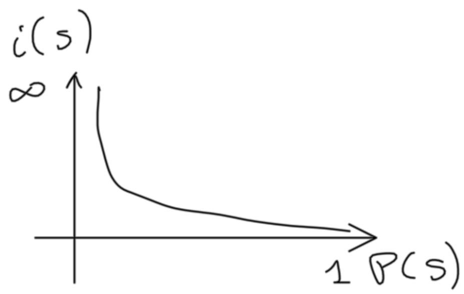
Medida probabilística de informação

$$i(s) = \log_2 \frac{1}{P(s)} = -\log_2 P(s)$$

símbolo do alfabeto

Se $P(s) = 1$, $i(s) = 0$

$P(s) = 0$, $i(s) = \infty$



Medida combinatória de informação

m objetos distintos (tamanho do alfabeto)

$$\underbrace{\text{informação é } \log_2(m) \text{ bits}}_{\text{Igualdade de Kraft}}$$

$P(A) \longrightarrow i(A) = -\log P(A)$

$P(B) \longrightarrow i(B) = -\log P(B)$

$P(A,B) \longrightarrow i(A,B) = -\log P(A,B) =$

$= -\log P(A) P(B) =$

$= -\log P(A) \cdot -\log P(B)$

$= i(A) + i(B)$

Média de info por símbolo ← $\sum_{s} i(s)P(s) = \underbrace{-\sum_{s} P(s) \times \log P(s)}_{\text{entropia}} = H$

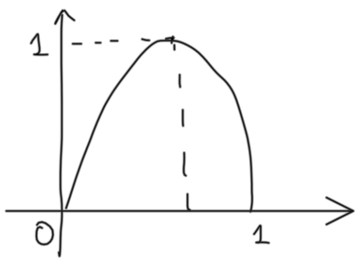
P(s) uniforme ⇒ entropia máxima

Alfabeto do tamanho m $\{s_1, s_2, \dots, s_m\}$

$$P_{s_i} = \frac{1}{m}$$

$$\underbrace{-\sum \frac{1}{m} \log\left(\frac{1}{m}\right)}_{} = \log m$$

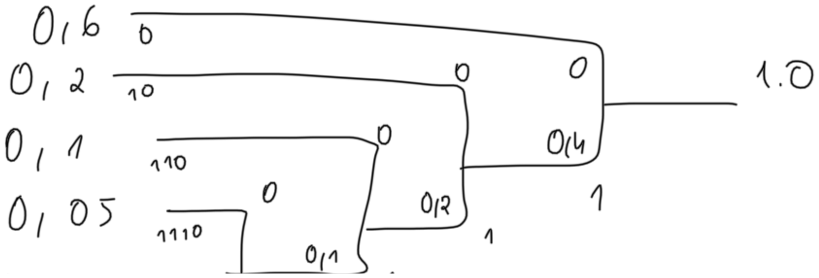
$$\frac{1}{m} \sum_{k=1}^{m} \log m =$$



$-p\log(p) - (1-p)\log(1-p)$

Códigos de Huffman

Ótimo na classe dos códigos de comprimento variável.



$0{,}05 \quad 1111 \rfloor_1 \quad 1$

Código de Golomb

Representa inteiros não negativos.

$n \in \mathbb{N}_0$

$q = n / m$

$r = n \% m$

q: código unário

r: código binário.

Exemplo:

$m = 4$

$n = 19$

$q = 19/4 = 4 \rightarrow 00001$

$r = 3 \rightarrow 11$

total $= \underbrace{0000}_{q} 1 \underbrace{11}_{r}$

$n = 9$

$q = 2 \rightarrow 001$

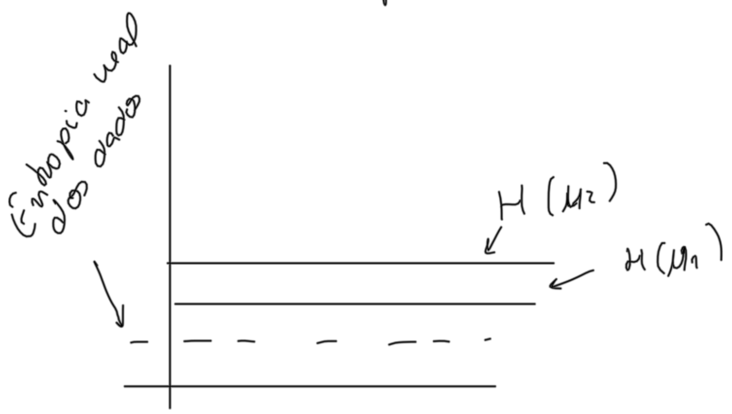
$r = 1 \rightarrow 01$

$\Big\} 00101$

x

$$H = - \sum_i p_i \log_2 p_i \quad \text{(bits / símbolo)}$$

Isto é a entropia 'verdadeira' se os símbolos ocorrerem de forma independente.



