Theoretical Foundations: Information Theory and Complexity Constraints

ICF Estimation Project · 2025-12-06

This document establishes the theoretical foundations for character-level ICF prediction, including information-theoretic bounds, Kolmogorov complexity constraints, and fundamental questions about feasibility. We explore whether the goal is even possible, what constraints we must satisfy, and what the fundamental limits are.

0.1. The Fundamental Question

Is it even possible to build a model that compresses word→ICF mappings better than a dictionary while generalizing to unseen words? This question leads us to explore Kolmogorov complexity, information theory, and the structure of language itself.

0.2. Notation

- **K(x)**: Kolmogorov complexity of x (length of shortest program producing x)
- **K(M)**: Kolmogorov complexity of model M
- **K(D)**: Kolmogorov complexity of dictionary D (word ⟶ ICF mapping)
- **H(ICF)**: Shannon entropy of ICF distribution
- **I(X; Y)**: Mutual information between X and Y
- **MDL**: Minimum Description Length = K(model) + K(data | model)
- **V**: Vocabulary size
- **N**: Number of training samples

0.3. Kolmogorov Complexity Constraint

0.3.1. The Theorem

Kolmogorov Complexity is invariant up to an additive constant:

$$K_{U(x)} = K_{V(x)} + O(1)$$

for any two universal Turing machines U, V. This means K is well-defined (up to a constant).

0.3.2. Implication for Our Problem

- **K(ICF_function)** is well-defined (up to a constant)
- **K(dictionary)** = K(ICF_function) + O(1) (if dictionary is optimal encoding)
- **K(model)** = K(ICF_function) + K(architecture) + O(1)

0.3.3. Should K(model) = K(dictionary) by Definition?

If model perfectly learns ICF:

$$K(\text{model}) \approx K(\text{ICF\_function}) + K(\text{architecture})$$

$$K(\text{dict}) \approx K(\text{ICF\_function})$$

(if optimal encoding)

Therefore:

$$K(\text{model}) \approx K(\text{dict}) + K(\text{architecture})$$

(up to constants)

But there's a crucial distinction:

- **Dictionary**: Stores explicit mapping (word ⟶ ICF for seen words)

- **Model**: Stores implicit function (f: words → ICF for all words)
- **If ICF has structure**: K(model) < K(dict) possible (structure compresses)
- **If ICF is random**: K(model) ≈ K(dict) (cannot compress, must memorize)

Answer: No, they don't have to be the same. If ICF has structure, the model can be smaller because it encodes the structure, not just the mapping.

0.4. Compression Constraint

0.4.1. Dictionary Complexity

Uncompressed Dictionary:

$$K(D) = V \times (\text{avg\_word\_bytes} + 4 \text{ bytes})$$

For V=100k words: 900 KB

Compressed Dictionary:

- gzip: 3-4× compression → 225-300 KB for V=100k
- LZMA/xz: 4-6× compression → 150-200 KB for V=100k
- zstd/brotli: 4-5× compression → 180-225 KB for V=100k
- **Typical compressed size**: 180-250 KB for V=100k

Additional optimizations:

- **Sparse dictionary** (only rare words, ICF > 0.5): 50k words → 90 KB compressed
- **Trie/prefix tree** (share common prefixes): 60-70% of flat → 130 KB compressed

0.4.2. Model Complexity

Our model M: f(word) → ICF has:

$$K(M) = |\theta| \times 4 \text{ bytes} + K(\text{architecture})$$

- |θ| = 40k parameters: 160 KB
- K(architecture) ≈ few KB (fixed, reusable code)
- **Total: 160 KB + architecture**

0.4.3. The Constraint (Revised)

K(M) < K(D_compressed) for the model to be useful compression.

Our case (compressed dictionary):

- Model: 160 KB vs Compressed dict: 180-250 KB ✓ (satisfied, but marginal)
- Model: 160 KB vs Sparse dict: 90 KB ✗ (VIOLATED - sparse dict is smaller)
- Model: 160 KB vs Trie dict: 130 KB ✗ (VIOLATED - trie dict is smaller)

Critical insight:

- If dictionary is compressed, model advantage is **marginal (1.1-1.6×)**, not 5.6×
- If dictionary is sparse/trie-optimized, model may be **larger** than dictionary
- **Generalization is the key advantage**: Dictionary cannot handle unseen words, model can
- Model must compress AND generalize to justify its existence

0.5. Minimum Description Length (MDL) Principle

0.5.1. Definition

$$\text{MDL} = K(\text{model}) + K(\text{data} \mid \text{model})$$

Where:

- **K(model)**: Model complexity (160 KB)

- **K(data | model)**: Compressed training data given model
  - **1. -log P(data | model) (bits)**
  - ≈ training_loss × N (information content)

1.0.1. Optimal Model

Minimizes MDL = model_size + training_loss.

