The Embedding Conjecture: Word Representations from Byte-Level Universal Models

Claude and MJC

February 2026

Abstract

We conjecture that word embeddings arise naturally from byte-level universal models (UMs) through three mechanisms: *causal capture* (compressing away spelling once word identity is determined), *forward inference* (propagating the compressed word representation to predict subsequent context), and *backward surprise response* (reversing the compression when downstream evidence demands it). The conjecture connects traditional word embeddings (as "20 questions" over a binary event space in 2^ℓ) to the UM framework where pattern chains from input bytes through hidden states to output predictions naturally form a word-level representation. We formalize the three mechanisms in UM terms, show that arithmetic coding provides the bridge between byte-level spelling and word-level identity, and identify causal capture as the point where the UM discards positional/spelling information in favor of semantic content.

1 Introduction

A word embedding maps a word w to a vector $\mathbf{e}(w) \in \mathbb{R}^d$ such that semantically similar words have nearby embeddings. Traditional embeddings (word2vec, GloVe) are trained on word co-occurrence statistics; modern language models learn embeddings as part of end-to-end training on character or subword sequences.

Our starting point is different: we have a byte-level universal model (UM) that predicts $P(b_t \mid b_1, \ldots, b_{t-1})$ for each byte b_t in a text stream. The UM has no notion of "words"—it sees only bytes. Yet the hidden state $\mathbf{h}_t$ of an RNN-based UM, or equivalently the pattern-chain statistics of a counting-based UM, necessarily encodes information about word boundaries, word identity, and even word meaning, because this information improves byte prediction.

Conjecture 1 (Embedding Conjecture)**.** *In any sufficiently powerful byte-level UM operating on natural language:*

(a) Causal capture*: As letters of a word are processed, the hidden state converges to a word-level representation that discards positional and spelling information. Pattern chains from individual letters do not propagate past the word embedding layer.*

(b) Forward inference*: The captured word representation predicts subsequent bytes (the next word, punctuation, structure) at word-level granularity, achieving compression gains beyond what byte-level n-grams provide.*

(c) Backward surprise response*: When downstream evidence contradicts the captured word identity (e.g., a garden-path sentence, a misspelling that changes meaning), the compression can be reversed to recover the original byte-level information, at a cost proportional to the surprise.*

2 Words as Binary Event Spaces

2.1 The 20-Questions Representation

A traditional word embedding in $\mathbb{R}^d$ can be viewed as a binary event space (ES) in 2^ℓ: each bit of the embedding answers one yes/no question about the word. For a vocabulary of V words, we need $\ell \geq \log_2 V$ bits. English has roughly $2^{17} \approx 131{,}000$ word forms, so $\ell \geq 17$ bits suffice for identity.

In the UM framework, each bit corresponds to a binary ES E_i with events e_i^+ ("yes") and e_i^- ("no"). The word embedding is the joint state across all ℓ binary ESs:

$$\mathbf{e}(w) = (e_1(w), e_2(w), \ldots, e_\ell(w)) \in \{0,1\}^\ell.$$

This is the *minimal* representation: ℓ bits for word identity. Practical embeddings use $d \gg \ell$ dimensions because the extra dimensions encode distributional semantics—not just which word, but what the word means in context.

2.2 Spelling as Additional Bits

A word's spelling is a sequence of bytes $s_1 s_2 \ldots s_k$. Given word identity w, the spelling is highly constrained: for most words, there is exactly one standard spelling, so the conditional entropy $H(s_1 \ldots s_k \mid w) \approx 0$.

For words with variant spellings ("color"/"colour", misspellings like "teh" for "the"), the spelling contributes a small number of additional bits. Via arithmetic coding, we can represent any spelling variant as the word identity plus trailing bits that encode the deviation from the canonical spelling:

$$\text{encode}(s_1 \ldots s_k) = \underbrace{\text{word\_id}(w)}_{\ell \text{ bits}} \parallel \underbrace{\text{variant\_bits}}_{\approx 0 \text{ bits (usually)}}$$

This is the bridge between byte-level and word-level representations: arithmetic coding compresses the byte sequence into the word identity plus minimal residual information.

3 Causal Capture in the UM

3.1 The Mechanism

Consider a byte-level UM processing the word "the":

- After "t": the hidden state encodes "starts with t". Many words are possible (the, that, this, there, ...). Pattern chains from "t" propagate forward.
- After "th": the state narrows to "th"-words. Pattern chains from "t" alone are less relevant; "th" patterns dominate.
- After "the": if followed by a word boundary (space, punctuation), the word is identified as "the". The state encodes the word identity.
- After "the ": the space confirms the word boundary. Pattern chains from individual letters ("t", "h", "e") no longer propagate—the state has *captured* the word "the" and discarded the spelling.

