



Nils Holzenberger
& Jean-Louis Dossalles



Algorithmic information and artificial intelligence

Jean-Louis Dossalles
septembre 26

Chapter 2: Complexity
& Frequency

Telecom Paris



Information is compression

$Dist(\text{stove}, \text{refrigerator}) = 0.25$

3

- ~ Using frequency as a proxy for KC ←
- ~ Using KC as universal similarity measure
- ~ Using KC to explain frequencies

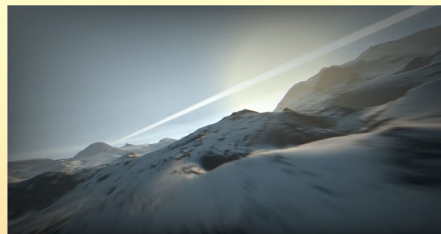
4

✓ What minimal size can a 3'30 video with realistic rendering of montain landscape + music be?

- A. 4 GB
- B. 40 MB
- C. 4 MB
- D. 400 kB
- E. 40 kB
- F. 4 kB



5



Demoscene

<https://www.youtube.com/watch?v=jB0vBmiTr6o>



6

✓ How many numbers can I represent with 10 bits?

- A. 2048
- B. 2047
- C. 1024
- D. 1023
- E. 512
- F. 511
- G. 42

7

✓ Many binary sequences (such as 00000...0 or 010101...01) are compressible.

What is the proportion of binary sequences of size 100 that are compressible by 10 bits?

- A. 10%
- B. 1 %
- C. 0.1 %
- D. 0.01 %
- E. 0.001 %

8

Most objects are complex

- (a) 11001100110011001100110011001100110011001100110011001100
- (b) 0010010000111111011010101000100010000101101000110000
- (c) 0010100101110110101101000101101011011010001011001011

proportion of sequences that are compressible by k bits: 2^{-k}

Number of compressible sequences:

$$1 + 2 + 2^2 + \dots + 2^{n-1} = 2^n - 1$$

Some finite sequences are strictly incompressible.

9

The compressor paradox

- ✓ One cannot compress "everything"
- ✓ Most objects are incompressible
- ✓ If at least one object gets compressed, then at least another one gets expanded!
- ✓ Iterating compression leads
 - either to a periodic sequence
 - or to an unbound sequence

