Kneser–Ney as Quotient: Pulling n-gram Smoothing into the Universal Model

Claude and MJC

February 16, 2026

Abstract

Interpolated Kneser–Ney (KN) smoothing achieves 1.784 bpc on enwik9 with byte-level 6-grams and zero structure [**?**]. We have tried to combine KN with other models via mixture approaches, with "mixed results" [**?**]—the combination problem is unsolved because KN lives outside the UM framework, requiring ad-hoc mixing. We pull KN back into the UM by showing that its three core operations—the discount, the backoff distribution, and the interpolation recursion—are quotient operations on the joint count table $c(\text{context}, \text{output})$. The discount D approximates the removal of *common evidence* (the GCD of the Bayes-from-counting framework [**?**]): it subtracts evidence that the lower-order model already explains. The KN continuation count is the column-side type count—a quotient measuring each output's *generality* across contexts. The interpolation recursion is the hierarchical combination rule [**?**] applied to a tower of decreasing-order models, each a projection of the one above. With KN inside the UM, the combination problem dissolves: KN patterns at different orders coexist in a single pattern table P, and extending E with lexical events [**?**] gives them the word-level structure they currently lack.

1 The Event Spaces of n-gram Prediction

An order-k byte-level n-gram model has event space $E_k = I_k \times O$, where:

$$I_k = \{0, \ldots, 255\}^{k-1} \quad \text{(context: previous } k{-}1 \text{ bytes)}, \tag{1}$$

$$O = \{0, \ldots, 255\} \quad \text{(output: next byte)}. \tag{2}$$

The joint event $(c, o) \in E_k$ records "context c was followed by byte o." The count table is:

$$c_k(c, o) = |\{t : (b_{t-k+1}, \ldots, b_{t-1}) = c,\ b_t = o\}|. \tag{3}$$

This is exactly the log product pattern (LPP) of the UM applied to E_k: each entry $s_k(c, o) = \lfloor \log_2 c_k(c, o) \rfloor$ is an atomic pattern weight.

Proposition 1 (The n-gram count table is an LPP)**.** *The order-k count table $c_k : I_k \times O \to \mathbb{N}$ is the output of the standard learning function ω_0 applied to the event space E_k. Each nonzero entry is an atomic pattern $(c, o, s_k(c, o))$ in the pattern space $P_k \subseteq E_k^2 \times T$.*

For a full KN model of order K, we have a tower of event spaces:

$$E_K \twoheadrightarrow E_{K-1} \twoheadrightarrow \cdots \twoheadrightarrow E_2 \twoheadrightarrow E_1, \tag{4}$$

where each projection $\pi_k : E_k \to E_{k-1}$ drops the oldest byte from the context:

$$\pi_k(b_1, \ldots, b_{k-1}, o) = (b_2, \ldots, b_{k-1}, o). \tag{5}$$

2 The Order Projection as Quotient

Definition 2 (Order projection)**.** *The projection $\pi_k : I_k \to I_{k-1}$ identifies contexts that differ only in their oldest byte:*

$$[c]_{k-1} = \{c' \in I_k : \pi_k(c') = \pi_k(c)\} = \{(b, c_2, \ldots, c_{k-1}) : b \in \{0..255\}\}. \tag{6}$$

Each equivalence class has $|[c]_{k-1}| = 256$ elements (one per value of the dropped byte).

Proposition 3 (Projection as UM quotient)**.** *The order projection π_k is a quotient in the UM sense: it maps a finer event space (I_k, which distinguishes 256 more contexts) onto a coarser one (I_{k-1}, which pools them). The quotient $Q_k = |I_k|/|I_{k-1}| = 256$ measures the compression at each step.*

The count tables are related by marginalization:

$$c_{k-1}(\pi_k(c), o) = \sum_{c' \in [c]_{k-1}} c_k(c', o). \tag{7}$$

The lower-order count is the sum over the equivalence class—the standard UM marginalization over a projected event space.