- **Too simple**: High K(data|model) (underfitting, high training loss)
- **Too complex**: High K(model) (overfitting, memorization)
- **Optimal**: Balance where MDL is minimized

1.0.2. For Our Case

$$\text{MDL}(\text{model}) = 160 \text{ KB} + \text{training\_loss} \times N$$

$$\text{MDL}(\text{dictionary}) = 900 \text{ KB} + 0$$

(perfect fit, no compression of data)

Model wins if: 160 KB + loss×N < 900 KB

This requires: loss×N < 740 KB

For N=50k: loss < 0.015 per example (very strict requirement)

Current status: Our training loss is 0.1-0.2, so loss×N ≈ 5-10 MB >> 740 KB. This suggests either:
1. Model is not compressing well (high K(data|model))
2. Need better regularization (reduce effective capacity)
3. Need more data (increase N to reduce per-example loss)

1.1. Information-Theoretic Lower Bound

1.1.1. Shannon Entropy

The Shannon entropy H(ICF) of the ICF distribution:

$$H(\text{ICF}) = -\sum_i P(\text{ICF}_i) \log_2 P(\text{ICF}_i)$$

- **If ICF values are random**: H(ICF) ≈ $\log_2(V)$ bits per word
- **If ICF has structure**: H(ICF) < $\log_2(V)$
- **For ICF in [0,1]**: H(ICF) ≤ $\log_2(V)$ (equality if uniform)

1.1.2. Model Capacity Requirement

Model must capture at least H(ICF) bits:

$$\text{Model capacity} \geq H(\text{ICF}) \times N$$

(for N words)

But can compress if structure exists.

Compression ratio: H(ICF) / actual_model_capacity

1.1.3. For Our Model

- **Capacity**: 40k params × 32 bits = 1.28 Mbits
- **If H(ICF) ≈ 10 bits/word** (structured, not uniform):
  - For N=50k: need 500k bits = 62.5 KB
  - Our model: 160 KB > 62.5 KB ✓ (sufficient capacity)

1.2. Maximum Spearman Correlation Bound

1.2.1. Information-Theoretic Bound

Maximum Spearman correlation is bounded by:

$$\rho_{\max} \leq \sqrt{\frac{I(X;Y)}{H(Y)}}$$

Where:
- **X** = Character features
- **Y** = ICF values
- **I(X; Y)** = Mutual information between characters and ICF
- **H(Y)** = Shannon entropy of ICF distribution

1.2.2. Formal Derivation

$$\rho^2_{\max} \leq \frac{I(X;Y)}{H(Y)} \leq \frac{H(X) - H(X|Y)}{H(Y)}$$

This follows from the data processing inequality and the relationship between correlation and mutual information.

1.2.3. Key Insight

If `I(Characters; ICF)` is low relative to `H(ICF)`, the maximum achievable correlation is inherently limited.

Hypothesis: Character-level features capture approximately 18-19% of ICF variance because:
- Character patterns ⟶ semantic frequency is an **indirect mapping**
- Many words with similar character patterns have different ICF values
- ICF depends on corpus/domain characteristics (not just characters)

1.3. Generalization Constraint

1.3.1. Requirement

Model must learn function f: words ⟶ ICF that:
- **Fits training**: f(word_i) ≈ ICF_i for i ∈ training set
- **Generalizes**: f(word_new) ≈ ICF_new for unseen words

1.3.2. Implies Regularity

This requires regularity in the ICF function:
- Words with similar patterns ⟶ similar ICF
- Morphological structure ⟶ frequency patterns
- Character sequences ⟶ rarity indicators

1.3.3. Model Capacity

Must be:
- **Sufficient**: Capture regularity (not too simple)
- **Limited**: Prevent memorization (not too complex)
- **Optimal**: Match complexity of true function

1.4. Summary: All Constraints

1.4.1. Explicit Constraints

1. **K(model) < K(dictionary_compressed)** [Kolmogorov complexity]
   - ✓ vs compressed dict (180-250 KB): 160 KB < 180-250 KB (marginal)
   - ✗ vs sparse dict (90 KB): 160 KB > 90 KB (violated)

- ✗ vs trie dict (130 KB): 160 KB > 130 KB (violated)