$$M_1 \longrightarrow P_i^1 \qquad\qquad M_2 \longrightarrow P_i^2$$

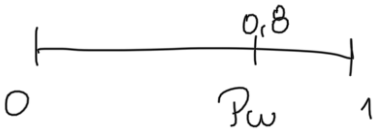
Se $H(M_1) < H(M_2)$ então M_1 é melhor que M_2.

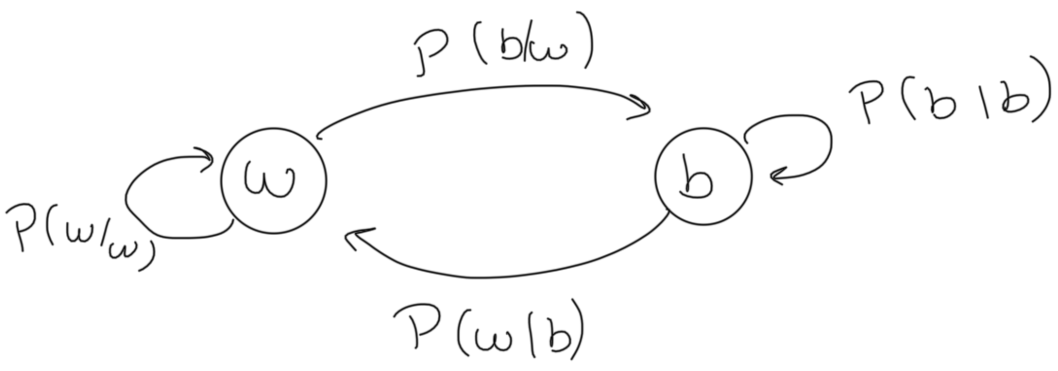
Modelo de Markov (de ordem k)

$$P(S \mid u_1 u_2 u_3 \dots u_n) = p(S \mid u_{n-k+1} \dots u_n)$$

u_{n+1}

$$P_b = 1 - P_w$$





$$p(w|b) + p(b|b) = 1$$

$p(w|b) = 0,3$

$p(b|w) = 0,01$

$$p(w) = \frac{p(w|b)}{p(w|b) \cdot p(b|w)}$$

$p(b|w) = 0,01$

$H_w = -p(w|w) \log p(w|w) - p(b|w) \log p(b|w)$

$H_b = p(b|b) \log p(b|b) - p(w|b) \log p(w|b)$

$H = H_w P_w + H_b P_b$

Se considerar independentes

$H = 0,204$ bits/símbolo

Se considerar Markov ordem 1

$H = 0,107$ bits/símbolo

Σ : alfabeto

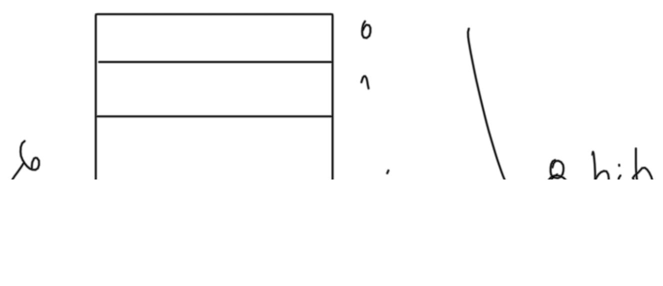
$|\Sigma| = 32$

Palavras de 4 símbolos

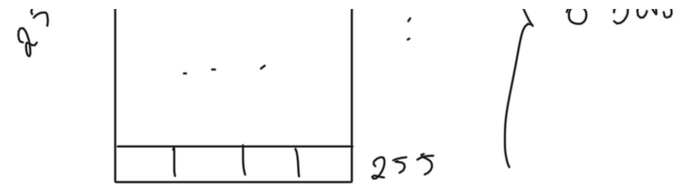
$|\Sigma|^4 = 32^4 \simeq 1M$

Probabilidade de uma palavra aleatória $\simeq 10^{-6}$

Escolho as 256 mais freq.



p: prob de estar no dicionário..