Remark 4 (The tower is the factorization tower)**.** *The tower (**??**) is a factorization tower in the sense of the nested model [**?**]: each level is a UM with its own $E_k = I_k \times O$, and the projection π_k is a factor map between adjacent levels. The full KN model interpolates predictions from all levels—but in the UM, these should be patterns in a single unified pattern table, not separate models mixed externally.*

3 The Discount as Common Evidence Removal

The interpolated KN prediction at order k is:

$$P_k(o \mid c) = \frac{\max(c_k(c, o) - D, 0)}{c_k(c, \cdot)} + \frac{D \cdot \tau_k(c)}{c_k(c, \cdot)} \cdot P_{k-1}(o \mid \pi_k(c)), \tag{8}$$

where $c_k(c, \cdot) = \sum_o c_k(c, o)$ is the total count for context c, $\tau_k(c) = |\{o : c_k(c, o) > 0\}|$ is the type count, and $D \in [0, 1]$ is the discount.

The discount subtracts D from each nonzero count. What does this mean in quotient terms?

Definition 5 (Common evidence in the GCD framework)**.** *The Bayes-from-counting framework [**?**] decomposes each row of the count table via the GCD:*

$$c_k(c, o) = g(c) \cdot r(c, o), \qquad g(c) = \gcd_{o : c_k(c,o) > 0} c_k(c, o), \tag{9}$$

where $g(c)$ is the common evidence*—the largest integer that divides every count in the row—and $r(c, o)$ is the* differential evidence *(the reduced counts, coprime across the row).*

The conditional probability depends only on the reduced counts: $P(o \mid c) = r(c, o)/R(c)$ where $R(c) = \sum_o r(c, o)$. The GCD cancels.

Proposition 6 (The discount approximates GCD removal)**.** *The KN discount D approximates the removal of common evidence in the following sense. Subtracting D from each nonzero count gives:*

$$c_k(c, o) - D = g(c) \cdot r(c, o) - D. \tag{10}$$

When $D = g(c)$ (the per-row GCD), this removes exactly one unit of common evidence:

$$c_k(c, o) - g(c) = g(c) \cdot (r(c, o) - 1). \tag{11}$$

For contexts where $\min_o c_k(c, o) = 1$ (the minimum count is 1, so $g(c) = 1$), this removes one occurrence of each continuation, leaving the reduced counts minus one.

KN uses a single global D rather than the per-row $g(c)$, which is an approximation. The optimal D (Chen & Goodman, 1999) is:

$$D^* = \frac{n_1}{n_1 + 2n_2}, \tag{12}$$

where n_1, n_2 are the global counts of n-grams occurring exactly once or twice. For natural language, $D^* \approx 0.75$–0.9, which is close to the mean GCD: most rows in a byte-level count table have $g(c) = 1$ (because most contexts have at least one continuation with count 1), so the global D tracks the typical GCD well.

Remark 7 (Why D is not exactly the GCD)**.** *The GCD is an integer; D is a real number $\in (0, 1)$. The GCD removes evidence in integer units ("one packet of common evidence"); D removes a fractional amount. This fractional removal has no direct UM interpretation—it is an artifact of the estimation procedure.*

The UM-native version would use $g(c)$ directly: subtract the GCD from each count, leaving the reduced counts. But this creates a problem: for contexts with $g(c) > 1$, removing the full GCD can remove too much evidence, and the reduced counts may be too sparse for reliable estimation. KN's fractional D is a regularization that trades optimality for robustness.

4 The Continuation Count as Column Quotient

Standard KN differs from simple interpolation in its backoff distribution. Instead of using the raw unigram $P(o) = c(o)/N$, KN uses the *continuation count*:

$$c_{\mathrm{KN}}(o) = |\{c : c_k(c, o) > 0\}|, \tag{13}$$

the number of distinct contexts in which output o has appeared.