proof.: the compressor can be seen as an injective function

10

Complexity and Frequency

✓ Simple objects are

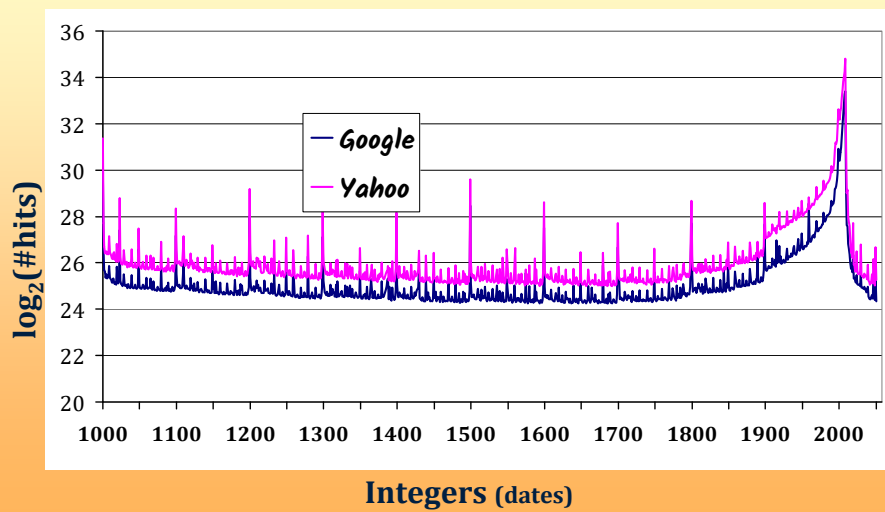
- A. More frequent
- B. Less frequent



11

Complexity and Frequency

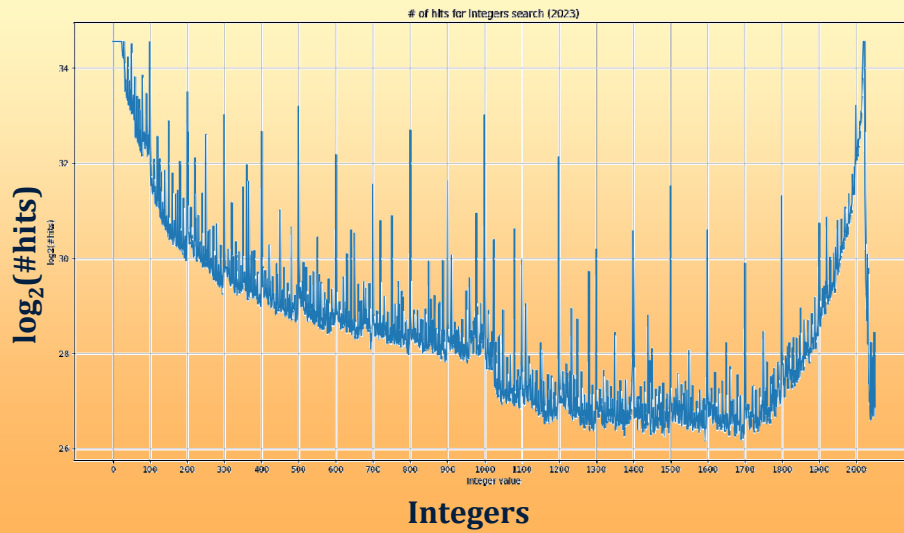
#hits for integers on two Web search engines in 2009



