Open Research Questions for Archive 20260131_3

Hutter RNN Project

2026-01-31

1 What We Learned

1. ES features provide ≤ 0.19 bits/char (3.7%), not 53%
2. ES captures 15.9% of byte-level Markov mutual information
3. Max entropy renormalization loses 2.09 bits/char within-ES
4. Pattern extraction must use data statistics, not weight products
5. Current ES definition is poor: "Other" wastes 3 bits, "Whitespace" wastes 1.6 bits

2 Open Questions

2.1 Q1: Better ES Definitions

Current ESs: Digit(10), Punct(6), Vowel(10), Whitespace(4), Other(226).

"Other" is 88% of the alphabet but only captures residual structure. Should we split:

- Consonants (21) vs Other-other?
- Uppercase (26) vs lowercase?
- Brackets/parens vs other punctuation?
- High-frequency consonants (t,n,s,r) vs low-frequency?

Hypothesis: Optimal ES count is $\approx \log_2(256) = 8$ classes, balancing granularity vs complexity.

2.2 Q2: Hierarchical Coding

Instead of flat $P(\text{byte})$, use:

$$P(\text{byte}) = P(\text{ES}) \times P(\text{byte}|\text{ES})$$

Question: Does hierarchical prediction help the RNN, or is the bottleneck elsewhere?

2.3 Q3: Converged Performance

Our experiments used 1M chars (0.1% of data). Training is now at 1.6%.

Question: What is the actual improvement from ES features at convergence?

Prediction: ≤ 0.19 bits/char improvement, per Bayesian bound.

2.4 Q4: Learning Within-ES Structure

The 2.09 bit gap is within-ES structure that must be learned.

Question: How efficiently can an RNN learn within-ES patterns vs across-ES patterns?

Experiment: Compare learning curves for:

- Predicting ES given prev-ES (should be fast)
- Predicting byte-within-ES given byte-within-ES (should be slower)

2.5 Q5: Second-Order ESs

ES-pairs (ES_t, ES_{t+1}) have 25 states vs 5 for first-order.

Question: Do ES-pairs capture significantly more structure?

From our analysis:

$$H(ES_{t+1}|ES_t) = 1.46 \text{ bits} \quad \text{vs} \quad H(ES) = 1.65 \text{ bits}$$

Only 0.19 bits gained. But what about:

$$H(\text{byte}|ES_t, ES_{t+1}) \quad \text{vs} \quad H(\text{byte}|ES_t)?$$

2.6 Q6: Positional ESs

Hypothesis: Position within word matters. "e" at word-end behaves differently than "e" mid-word.

Question: Can we learn positional ESs automatically?

Approach: Cluster hidden states by (byte, position) and look for separable clusters.

2.7 Q7: The Actual Tock Problem

We assumed ESs are given. The real challenge: extract ESs from trained RNN.

Question: Which hidden neurons encode ES-like structure?

Approach:

1. Find neurons whose activation predicts byte class
2. Cluster neurons by what they predict
3. Interpret clusters as candidate ESs

2.8 Q8: Why Did Training Dynamics Differ?

The 53% early-training gap was real, even if not information-theoretic.

Question: Why did ES features stabilize training?
Hypotheses:

- Gradient normalization (ES features add constant-magnitude input)
- Easier credit assignment (ES → output is shorter path than byte → hidden → output)
- Regularization (ES features constrain hidden representations)

3 Proposed Experiments for 20260131_3

1. **Wait for convergence:** Let current training finish (ETA: 60 hours)
2. **Measure final gap:** Compare converged aug vs baseline bpc
3. **Try better ESs:** Split Other into 4 subcategories
4. **Second-order test:** Add ES-pair features, measure improvement
5. **Hidden state analysis:** Cluster activations, look for emergent ESs
6. **Ablation:** Remove individual ES features, measure impact

4 Summary Table

Claim	This Archive	Next Steps
ES improvement	3.7% max	Verify at convergence
ES explains	15.9% of MI	Try better ES defs
Within-ES gap	2.09 bits	Learn to close it
Pattern extraction	Data-based	Extend to trigrams?
Tock (ES from RNN)	Not attempted	Hidden state clustering