Shrink Files, Win Cash

The Hutter Prize Explained

Francesco Gargiulo | April 11, 2025

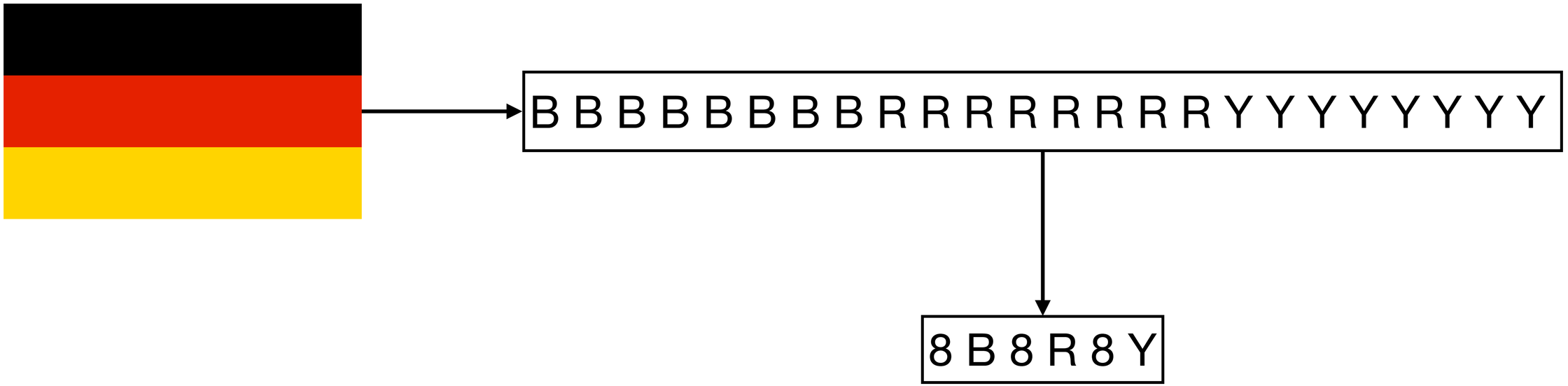
Motivation

- What if improving text compression is not just about reducing file sizes?
- Good compression requires good prediction – and good prediction requires understanding
- If a machine compresses Wikipedia better than anything else, what may this imply?
  - Understanding of patterns in language
  - Grasp of syntax and semantics
  - Ability of model context and handle ambiguity
  - Encoding of world knowledge

Primer on Data Compression

What is Data Compression?

- Finding shorter representations of data while preserving information
- Simple example: Run-Length Encoding



Randomness

- Consider these three sequences:

11111111111111111111…

31415926535897932384 6…

48192736451102837465 5…

- Would you say they “look random”?
- What makes them *fundamentally* different?

Kolmogorov Complexity

- K(x) = length of the shortest program that outputs string x
- Examples:
  - Sequence of all 1s: Low complexity (tiny loop generates it)
  - Digits of π: Low complexity (can be computed)
  - Random-looking sequence: High complexity (needs explicit storage)

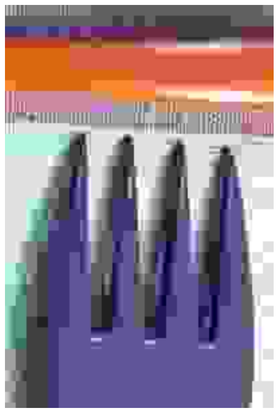
- Kolmogorov complexity is uncomputable

Lossless vs Lossy Compression

- Lossless: Reconstructs original data exactly
- Lossy: Reconstructs an approximation of original data (e.g., JPEG images)

Q99 - 54 KB

Q90 - 22 KB



Q1 - 2 KB

Codes and Codewords

- Let Σ be an alphabet of symbols (e.g., the English alphabet)
- A *code* C maps symbols $\sigma \in \Sigma$ to sequences of bits, named *codewords*
- Example: ASCII