12

Complexity and Frequency

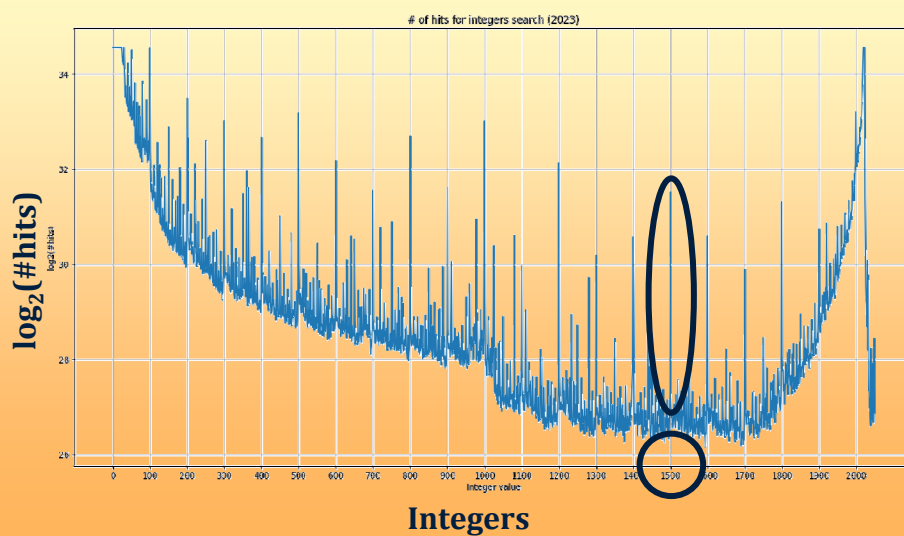
#hits for integers on Google in 2023



13

Complexity and Frequency

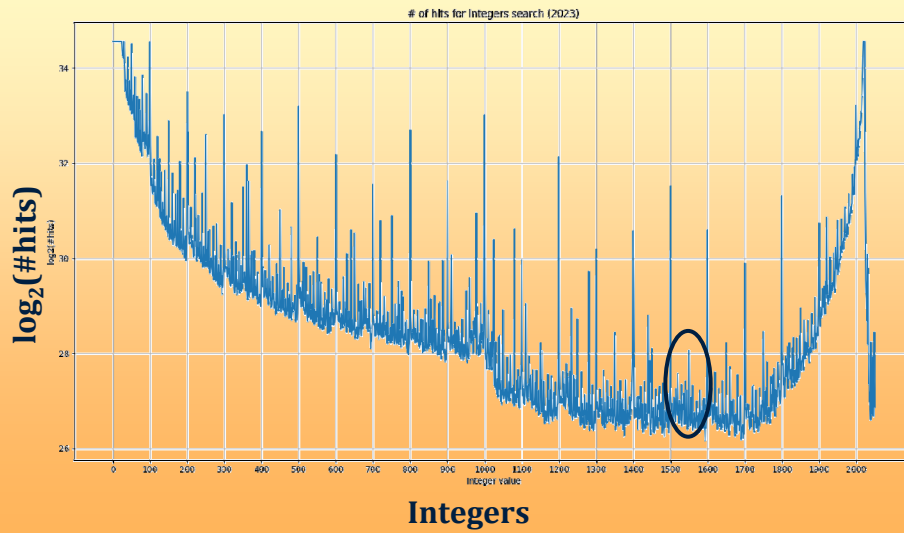
#hits for integers on Google in 2023



14

Complexity and Frequency

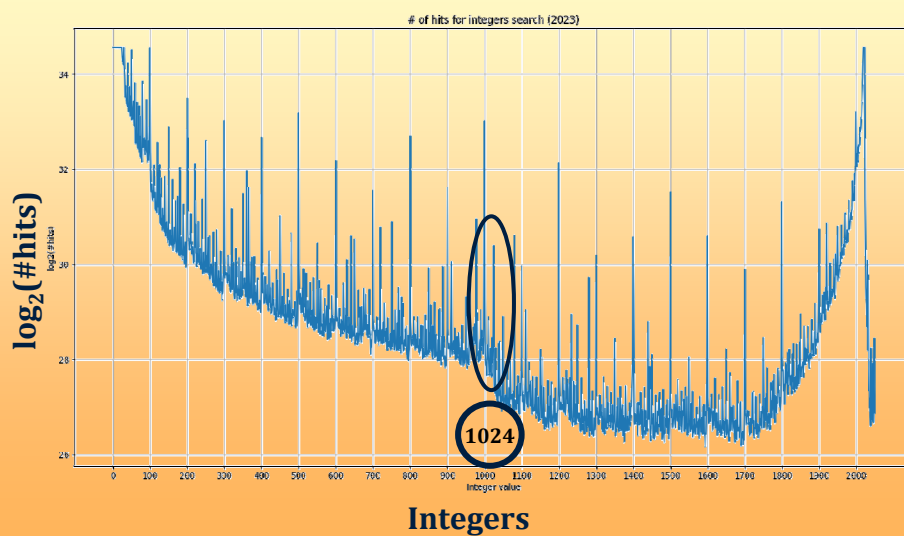
#hits for integers on Google in 2023



15

Complexity and Frequency

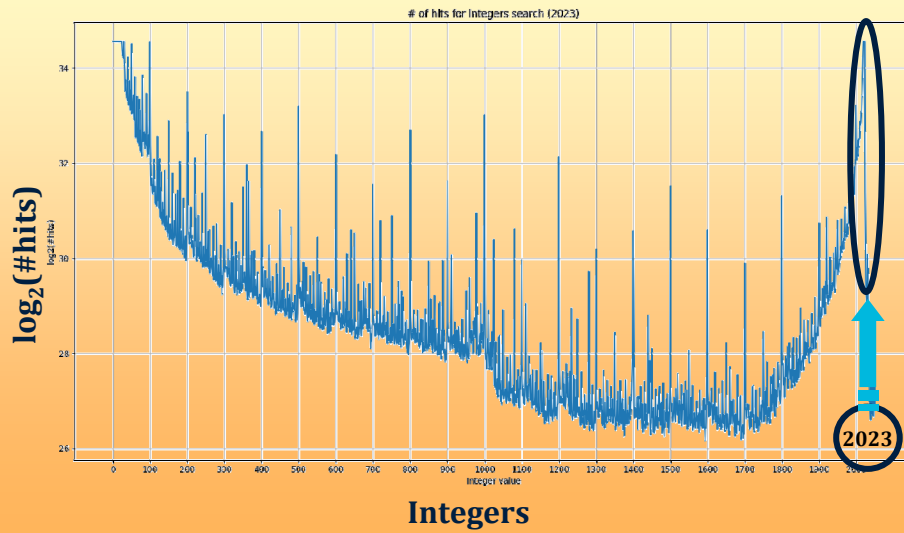