Proposition 8 (The continuation count is the column type count)**.** *In the count table $c_k : I_k \times O \to \mathbb{N}$, the continuation count $c_{\mathrm{KN}}(o)$ is the column type count: the number of nonzero entries in column o. This is the column-side analogue of the row type count $\tau_k(c)$.*

Theorem 9 (The continuation count as column quotient)**.** *The continuation count measures the* generality *of output o across contexts—a quotient in the UM sense:*

$$Q_{\mathrm{gen}}(o) = \frac{|I_k|}{c_{\mathrm{KN}}(o)}. \tag{14}$$

Low generality quotient (many distinct contexts) means o is "general"—it appears in diverse situations, like the space character. High generality quotient (few distinct contexts) means o is "specific"—it appears only in restricted contexts, like a rare byte value.

The KN backoff distribution normalizes the continuation count:

$$P_{\mathrm{KN}}(o) = \frac{c_{\mathrm{KN}}(o)}{\sum_{o'} c_{\mathrm{KN}}(o')}, \tag{15}$$

which is the type-count-based estimate of the marginal. This estimates how likely o is to appear in a new *context—i.e., one not yet in the count table—rather than how often it appears overall.*

Remark 10 (Why continuation counts work)**.** *Raw counts are dominated by frequency: the space character has the highest $c(o)$ because it appears most often. But much of this count comes from a few very common contexts ("the ", "of ", etc.). The continuation count gives each context equal vote, regardless of how many times it appeared. This is a de-biasing operation—it removes the effect of context frequency, leaving only the diversity of contexts in which o appears.*

In UM quotient terms: the raw count $c(o)$ conflates the common evidence *(how often each context occurs) with the* differential evidence *(how broadly o is distributed). The continuation count extracts the differential evidence by counting types rather than tokens. This is a column-side GCD-like operation: it factors out the context-frequency effect.*

5 Interpolation as Hierarchical Combination

The KN recursion (**??**) interpolates between orders. We now show this is the hierarchical combination rule from the pattern-space framework [**?**].

Theorem 11 (KN interpolation as hierarchical combination)**.** *The interpolated KN recursion has the form:*

$$P_k(o \mid c) = \underbrace{\frac{\max(c_k(c,o) - D, 0)}{c_k(c,\cdot)}}_{\textit{high-order residual}} + \underbrace{\frac{D \cdot \tau_k(c)}{c_k(c,\cdot)}}_{\textit{backoff weight}} \cdot P_{k-1}(o \mid \pi_k(c)). \tag{16}$$

This is a two-stage hierarchical combination:

1. ***Stage 1 (high-order residual)****: Use the high-order count, discounted by D, to predict o. This is the* differential evidence *from the full context c—the part that the lower-order model does not capture.*

2. ***Stage 2 (backoff)****: For the remaining probability mass ($D \cdot \tau_k(c)/c_k(c,\cdot)$), delegate to the lower-order model P_{k-1}. This is the* common evidence *that is equally well explained by the shorter context $\pi_k(c)$.*

The two stages condition on different information: Stage 1 uses the full context c (all $k-1$ bytes); Stage 2 uses only the shorter context $\pi_k(c)$ ($k-2$ bytes). The dropped byte c_1—the oldest byte in the context—is the information that separates them.

Proof. The backoff weight $\alpha_k(c) = D \cdot \tau_k(c)/c_k(c,\cdot)$ sums to $1 - \sum_o \max(c_k(c,o) - D, 0)/c_k(c,\cdot)$, so the two terms form a proper probability distribution. The high-order term uses the full context c; the backoff term marginalizes out the oldest byte by delegating to P_{k-1}.

This matches the hierarchical combination rule: information flows from the full context through the high-order model (Stage 1), and the residual (unexplained by the full context) is handled by the lower-order model (Stage 2), which has absorbed the shorter context via its own counting. The two stages are approximately independent because the high-order residual captures what the oldest byte adds, while the backoff captures what is predictable without it. □

Remark 12 (The full recursion is a factorization tower)**.** *Applying the recursion from order K down to order 1 gives a tower:*

$$P_K \xrightarrow{\textit{residual + backoff}} P_{K-1} \xrightarrow{\textit{residual + backoff}} \cdots \xrightarrow{\textit{residual + backoff}} P_1. \tag{17}$$

Each step projects to a coarser event space (shorter context) and uses the residual/backoff decomposition to separate high-order from low-order evidence. The full prediction is a weighted sum over all orders, with each order contributing its differential evidence.

