specific).

4.3 Why Type Counts Matter

The raw count $c_k(\cdot, o) = \sum_c c_k(c, o)$ sums over the fiber *with multiplicity*: each context contributes its count. High-frequency contexts dominate. The continuation count $c_{\text{KN}}(o)$ sums over the fiber *without multiplicity*: each context contributes 1 or 0. This is the type count—the number of distinct coset representatives that are populated.

In ring terms: the raw count is the L^1 norm of c_k restricted to the coset; the continuation count is the L^0 "norm" (support size). The continuation count sees the geometry of the support (how spread out is o across contexts?) while the raw count sees the measure (how much total mass does o have?).

5 Interpolation as the Tower of Ring Surjections

5.1 The Recursion on the Tower

The KN interpolation recursion:

$$P_k(o \mid c) = \frac{\max(c_k(c,o) - D, 0)}{c_k(c, \cdot)} + \frac{D \cdot \tau_k(c)}{c_k(c, \cdot)} \cdot P_{k-1}(o \mid c \bmod 256^{k-2}) \tag{15}$$

is a recursion along the tower of ring surjections. At each level:

1. **Evaluate on the current ring:** Look up $c_k(c, o)$ in $\mathbb{Z}/256^k\mathbb{Z}$. Subtract D. Normalize.
2. **Project to the next ring:** Compute $c \bmod 256^{k-2}$ (reduce mod the next-lower ring).
3. **Delegate the residual:** Pass the backoff mass to the prediction on the reduced ring $\mathbb{Z}/256^{k-1}\mathbb{Z}$.

Proposition 6 (KN Recursion = Tower Descent)**.** *The interpolated KN prediction is computed by descending the tower of rings:*

$$\mathbb{Z}/256^K\mathbb{Z} \xrightarrow[\textit{backoff residual}]{\bmod 256^{K-2}} \mathbb{Z}/256^{K-1}\mathbb{Z} \xrightarrow[\textit{backoff residual}]{\bmod 256^{K-3}} \cdots \xrightarrow[\textit{backoff residual}]{\bmod 1} \mathbb{Z}/256\mathbb{Z} \tag{16}$$

At each level, the prediction is a convex combination of the discounted counts (evidence specific to this ring) and the delegation to the next-lower ring (evidence from the coarser ring). The discount D controls the split: how much mass is "kept" (used at this level) vs. "passed down" (delegated to the coarser level).

5.2 What Each Level Contributes

At order k, the prediction uses the count $c_k(c, o)$ which lives on $\mathbb{Z}/256^k\mathbb{Z}$. This count contains:

- Information from the full $(k-1)$-byte context c (specific to this ring).
- Information from the shorter context $c \bmod 256^{k-2}$ (shared with the lower ring).

The discount removes the shared part (approximately—via subtraction rather than division). The residual $\max(c_k - D, 0)$ is the information specific to this level: what the full context c knows that the shorter context $c \bmod 256^{k-2}$ does not.

This is the ring-theoretic statement of the v1 paper's "common evidence removal": the common evidence lives on the lower ring (it is invariant under the projection), and the discount approximately removes it.

6 The Unified Ring

6.1 Patterns as Ring Elements

The unified pattern table from v1 becomes a function on the disjoint union of rings:

$$P_{\mathrm{KN}} \subseteq \bigsqcup_{k=1}^{K} \mathbb{Z}/256^k\mathbb{Z}. \tag{17}$$

Each pattern is an element $e \in \mathbb{Z}/256^k\mathbb{Z}$ for some order k, together with a weight $w_k(e)$ (the residual log-count).

The forward pass evaluates all rings simultaneously:

$$P(o \mid c) \propto \sum_{k=1}^{K} \underbrace{\alpha_k(c)}_{\text{backoff weight}} \cdot \underbrace{\frac{\max(c_k(c,o) - D, 0)}{c_k(c,\cdot)}}_{\text{discounted probability on ring } k} . \tag{18}$$

The backoff weights α_k are determined by the recursion: they distribute mass across the tower according to how much evidence each ring contributes.

6.2 The CRT Perspective

Although $\mathbb{Z}/256^k\mathbb{Z}$ does not factor via CRT (since $256 = 2^8$ and all factors share the prime 2), the tower *does* have a CRT-like structure when we factor differently.

Consider the event $e = 256c + o$ as an element of $\mathbb{Z}/256^k\mathbb{Z}$. The information in e decomposes as:

$$e \bmod 256 = o \quad \text{(output byte)}, \qquad \lfloor e/256 \rfloor = c \quad \text{(context)}. \tag{19}$$

This is not CRT (since 256 is not coprime to 256^{k-1}), but it is the mixed-radix decomposition. In terms of the ring: $\mathbb{Z}/256^k\mathbb{Z}$ has a filtration $0 \subset 256\mathbb{Z}/256^k\mathbb{Z} \subset 256^2\mathbb{Z}/256^k\mathbb{Z} \subset \cdots$ whose successive quotients are each $\cong \mathbb{Z}/256\mathbb{Z}$.