Definition 2 (Causal Capture)**.** *A position t in the byte stream exhibits* causal capture *for word w if the mutual information between the hidden state $\mathbf{h}_t$ and subsequent bytes $b_{t+1}, b_{t+2}, \ldots$ is mediated entirely through the word identity:*

$$I(\mathbf{h}_t; b_{t+1}^T) = I(w; b_{t+1}^T) + \epsilon$$

where $\epsilon \to 0$ as the UM becomes more powerful. That is, the hidden state carries no information about future bytes beyond what the word identity provides.

Causal capture does *not* mean the spelling information is destroyed. It means the UM has compressed the spelling into the word identity, and the residual (spelling-specific) information has been "captured" in the sense that it is no longer needed for forward prediction. The UM's hidden state at the word boundary is functionally equivalent to the word embedding plus a small residual.

3.2 Evidence from the Sat-RNN

Our factor map analysis of the 128-hidden-unit saturated RNN provides empirical support for causal capture:

- The `word_len` feature is the best predictor of every neuron's state, explaining $R^2 = 0.50$ of the total hidden state variance. This is the UM tracking position within the current word.
- At word boundaries, the hidden state undergoes a "reset" with $\|\Delta\mathbf{h}\| = 5.31$ (large compared to mid-word updates of $\sim$1.5). This reset is the causal capture event: the UM transitions from letter-tracking to word-level representation.
- The word length direction and tag-detection direction are entangled ($\cos = 0.50$), suggesting the word-level representation encodes both identity and structural role (is this word inside a tag?).
- Post-hoc subtraction of the word-length direction adds only +0.15 bpc, but step-by-step subtraction is catastrophic (+7.3 bpc), confirming that the word-tracking information is essential for the causal capture mechanism itself.

3.3 Pattern Chain Cutoff

In the counting-based UM (pattern chains from data), causal capture manifests as a cutoff in the pattern chain length distribution. For the word "the" at order 12, the pattern chain "e" $\to$ "h" $\to$ "t" $\to \cdots$ includes bytes from the current word. But the chain "e" $\to$ " " $\to$ "the" (crossing a word boundary backwards) is captured by the word "the" identity.

At order k greater than the average word length ($\sim$5 bytes), the pattern chains increasingly represent word-level rather than letter-level dependencies. This is why our sparse context patterns at offsets $\{1, 2, 4, 8\}$ (span 8) capture word-boundary structure and provide +0.089 bpc beyond KN-6.

4 Forward Inference

Once causal capture has compressed the current word into its embedding, forward inference uses this representation to predict subsequent bytes. The gain comes from word-level regularities:

- After "the", the next word is likely a noun, adjective, or adverb (not a verb, preposition, or another determiner). This narrows the first byte of the next word.
- After "United", the next word is very likely "States", "Kingdom", "Nations", or "Arab"—a tiny subset of the vocabulary.
- Inside XML tags, after "`<title>`", the content follows article-title distributions.

The forward inference gain is the difference between word-level and byte-level prediction:

$$\Delta_{\text{forward}} = H_{\text{byte}}(b_{t+1} \mid b_1^t) - H_{\text{word}}(b_{t+1} \mid w_1^j)$$

where w_1^j is the word sequence up to the current word boundary.

Our word bigram analysis (Section 3 of the Tock empirical paper) shows that word-level mutual information is 2.845 bits per word transition, equivalent to $\sim$0.10 bpc. This is the forward inference gain available in principle. Our current system captures +0.094 bpc total (sparse + match), suggesting that most of the word-level information is already being exploited by long-range models.

5 Backward Surprise Response

The most novel aspect of the conjecture concerns what happens when downstream evidence contradicts the captured word identity.

5.1 Garden-Path Resolution

Consider: "The old man the boats." After "The old man", the UM captures "man" as a noun (subject of the sentence). But "the boats" forces reinterpretation: "man" is a verb ("to man"), and "the old" are the subject (the elderly people).

In UM terms, the byte sequence "man " was causally captured as noun-"man". The surprise from "the boats" (low probability given the noun interpretation) triggers a backward information flow:

1. The high surprise ($-\log P$("the boats" | noun-man)) indicates the captured representation is wrong.
2. The UM must "un-capture" the word, recovering the byte-level spelling information.
3. The re-interpretation as verb-"man" produces a new embedding with much lower surprise for "the boats".

The energy cost of this reversal is proportional to the surprise: $\Delta E \propto -\log P(\text{continuation} \mid \text{wrong capture})$. This connects to the broader UM framework where surprise drives information-restoring processes.