This tower is the order-projection factorization tower *of the n-gram model. In the UM, the whole tower should be a single pattern table P over the union of event spaces $\bigcup_k E_k$, with patterns at each order contributing to the same forward pass.*

6 The Unified Pattern Table

Definition 13 (The KN pattern table)**.** *We define the UM pattern table for KN as the union of patterns at all orders:*

$$P_{\text{KN}} = \bigcup_{k=1}^{K} P_k, \tag{18}$$

where P_k contains atomic patterns $(c, o, w_k(c, o))$ for each context $c \in I_k$ and output $o \in O$ with $c_k(c, o) > 0$.

The weight $w_k(c, o)$ is NOT the raw log count. It is the residual *weight after the quotient decomposition:*

$$w_k(c, o) = \log_2 \max(c_k(c, o) - D, 0), \tag{19}$$

which is the log of the discounted count—the differential evidence from order k after removing the common evidence D.

Proposition 14 (No double-counting in the unified table)**.** *The unified pattern table P_{KN} avoids double-counting because each order contributes only its* residual *evidence:*

- *Order k contributes what the full context $c \in I_k$ explains* above *the shorter context $\pi_k(c) \in I_{k-1}$.*
- *The discount D removes the common evidence between adjacent orders.*
- *The backoff weight distributes the removed mass to the lower-order patterns.*

This is the safe combination principle [?]: the evidence at each order is the residual after accounting for lower-order evidence, and residuals at different orders are (approximately) independent because they condition on disjoint information (the oldest byte at each level).

Remark 15 (The problem with the current implementation)**.** *Our current KN implementation [?] stores counts in a hash table and computes the interpolated prediction procedurally at each position. It is not a UM: there is no event space, no pattern table, and no forward pass. The match-model experiments [?] tried to combine KN with context matching via external mixing—this is the "mixing $\neq$ extending" problem [?] applied to the counting model itself.*

Pulling KN into a unified pattern table solves the combination problem: the KN patterns and the match-model patterns (and eventually the word-level patterns) all live in the same P, and the forward pass combines them via max-min *without ad-hoc mixing weights.*

7 The Discount–GCD Gap

The central open question: what exactly is the relationship between D and the per-row GCD $g(c)$?

Conjecture 16 (The discount–GCD correspondence)**.** *The optimal KN discount D^* minimizes the expected divergence between the discounted distribution and the GCD-reduced distribution:*

$$D^* = \arg\min_D \sum_c P(c) \cdot \text{KL}\left(\frac{c_k(c, \cdot) - D}{c_k(c, \cdot) - D\tau_k(c)} \,\middle\|\, \frac{r(c, \cdot)}{R(c)} \right), \tag{20}$$

where $r(c, o) = c_k(c, o)/g(c)$ are the GCD-reduced counts and $R(c) = c_k(c, \cdot)/g(c)$ is the reduced total.

When the discount equals the GCD ($D = g(c)$ per-row), the KL divergence is zero for that row. The global D is the best single number that approximates the per-row GCDs.

Proposition 17 (Empirical GCD distribution)**.** *For byte-level n-gram count tables on natural language:*

- *The vast majority of rows have $g(c) = 1$ (at least one continuation appears exactly once).*
- *Rows with $g(c) > 1$ are typically high-frequency contexts ("th", "he", etc.) where all continuations are well-observed.*
- *The optimal global $D \approx 0.8$–0.9 is consistent with $g(c) = 1$ for most rows, since subtracting $D < 1$ from a count of 1 gives $\max(1 - D, 0) = 1 - D > 0$—the continuation survives with reduced weight.*

The GCD-based discount would use $D = g(c)$ per row, which equals 1 for most rows and > 1 for high-frequency contexts. This is essentially modified KN (which uses different D values depending on count), but derived from the quotient structure rather than estimated from held-out data.