2. **MDL = K(model) + K(data|model) minimized** [MDL principle]
3. **Generalization: f(word_new) ≈ ICF_new** [generalization] ✓
4. **Data efficiency: N samples sufficient** [sample complexity] ⚠️
5. **Computational: fast inference** [speed constraint] ⚠️ (dict is 100-1000× faster)
6. **Storage: K(model) < K(dict_compressed)** [compression] ⚠️ (marginal)

1.4.2. Implicit Constraints

7. **Regularity: ICF has structure** [regularity assumption] ⚠️
8. **Capacity: VC_dim matches data** [capacity constraint] ⚠️
9. **Architecture: expressive but compact** [architecture design] ✓
10. **Information: H(ICF) < $\log_2(V)$** [entropy constraint] ⚠️

1.5. Why We Don't Use Nearest-Word Mapping

A natural question: why not just map unseen words to their nearest neighbors in our frequency dictionary? This approach is problematic because **frequency ≠ similarity**.

Consider these examples:
- "cat" (edit distance 1 from "bat") - very different frequencies
- "the" (most common) vs "thy" (rare, archaic) - similar form, opposite frequency
- "computer" vs "computers" - similar, but frequencies differ
- "run" vs "fun" - edit distance 1, but "run" is much more common

If we mapped unseen words to their nearest neighbors, we'd train the model with wrong frequency labels, teaching it incorrect patterns. Instead, we learn the structure directly from character patterns.

1.6. Key Insights

1. **The Kolmogorov complexity constraint is nuanced**:
   - **Uncompressed**: Model (160 KB) << Dictionary (900 KB) ✓ (5.6× advantage)
   - **Compressed**: Model (160 KB) vs Dictionary (180-250 KB) ⚠️ (1.1-1.6×, marginal)
   - **Sparse/Trie**: Model (160 KB) > Dictionary (90-130 KB) ✗ (violated)
2. **Compression matters critically**: Dictionary compression (3-6×) reduces K(dict) from 900 KB to 180-250 KB, making model advantage marginal.
3. **Generalization is the key differentiator**:
   - Dictionary: Only seen words (sparse coverage)
   - Model: Any UTF-8 string (dense coverage)
   - **This is why model is useful despite size constraints**
4. **MDL suggests issues**: High training loss means K(data|model) is large, so total MDL may not be optimal.
5. **Sample complexity is tight**: We have 50k samples but model has 40k params, suggesting we need either:
   - More data (increase N)
   - More regularization (reduce effective capacity)
   - Smaller model (reduce |θ|)

6. **Regularity assumption is critical**: If ICF has no structure, model cannot compress better than dictionary. We assume morphology/phonotactics predict frequency.
7. **Use case determines winner**:
   - **If only need seen words**: Dictionary wins (smaller, faster, exact)
   - **If need OOV/generalization**: Model wins (handles unseen words)
   - **If need both**: Hybrid approach (dict for seen, model for OOV)

1.7. Is Our Goal Even Interesting?

This is a question worth asking explicitly. What makes a goal "interesting"?

1. **Novelty**: Character-level frequency prediction is a novel approach, though frequency dictionaries exist.
2. **Practical utility**: OOV handling and RAG cost reduction are useful, though dictionaries work for seen words.
3. **Theoretical insight**: Understanding the structure of language/frequency relationships is interesting, though the structure may be weak.
4. **Compression ratio**: Uncompressed gives 5.6× advantage, but compressed gives only 1.1-1.6× (marginal).

Verdict: The goal is interesting IF structure exists and generalization is needed. The compression advantage is marginal, but generalization is the key differentiator - dictionaries cannot handle unseen words, models can.

1.8. The Regularity Assumption

For compression to work, the ICF function must have structure:

- **Regularity**: Similar words → similar ICF
- **Patterns**: Morphology/phonotactics → frequency
- **Redundancy**: Not all word→ICF pairs are independent

If ICF is random (no structure), then K(ICF) = V × 32 bits (cannot compress), and the model must memorize everything. If ICF has structure, then K(ICF) << V × 32 bits (can compress), and generalization is possible.

Current evidence: Spearman correlation of 0.18-0.19 suggests we're approaching the theoretical bound for character-level models. This could indicate either:

1. The structure is weak (character patterns provide limited information about frequency)
2. The model is not learning the structure (training issues)
3. We've reached the fundamental limit (information-theoretic bound)

This is an open question that motivates our research. The fact that we're consistently hitting 0.18-0.19 across multiple experiments suggests we may be hitting a fundamental limit rather than an optimization problem.