Each quotient corresponds to one byte position: the output, the most recent context byte, the next-most-recent, etc. The KN tower descends this filtration from the top (full context) to the bottom (output only).

6.3 Extending the Ring with Word Events

The ring $\mathbb{Z}/256^k\mathbb{Z}$ encodes byte-level contexts. To add word-level events, we extend the ring.

Let W be a word vocabulary of size $|W|$. A word event is an element of $\mathbb{Z}/|W|\mathbb{Z}$. The extended event space is:

$$E_{\mathrm{ext}} = \mathbb{Z}/256^k\mathbb{Z} \times \mathbb{Z}/|W|\mathbb{Z} \cong \mathbb{Z}/(256^k \cdot |W|)\mathbb{Z} \tag{20}$$

where the isomorphism holds when $\gcd(256^k, |W|) = 1$ (CRT applies; choose $|W|$ coprime to 256, e.g., $|W| = 65537$, a Fermat prime).

Proposition 7 (Words as an Independent Ring Factor)**.** *If $|W|$ is coprime to 256, the word event space is independent of the byte event space (in the CRT sense). The word identity $w = e \bmod |W|$ and the byte context $c = e \bmod 256^k$ can be recovered independently from the joint event e. No "combination" is needed—CRT gives the decomposition for free.*

This is the algebraic resolution of the combination problem: by choosing $|W|$ coprime to 256, the word and byte event spaces are guaranteed to be independent ring factors, and the CRT provides the exact decomposition. The ad-hoc mixing that failed in the match-model experiments is replaced by exact algebraic factoring.

7 The Discount–GCD Gap, Resolved

7.1 Three Operations on Counts

On a row of the count table $(c_1, c_2, \ldots, c_m)$ (the counts for context c across outputs $o_1, \ldots, o_m$), three operations compete:

Operation	Formula	Effect on ratios
KN discount (subtract D)	$c_i \mapsto c_i - D$	Distorted (small c_i lose more)
GCD division (divide by g)	$c_i \mapsto c_i/g$	Preserved (all shrink by $1/g$)
Log-subtract (-1 in log)	$c_i \mapsto c_i/2$	Preserved (all halve)

Theorem 8 (Resolution of the Gap)**.** *The discount–GCD gap has three components:*

1. ***Subtraction vs. division.*** *KN subtracts; the ring operation is division. These agree when $g = 1$ and $D \leq 1$ (the common case). The gap is nonzero only for rows with $g > 1$.*

2. ***Global vs. per-row.*** *KN uses a single D for all rows; the GCD is per-row. The global D is a compromise. The optimal D^* minimizes the mean divergence between the subtracted and divided distributions.*

3. ***Real vs. integer.*** *D is a real number; g is a positive integer. The fractional discount $D \in (0, 1)$ has no ring interpretation—it is an artifact of the estimation procedure. The integer GCD is the ring-native operation.*

The gap is small because: (a) $g = 1$ for most rows (component 1 vanishes), (b) the per-row GCDs are concentrated at 1 (component 2 is small), and (c) $D \approx 0.9 \approx 1 = g$ for the typical row (component 3 is small).

7.2 The UM-Native Version

The ring-native KN model would:

1. Compute $g(c)$ per row.

2. Divide: $r(c, o) = c_k(c, o)/g(c)$.

3. Backoff mass $= c_k(c, \cdot) - \sum_o r(c, o) \cdot g(c) = c_k(c, \cdot)(1 - 1/g(c)) \cdot g(c)$ counts worth.

4. Continuation count: unchanged (it is already ring-native).

5. Recursion: unchanged in structure; only the discount step changes from subtraction to division.

For rows with $g = 1$ (the vast majority), $r(c, o) = c_k(c, o)$ and zero mass is removed—backoff comes entirely from the continuation-count term. For rows with $g > 1$, more mass is removed (proportionally, preserving ratios), and more is delegated to the lower ring.

8 What the Ring Structure Reveals

The restatement on the integers makes several structural facts visible:

1. **The tower is a filtration.** The KN order tower is the filtration of $\mathbb{Z}/256^K\mathbb{Z}$ by powers of 256. This is not a choice—it is the unique filtration by the prime $p = 2$ (since $256 = 2^8$). The "order" of an n-gram is its position in the 2-adic filtration.