Remark 18 (The un-worked-out detail)**.** *The exact mapping from KN's discount to the quotient framework has subtleties:*

1. *The GCD operates on* integer *counts; the discount operates on counts as* real *numbers. The tropical approximation (min vs GCD) from [?] mediates between these, but the gap is nonzero.*
2. *The discount is applied* before *normalization; the GCD cancels* after *normalization. These are not the same operation. Subtracting D and then normalizing gives a different distribution than dividing by $g(c)$ and then normalizing (the latter gives the same distribution—the GCD cancels).*
3. *The continuation count in the backoff distribution has no direct GCD analogue. It is a type count (column-side), while the GCD operates on the count magnitudes. The connection between type counts and GCDs is unclear.*

These are the details that need to be resolved to make the discount–quotient correspondence precise. We suspect the resolution involves the two-sided *GCD decomposition from [?]: the row GCDs g_I and column GCDs g_O together determine the full structure, and the discount/continuation-count pair approximates the (g_I, g_O) pair.*

8 The UM-Native KN Model

We now sketch the target: a UM that achieves KN-level performance while remaining inside the framework.

Definition 19 (The UM-native n-gram model)**.** *The event space is the full tower:*

$$E = I_K \times O, \tag{21}$$

where $I_K = \{0..255\}^{K-1}$ is the order-K context. Lower-order contexts are projections: $I_k = \pi_{k+1} \circ \cdots \circ \pi_K(I_K)$.

The pattern table P contains atomic patterns at all orders:

$$P = \{(c_k, o, w_k(c_k, o)) : k = 1, \ldots, K,\ c_k \in I_k,\ c_k(c_k, o) > 0\}. \tag{22}$$

The weight w_k encodes the residual *evidence at order k: the part of the joint count that is not explained by order $k-1$.*

Proposition 20 (The forward pass recovers KN)**.** *With appropriate weights, the UM forward pass over P recovers the interpolated KN prediction. The* max *operation in the forward pass selects the best-supported syllogism at each order; the hierarchical structure of the weights (residuals at each order) ensures that evidence is not double-counted.*

Specifically, the prediction for output o given context c is:

$$t_{\text{out}}(o) = \max_k \min(t_{\text{in}}(c_k), w_k(c_k, o)), \tag{23}$$

where $t_{\text{in}}(c_k) = 255$ (the context is observed with certainty) and the max *ranges over all orders that have a pattern for this context-output pair.*

Remark 21 (The max vs interpolation gap)**.** *The UM forward pass uses* max *(take the best syllogism); KN uses interpolation (weighted average over all orders). These are not the same. The* max *operation selects the single best-supported order for each output o, while KN averages over all orders.*

For outputs that are well-supported at high order, max *and interpolation agree (the high-order evidence dominates). For outputs supported only at low order,* max *gives the low-order weight directly, while KN gives a weighted average that includes zero contributions from higher orders.*

The gap between max *and interpolation is the "tropical–integer gap" of [?]: the tropical semiring's* max *approximates the integer arithmetic's sum. For compression, the gap matters: interpolation spreads probability mass more evenly, which is better for arithmetic coding.*

This gap is the first implementation question: can we close it by using the log-stochastic form (sum of log-supports, i.e., the standard softmax) instead of the tropical max-min*? The answer should be yes—the log-stochastic UM forward pass uses* $\log \sum \exp$ *in place of* max*, recovering the interpolated form.*

9 Extending with Lexical Events

Once KN lives inside the UM, extending it with lexical events becomes natural.