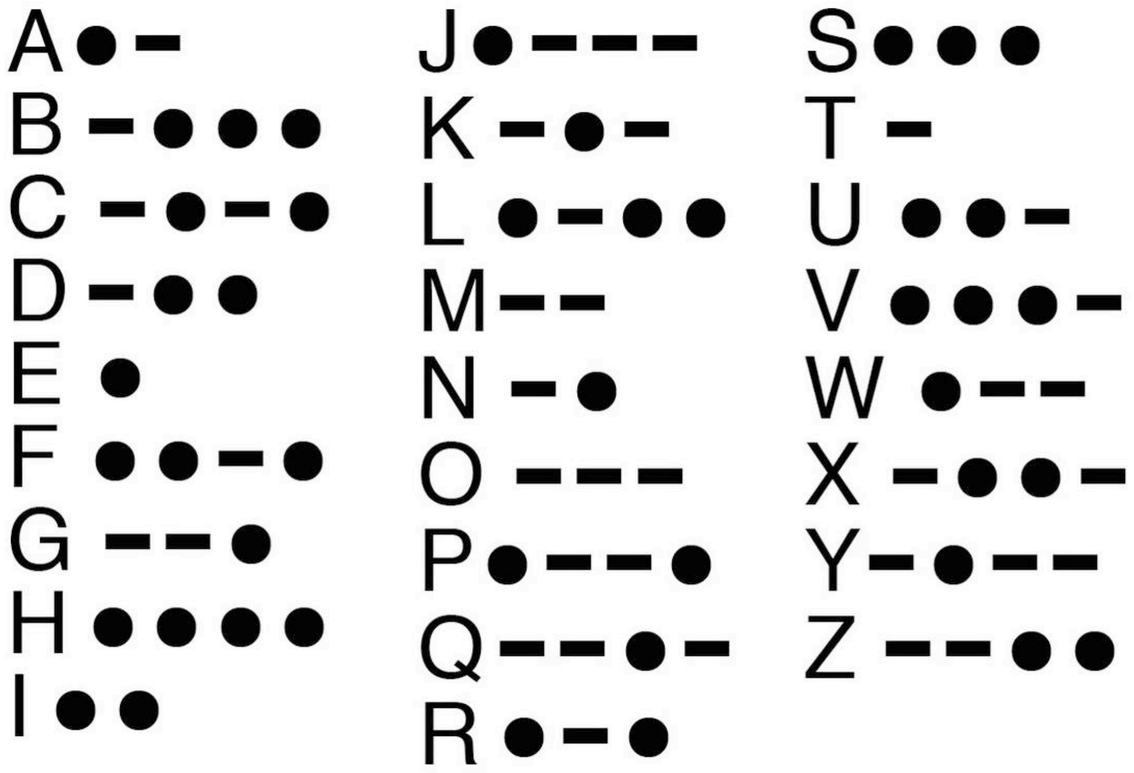
$$C('a') = (01100001)_2$$

Variable-Length Codes

- Problem: ASCII uses 8 bit per character regardless of frequency
- English letters appear with vastly different frequencies:
  - Common: ‘E’, ’T’, ‘A’
  - Rare: ‘J’, ‘Q’, ‘Z’
- Solution: Assign short codewords to frequent symbols, longer codewords to rare ones
- Requirement: Codewords must be uniquely decodable

Variable-Length Codes

Historical Example: Morse Code (1830s)



Optimal Codes and Contexts

- Given an alphabet Σ and a probability distribution P, an optimal code C is such that:

$$\forall \sigma \in \Sigma\,. \quad |\,C(\sigma)\,| = \log_2 \frac{1}{P(\sigma)}$$