#hits for integers on Google in 2023



16

Complexity and Frequency

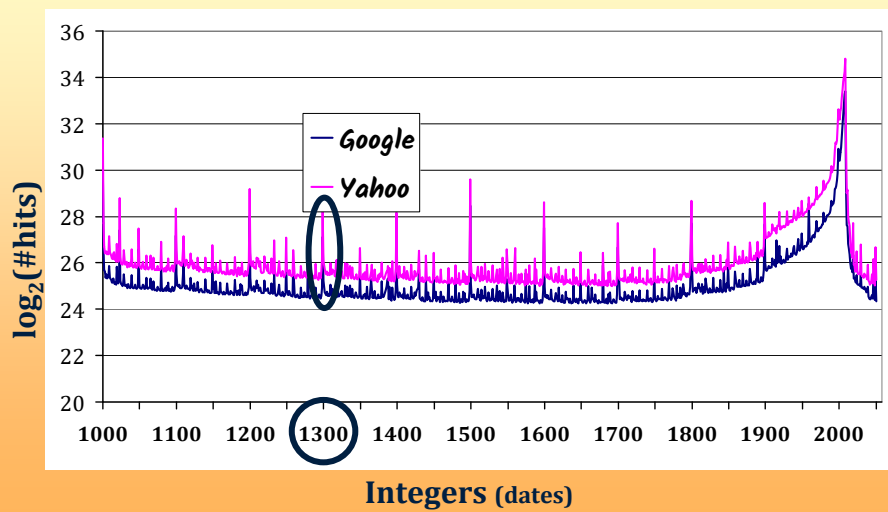
#hits for integers on Google in 2023



17

Complexity and Frequency

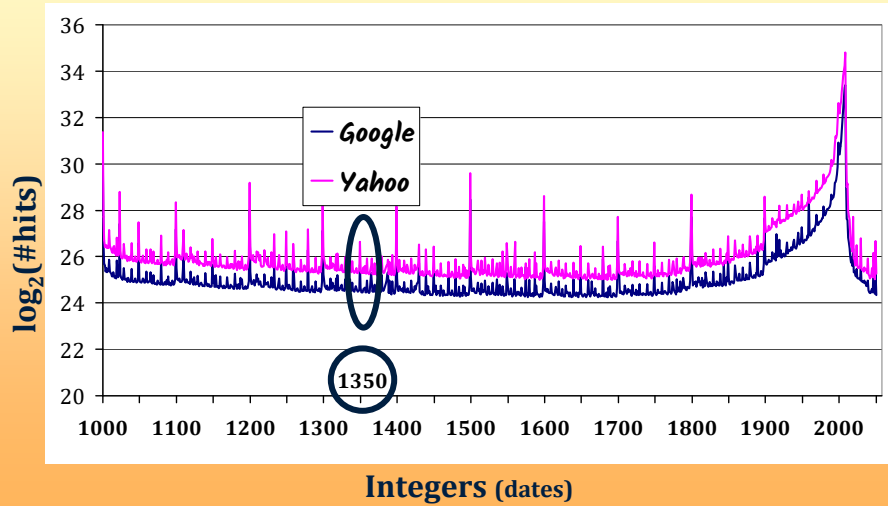
#hits for integers on two Web search engines in 2009



18

Complexity and Frequency

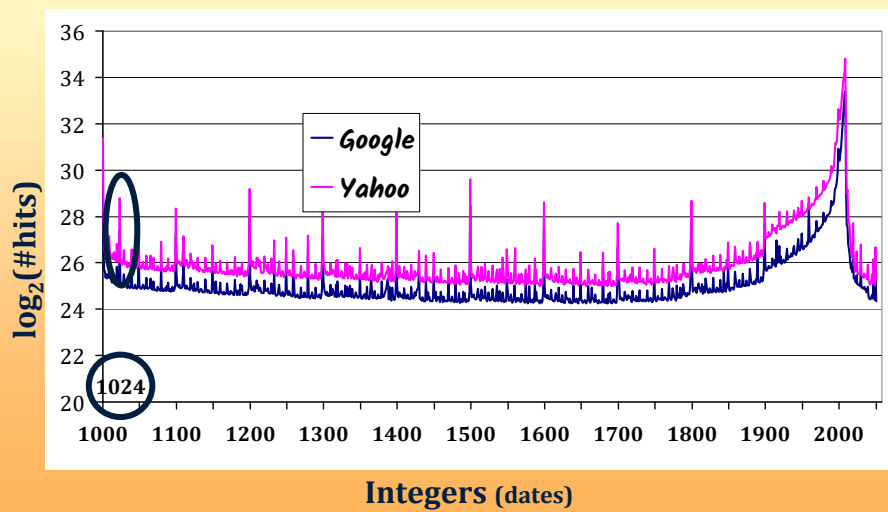
#hits for integers on two Web search engines in 2009