Proposition 22 (KN + words in a single P)**.** *The extended event space [?] adds position, accumulator, bag-of-letters, and word events to H. The pattern table grows by the four families of new patterns [?]: recognition, prediction, language-model, and structural.*

The KN patterns (n-gram byte contexts $\to$ output bytes) coexist with the word-level patterns (word identity $\to$ predicted bytes, word bigrams $\to$ word identity) in a single P. The forward pass combines them via the hierarchical combination rule:

1. *Within-word prediction: the word-level patterns provide strong evidence for the remaining bytes once the word is identified. This evidence is* independent *of the byte-level KN patterns at distant offsets (different information sources).*

2. *Cross-word prediction: word bigram patterns provide evidence for the next word's identity, which then cascades into byte-level predictions via the prediction patterns. This evidence is independent of within-word byte patterns (the word boundary is a Markov blanket [?]).*

3. *Byte-level fallback: for positions where no word event is active (rare words, non-word tokens, XML markup), the KN patterns provide the baseline prediction.*

No external mixing is needed. The combination is internal to the forward pass over the unified P.

10 Research Agenda

1. **Compute the GCD distribution.** For our existing order-6 byte count table on enwik9, compute $g(c)$ for every context c. Characterize the distribution. Verify that most rows have $g(c) = 1$ and that the global $D^* = 0.9$ tracks the empirical GCD.

2. **Per-row discount.** Implement a version of KN that uses $D = g(c)$ per row (integer GCD discount) instead of a global D. Compare bpc against standard KN. This tests whether the quotient decomposition is a better discount than the global estimate.

3. **Build the unified pattern table.** Construct P as a hash table of (context, output, weight) triples, with weights set to the residual log-counts at each order. Implement the forward pass as a lookup + max over orders. Compare against the procedural KN implementation.

4. **Log-stochastic forward pass.** Implement $\log \sum \exp$ in place of max to close the tropical–integer gap. This should recover interpolated KN exactly (or within numerical precision).

5. **Add word events.** Inject the top-N words as events in H', with recognition, prediction, and language-model patterns. Measure Δbpc from the word-level patterns above the KN baseline. This is the first test of whether the unified approach beats the best ad-hoc mixture (1.634 bpc from match-model v12 [**?**]).

6. **Resolve the discount–GCD gap.** Work out the exact relationship between the two-sided GCD decomposition (g_I, g_O) and the KN discount/continuation-count pair (D, c_{KN}). This is the theoretical core.

11 Discussion

11.1 Why this matters

The combination problem has been our bottleneck since the match-model paper [**?**]: five of nine mixture strategies made things worse, and the best valid strategy gained only 0.05 bpc. The root cause is that external mixing is the wrong approach—it is the same mistake as the `the_inject` experiments [**?**], applied to the counting model rather than the lexical model.

Pulling KN into the UM dissolves the combination problem. Models don't need to be "combined"—they are all patterns in a single P, and the forward pass handles combination natively. The discount is not an ad-hoc parameter but a quotient operation that separates residual from common evidence. The continuation count is not a heuristic but a column-side type quotient measuring output generality.

11.2 The path to 1.0 bpc

Our current best is 1.634 bpc (KN-6 + extended match, logistic mixer [**?**]). The gap to 1.0 bpc decomposes as:

- ∼0.08 bpc: hash table saturation and higher order (addressable by larger HT or better data structure).
- ∼0.10 bpc: word-level patterns (addressable by the extended event space with word events in H').
- ∼0.45 bpc: longer-range dependencies, syntax, semantics (requires the factorization tower beyond word bigrams).

The UM-native KN + word events should capture the first two components, reaching ∼1.45 bpc. The third requires deeper nesting (the factorization tower [**?**]) and is the subject of future work.

References

[1] Michaeljohn Clement. *CMP*. `https://cmpr.ai/cmp.pdf`, 2026.

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

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

[4] Claude and MJC. *Bayes from Counting: Partial Quotients, GCD, and the Symmetric Learning Function on $E = I \times O$*. Hutter archive, 12 Feb 2026.

[5] Claude and MJC. *Patterns in the Extended Event Space: Independence, Correlation, and the New Synapses*. Hutter archive, 15 Feb 2026.

[6] Claude and MJC. *The Extended Event Space: Injecting Lexical Structure into H*. Hutter archive, 15 Feb 2026.

[7] Claude and MJC. *The Nested Model: Self-Similar Architecture in the Extended Event Space*. Hutter archive, 15 Feb 2026.