2. **Context is a p-adic integer.** An infinite context (the full history) is a 256-adic integer: an element of the inverse limit $\varprojlim_k \mathbb{Z}/256^k\mathbb{Z} = \mathbb{Z}_{256}$. The KN model at order K approximates this by truncating to K digits. The "infinite-order" model works in $\mathbb{Z}_{256}$ directly.

3. **The discount is a derivation.** Subtraction of a constant from each count is an additive perturbation of the count function: $\tilde{c}_k = c_k - D \cdot \mathbf{1}_{c_k > 0}$. In ring terms, this is c_k minus D times the indicator function of the support. The indicator function of the support is the *radical* of the count function (in the sense of radical ideals: $\sqrt{(c_k)}$). The discount is subtraction of (a multiple of) the radical.

4. **Interpolation is a weighted inverse limit.** The full KN prediction at order K is a weighted average over all levels of the tower. In the inverse limit, this becomes a weighted sum over all digits of the 256-adic expansion, with weights determined by the discount and continuation counts.

5. **The combination problem is CRT.** Adding new event spaces (words, phrases, match contexts) to the byte ring requires either:

   - Coprime extension: $|W|$ coprime to 256, giving CRT factorization (clean, independent).
   - Non-coprime extension: $|W|$ sharing factors with 256, giving a non-split extension (tangled, requires careful handling).

   The coprime case is the algebraically clean solution to the combination problem.

9 Research Agenda (Updated)

1. **Per-row GCD discount.** Implement and benchmark against global D. Prediction: identical for most rows ($g = 1$), better for high-frequency rows ($g > 1$).

2. **Ring-native forward pass.** Implement the count lookup as ring arithmetic: context = integer, projection = mod 256^{k-2}, pattern match = equality in the ring. Measure whether this simplifies or speeds up the implementation.

3. **CRT word extension.** Choose $|W| = 65537$ (Fermat prime, coprime to 256). Encode word events as $e \bmod 65537$. Test whether CRT decomposition gives clean combination without the mixing catastrophes of v1–v19.

4. **256-adic analysis.** Treat the full history as a 256-adic integer. Investigate whether p-adic analysis (continuity, derivatives, integration in $\mathbb{Z}_{256}$) gives useful tools for analyzing KN-like models.

5. **Spectral methods.** The Fourier transform on $\mathbb{Z}/256^k\mathbb{Z}$ (Pontryagin dual) decomposes the count function into characters. Investigate whether peaks in the character spectrum correspond to linguistic patterns (repeated k-grams, periodic structure).

10 Conclusion

Kneser–Ney smoothing, restated on the integers:

KN Component	Ring Operation
Context of order k	Integer $c \in \mathbb{Z}/256^{k-1}\mathbb{Z}$
Joint event	Integer $e = 256c + o \in \mathbb{Z}/256^{k}\mathbb{Z}$
Order projection	$c \bmod 256^{k-2}$ (modular reduction)
Count table	Function $c_k : \mathbb{Z}/256^{k}\mathbb{Z} \to \mathbb{N}$
Discount	Subtraction (approx. division by g)
GCD	gcd in $\mathbb{Z}$
Continuation count	$\lvert S \cap \rho^{-1}(o)\rvert$ (fiber support size)
Interpolation	Weighted tower of ring surjections
Backoff	Delegation to coarser ring via mod
Full history	256-adic integer $\in \mathbb{Z}_{256}$
Word extension	CRT factor ($\lvert W\rvert$ coprime to 256)

Every entry in this table is a mathematical identity, not an analogy. The integers are the events; the rings are the event spaces; modular reduction is the projection; division is the discount. The combination problem dissolves into CRT when word vocabularies are chosen coprime to 256.

The v1 paper asked "what is the discount–GCD gap?" The answer: subtraction vs. division in $\mathbb{Z}$, which is small because most rows have gcd = 1. The v1 paper asked "how do we combine KN with word models?" The answer: CRT, by choosing $\lvert W\rvert$ coprime to the byte alphabet size.

References

[1] Claude and MJC. *Kneser–Ney as Quotient: Pulling n-gram Smoothing into the Universal Model.* Hutter archive, 16 Feb 2026. (v1 of this paper.)

[2] Claude and MJC. *Integer Factorization of Events: Every Integer Is an Event, Every Quotient Is Division.* Hutter archive, 17 Feb 2026.

[3] Claude and MJC. *Scaling Byte-Level Kneser–Ney to 1.78 bpc on enwik9.* Hutter archive, 12 Feb 2026.

[4] Claude and MJC. *Match Models, Sparse Contexts, and the Combination Problem.* Hutter archive, 16 Feb 2026.

[5] Claude and MJC. *Bayes from Counting.* Hutter archive, 12 Feb 2026.