5.2 Misspelling Propagation

A simpler example: "teh next word" where "teh" is a misspelling of "the". The UM processes:

- "t", "e", "h", " ": causal capture identifies this as likely "the" with a spelling variant. The embedding is word_id("the")$\|$variant("teh").
- The variant bits (encoding the transposition) are carried forward but do not affect prediction of the next word—"next" is equally likely after "the" or "teh".
- If downstream context makes the misspelling relevant (e.g., in a discussion about typos), the variant bits can be decoded.

5.3 Formal Connection to Arithmetic Coding

The backward surprise response has an exact formulation via arithmetic coding. The word identity determines a probability distribution over continuations. When the actual continuation falls in a low-probability region of this distribution, the arithmetic code requires extra bits to specify the actual outcome. These extra bits are precisely the information that was "discarded" during causal capture and must now be recovered.

Proposition 3 (Surprise Cost of Reversal)**.** *The cost of reversing a causal capture is*

$$C_{reversal} = -\log_2 P(\textit{actual} \mid \textit{captured}) + \log_2 P(\textit{actual} \mid \textit{correct})$$

bits, where "captured" is the (wrong) word identity from initial processing and "correct" is the revised interpretation.

6 Implications for UM Architecture

6.1 The Hybrid Character–Word Model

The embedding conjecture implies that the optimal UM is neither purely character-level nor purely word-level, but a hybrid that:

1. Processes bytes to build up word candidates (character → word)
2. Captures the most likely word identity at each boundary
3. Uses the word-level representation for forward prediction
4. Maintains the ability to reverse captures when surprised

This is exactly the architecture implied by the Tock lexicon notes: the UM's input space I includes byte-level events ("the input is 'e'") alongside time events ("the time step is 3"), and the hidden space H includes word-level events ("there is an 'e' in the current word"). The transition from I events to H events is the causal capture.

6.2 Connection to the Quotient Structure

The KN-quotient paper identified discount as GCD-based common evidence removal and interpolation as hierarchical combination. Causal capture is a related but distinct operation: it is a *projection* from byte-level to word-level event spaces, where the quotient of the byte-level pattern table by the word-level patterns yields the residual (spelling) information.

Formally, if P_{byte} is the byte-level pattern table and P_{word} is the word-level table, then:

$$P_{\text{byte}} = P_{\text{word}} \otimes P_{\text{spelling}}$$

where $\otimes$ is the UM combination operation and P_{spelling} is the residual. Causal capture discards P_{spelling} for forward prediction; backward surprise response recovers it.

7 Experimental Predictions

The embedding conjecture makes several testable predictions:

1. **Hidden state reset at word boundaries.** The sat-RNN hidden state should show larger $\|\Delta\mathbf{h}\|$ at word boundaries than within words. *Confirmed:* $\|\Delta\mathbf{h}\| = 5.31$ at boundaries vs. $\sim$1.5 mid-word.

2. **Word-level MI exceeds byte-level.** Mutual information between the hidden state at position t and bytes b_{t+k} for $k >$ word_len should be better predicted by word identity than by the byte sequence. *Partially confirmed:* word bigrams provide 2.845 bits/transition MI.

3. **Adding "the" to the UM is neutral.** Adding a binary ES for the word "the" (detected/not-detected) to the sat-RNN should not change predictions when properly marginalized, because the information is already encoded. This is the "break even" test for causal capture. *Not yet tested.*

4. **Lexicon breaks spurious long-range patterns.** The sat-RNN has learned long-range skip-gram patterns that are actually mediated by word identity. Adding an explicit lexicon and marginalizing out byte-level patterns should break these spurious connections without hurting (and potentially improving) prediction. *Not yet tested.*

5. **Synonym distance equals embedding distance.** In the UM-derived embedding, the number of extra bits needed to go from one word to a synonym should correspond to the cosine distance in traditional word embeddings. *Not yet tested.*

8 Conclusion

The embedding conjecture proposes that word embeddings are not an external construction imposed on language models but an inevitable consequence of optimal byte-level prediction. Any sufficiently powerful UM must develop word-level representations because they compress the byte stream more efficiently than pure byte-level prediction.

The three mechanisms—causal capture (compress spelling away), forward inference (predict from word identity), and backward surprise response (recover spelling when needed)—form a complete theory of how byte-level and word-level representations interact in a UM.

The immediate next step is the "adding 'the' " experiment: introduce a single word-level binary ES to the sat-RNN and verify that it breaks even when properly marginalized. Success would confirm that the UM already performs implicit causal capture; failure would indicate that the embedding conjecture requires modification.

References

[1] Claude and MJC, "Factor Map Analysis of the Saturated RNN," 2026.

[2] Claude and MJC, "Tock Phase Empirical Results," 2026.

[3] Claude and MJC, "Kneser–Ney as Quotient," 2026.

[4] Claude and MJC, "Match Models, Sparse Contexts, and the Combination Problem," 2026.