19

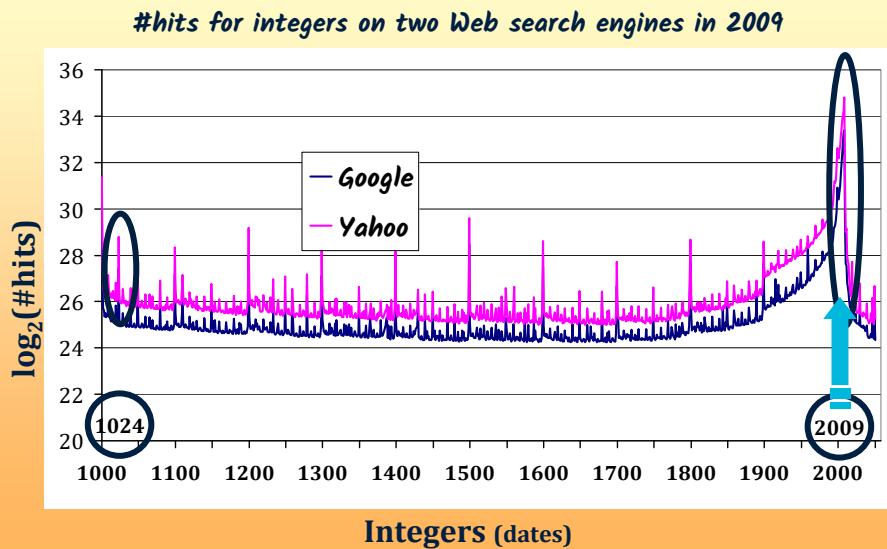
Complexity and Frequency

#hits for integers on two Web search engines in 2009



20

Complexity and Frequency



21

~ frequently occurring objects
should receive short codes

~ → frequent objects are simpler



22



Claude Shannon
1916-2001



mouse

ultimate

23



Claude Shannon
1916-2001



$$\text{code of } w = \log_2(1/\text{freq}(w)) + c$$



Shannon, C. E. (1948). *A mathematical theory of communication*. *The Bell System Technical Journal*, 27, 379-423,623-656.

24

Complexity and Frequency

~ Huffman codes

Huffman

Huffman generator

A Huffman code is built as a tree, from bottom to top.

- Nodes (in the case of texts) are characters or sets of characters with a probability.
- Initially, there are as many (disconnected) nodes as possible characters.
- Then the two lowest probability characters are connected through a parent node associated with the cumulated probability.
Then the connecting procedure starts anew with this new node replacing the two child nodes.

We end up with a fully connected tree in which lowest probability characters appear deepest.

The tree is then read from top to bottom, writing '0' when choosing the left child node and '1' when choosing the right child node.

This provides a prefix-code for each character.

25

~ Using frequency as a proxy for KC

~ Using KC as universal similarity measure

~ Using KC to explain frequencies



26

~ Why are simple strings more frequent on the Web?

A. Because ...

B. 42

27



Shannon-Fano code

$$\text{code of } w = \log_2(1/\text{freq}(w)) + c$$

address

$\approx K(w)$



word

w

The simplicity of concepts matches
a frequency-based code.

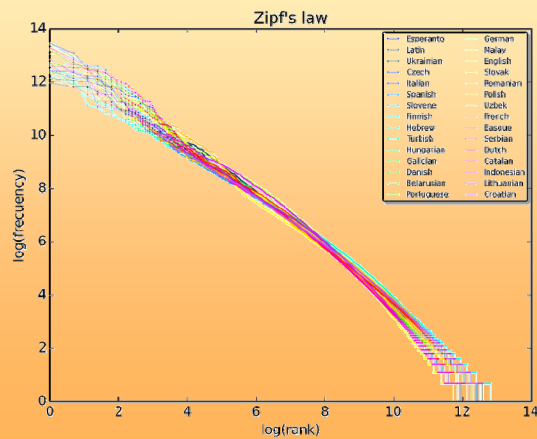
28

Loi de Zipf

Zipf

#3, #14

→ Stigler's law
of eponymy



29

~ Using frequency as a proxy for KC

~ Using KC as universal similarity measure



~ Using KC to explain frequencies

30

✓ How can we use compression to measure the resemblance between two objects A and B?

texts
viruses
any pair of objects

Similarity



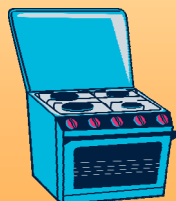
Clustering
Classification
Case-based reasoning
...