- However, symbol probabilities depend on *contexts, e.g.,*

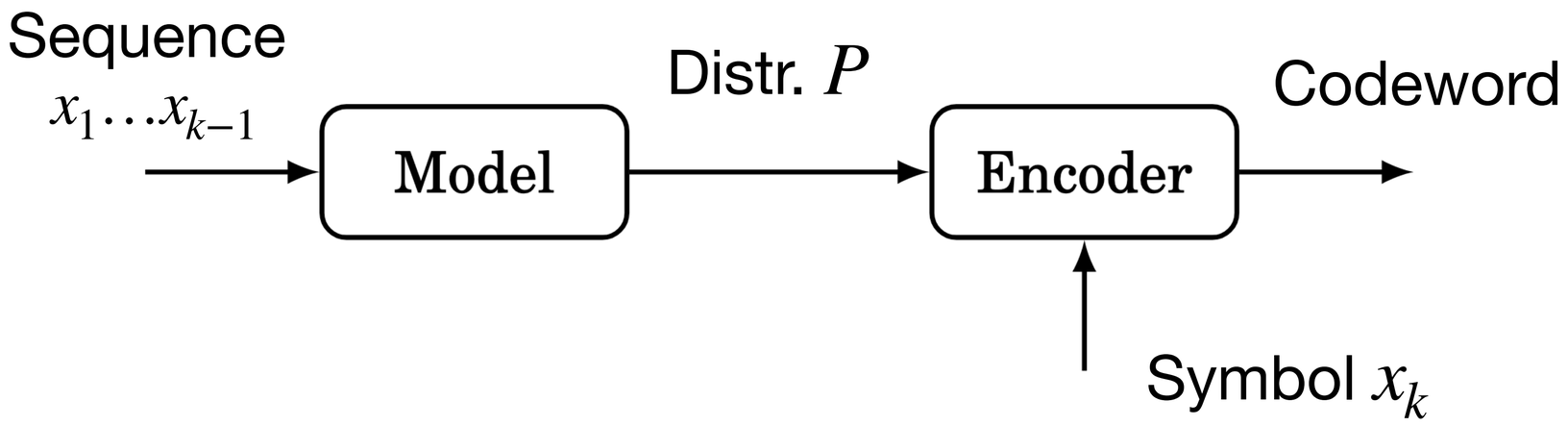
$$P(\text{"}s\text{"} \mid \text{"}breakfa\text{"}) \quad \text{and} \quad P(\text{"}e\text{"} \mid \text{"}breakfa\text{"})$$

- Solution: Use a different code for each context (and its probability distribution)

Statistical Data Compression

A statistical data compression algorithm has two stages:

1. *Modeling*: Analyze data to estimate probability distributions
2. *Encoding*: Use entropy coders (e.g., Huffman or Arithmetic Coding) to generate the compressed output



The Hutter Prize

Prediction ⇄ Compression

- Consider this language comprehension exercise:

  “*After the accident, they rushed him to the* _____”

- Why are you able to assign probabilities to words (e.g., 🥳 vs 🏥)?

- What are the implications of a machine accurately predicting words?

- Two related papers:

  1. Delétang, Grégoire, et al. "Language modeling is compression." arXiv preprint arXiv:2309.10668 (2023).

  2. Shannon, Claude E. "Prediction and entropy of printed English." Bell system technical journal 30.1 (1951): 50-64.

The Human Knowledge Compression Prize

- Challenge posed by Marcus Hutter in 2006:

http://prize.hutter1.net/

- Goal: Compress a 10^9 bytes long Wikipedia dataset (*enwik9*)
- How: Submit a self-extracting archive encoding *enwik9*
- Prize: 5000$ for every 1% improvement on the current best

FAQs

- Why Wikipedia?
- Why lossless compression?
- What about neural networks?
- Why self-extracting archives?

Previous Records

enwik9

Author	Date	Decompressor	Size (B)
Kaido Orav & Byron Knoll	3 Sep. 2024	fx2-cmix	110'793'128
Kaido Orav	2 Feb. 2024	fx-cmix	112'578'322
Saurabh Kumar	16 Jul. 2023	fast cmix	114'156'155
Artemiy Margaritov	31 May 2021	starlit	115'352'938
Alexander Rhatushnyak	4 Jul. 2019	phda9v1.8	116'673'681

enwik8

Author	Date	Decompressor	Size (B)
Alexander Rhatushnyak	4 Nov. 2017	phda9	15'284'944
Alexander Rhatushnyak	23 May 2009	decomp8	15'949'688
Alexander Rhatushnyak	14 May 2007	paq8hp12 -7	16'481'655
Alexander Rhatushnyak	25 Sep. 2006	paq8hp5 -7	17'073'018
Matt Mahoney	24 Mar. 2006	paq8f -7	18'324'887

Conclusions

Key Takeaways

- Compression and prediction are two sides of the same coin
- The Hutter Prize isn’t just about compressing files
  - An objective benchmark for machine understanding of human knowledge
  - A bridge between compression and modern AI

Thank You!