31

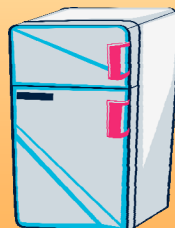
✓ How can we use compression to measure the resemblance between two objects A and B?

texts
viruses
any pair of objects

$Dist($



,



$) = 0.25??$



32

~ How can we use compression to measure the resemblance between two **texts** A and B ?

- A. Compare the sizes of $Z(A)$ and $Z(B)$
- B. Compare the size of $Z(A+B)$ with size of $Z(A) + \text{size of } Z(B)$
- C. Interleave A and B and compress what you get.

33

~ Step 1: evaluate $K(A|B)$

From the chain rule:

$$K(A | B) \geq \underbrace{K(A, B) - K(B)}$$

measures how much A resembles B

Approximation with compressor and concatenation:

$$Z(A+B) - Z(B)$$

34

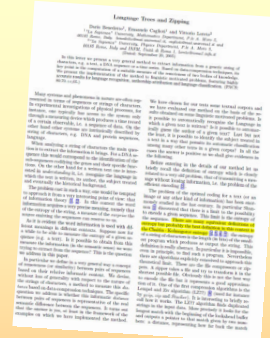
Compression as a proxy for Complexity

Benedetto, Caglioti & Loreto (2002)

$$\Delta_{Ab} = Z(A+b) - Z(A)$$

$$S_{AB} = (\Delta_{Ab} - \Delta_{Bb})/l(b)$$

AUTHOR	N. of texts	N. of successes 1	N. of successes 2
Alighieri	8	8	8
D'Annunzio	4	4	4
Deledda	15	15	15
Fogazzaro	5	4	5
Guicciardini	6	5	6
Macchiavelli	12	12	12
Manzoni	4	3	4
Pirandello	11	11	11
Salgari	11	10	10
Svevo	5	5	5
Verga	9	7	9
TOTALS	90	84	89



Benedetto, D., Caglioti, E. & Loreto, V. (2002).
Language trees and zipping.
Physical Review Letters, 88 (48702).

$$D_{AB} = (\Delta_{Ab} - \Delta_{Bb})/\Delta_{Bb} + (\Delta_{Ba} - \Delta_{Aa})/\Delta_{Aa}$$

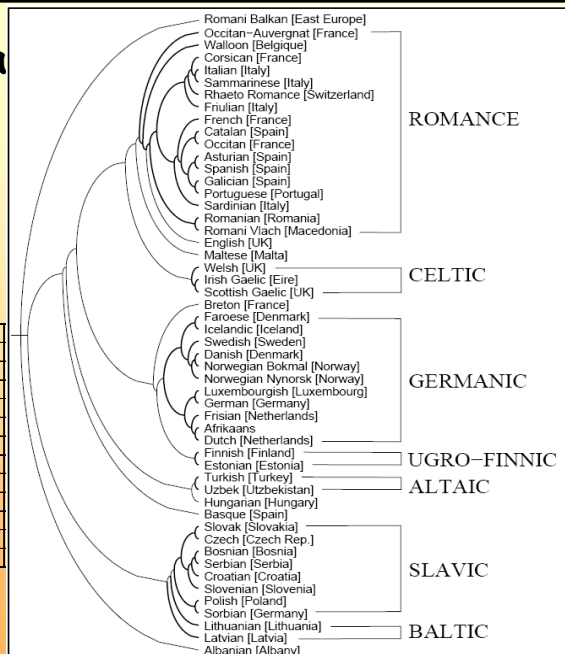
Compression as a Complexity

Benedetto, Caglioti & Loreto (2002)

$$\Delta_{Ab} = Z(A+b) - Z(A)$$

$$S_{AB} = (\Delta_{Ab} - \Delta_{Bb})/l(b)$$

AUTHOR	N. of texts	N. of successes 1
Alighieri	8	8
D'Annunzio	4	4
Deledda	15	15
Fogazzaro	5	4
Guicciardini	6	5
Macchiavelli	12	12
Manzoni	4	3
Pirandello	11	11
Salgari	11	10
Svevo	5	5
Verga	9	7
TOTALS	90	84



$$D_{AB} = (\Delta_{Ab} - \Delta_{Bb})/\Delta_{Bb} + (\Delta_{Ba} - \Delta_{Aa})/\Delta_{Aa}$$

NID: the "most intelligent" distance

~ Step 1: evaluate $K(x|y)$

~ From $K(x|y)$ to symmetrical similarity:

$$\max(K(x|y), K(y|x))$$

= amount of information that is *not shared* between x and y

~ Normalization

$$\max(K(x|y), K(y|x)) / \max(K(x), K(y))$$

~ Chain rule

replace $K(x|y)$ by $K(x, y) - K(y)$

37

Information distance

Information needed to tell x and y from each other.

$$ID(x, y) = \max(K(x|y), K(y|x))$$

$$ID(x, x) = 0; \dots$$

Normalized information distance

$$NID(x, y) = \frac{K(x, y) - \min(K(x), K(y))}{\max(K(x), K(y))}$$

Most "intelligent" distance

Bennett, C. H., Gács, P. & Li, M. (1998). [Information distance](#). *IEEE transactions on Information Theory*, 44 (4), 1407-1423.



38

Information distance

$$K(A | B) \geq K(A, B) - K(B)$$

$$ID(x, y) = K(x, y) - \min(K(x), K(y))$$

Normalized information distance

$$NID(x, y) = \frac{K(x, y) - \min(K(x), K(y))}{\max(K(x), K(y))}$$

Normalized compression distance

$$NCD(x, y) = \frac{Z(x, y) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

Vitanyi, P. & Cilibrasi, R. (2020).
[Phylogeny of the COVID-19 Virus](#)
[SARS-CoV-2 by Compression.](#)
 bioRxiv, 10.1101/2020.07.22.216242.

39

Normalized Google distance

$$NID(x, y) = \frac{K(x, y) - \min(K(x), K(y))}{\max(K(x), K(y))}$$



$$K(x) \approx -\log_2 \left(\frac{g(x)}{M} \right)$$

M = total number of indexed pages

g() = # pages

$$\text{code of } w = \log_2(1/\text{freq}(w)) + c$$

$$NGD(x, y) = \frac{\max(\log(g(x)), \log(g(y))) - \log(g(x, y))}{\log M - \min(\log(g(x)), \log(g(y)))}$$

Cilibrasi, R. & Vitányi, P. (2007).
 The Google similarity distance.
ACM Transactions on Knowledge and Data Engineering.
 Available at: <http://www.cwi.nl/~paulv/papers/amdug.pdf>



40

Normalized Google distance

$$NGD(x, y) = \frac{\max(\log(g(x)), \log(g(y))) - \log(g(x, y))}{\log M - \min(\log(g(x)), \log(g(y)))}$$

$g() = \# \text{ pages}$

$M = \text{total number of indexed pages}$ ← Zipf's law

$$Dist(\text{oven}, \text{fridge}) = 0.25$$

$$NGD(\text{oven}, \text{fridge}) = \frac{\log 674 - \log 176}{\log 60000 - \log 283} = 0.25$$

41

Normalized Google distance

$$NGD(x, y) = \frac{\max(\log(g(x)), \log(g(y))) - \log(g(x, y))}{\log M - \min(\log(g(x)), \log(g(y)))}$$

$g() = \# \text{ pages}$

$M = \text{total number of indexed pages}$

$$Dist(\text{oven}, \text{bird}) = 0.84$$

$$NGD(\text{oven}, \text{bird}) = \frac{\log 674 - \log 1.67}{\log 60000 - \log 48} = 0.84$$



42



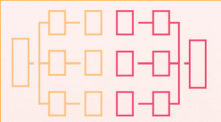
Information is compression

Using search engines as compressors

$$K(x) \approx -\log_2 \left(\frac{g(x)}{M} \right)$$

Using LLMs as compressors

Considering brains as compressors

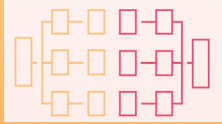


44



Information is
compression

Chiang, T. (2023). ChatGPT is a blurry JPEG of the Web. *Annals of Technology - The New Yorker*, Feb. Retrieved from <https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web>



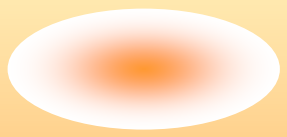
Using LLMs as compressors

Pan, Z., Wang, S. & Pengfei, L. (2026). *Understanding llm behaviors via compression*. *NEURIPS*, 38, 168404-168454. Retrieved from



CHATGPT IS A BLURRY JPEG OF THE
WEB
OpenAI's chatbot offers paraphrases, whereas Google offers quotes. Which do we prefer?

45



46