- Envia o idx no dicionário se a seq. estiver no dicionário (1)

- Envia 20 bits caso contrário (2)

$$L = 9p + 21(1-p) < 20$$

compensa neste caso

nº de bits médio

$p \geq 0{,}084$

$$\Sigma = \{A, B\}$$

(1)

AAA	00
AAB	01
AB	10
B	11

(2)

AAA	00
ABA	01
AB	10
B	11

(1) $\underbrace{AAB}_{01}\ \underbrace{B}_{11}\ \underbrace{AB}_{10}\ \underbrace{AAB}_{01}\ A \ldots$

(2) Não há forma de representar.

Código de Tunstall

Código de Tunstall

A , B , C

0,5 0.3 0.2 → Probs

A	0,5
B	0,3
C	0,2

A 0,5 → AA, AB, AC

AA	0,25
AB	0,15
AC	0,1
B	0,3
C	0,2

B 0,3 → BA, BB, BC

⋮

//

LZ 77 $\Sigma = \{a, b\}$

"Dict"

| \| \| \| \| \| \| | a a b | abba ... |

... 3 2 1

(0, 0, "a")

não há match

comprimento da match

primeiro caracter que faz match

| \| \| \| \| \| a | a b a | bba ... |

... 3 2 1

(1, 1, "b")

há match

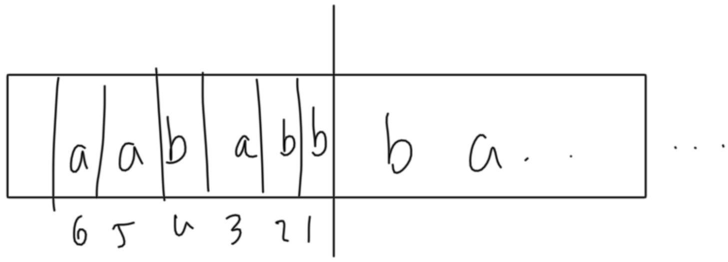
há 1 b

→ b

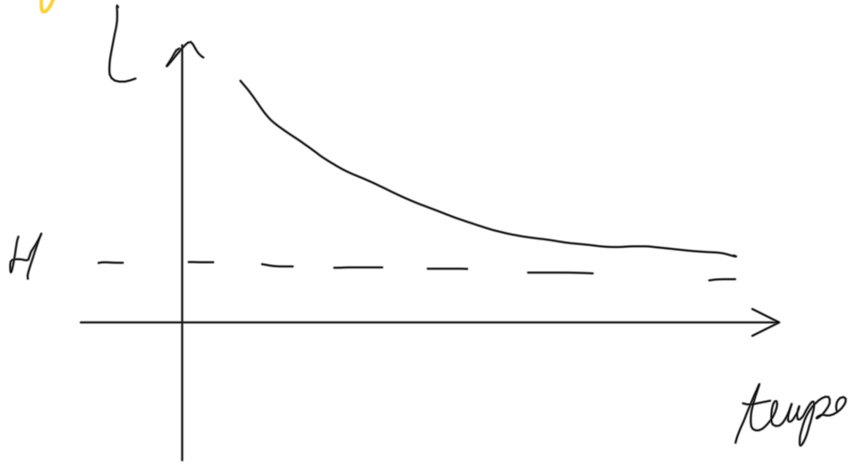
| \| \| \| \| a a b | a b b | a ... |

(2, 2, "b")

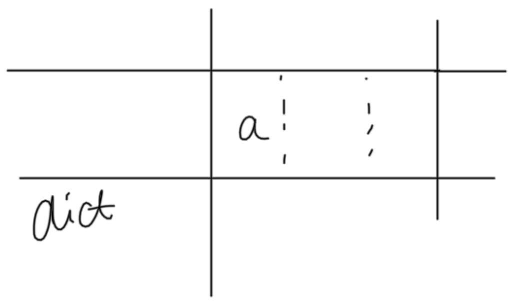
tamanho da match.



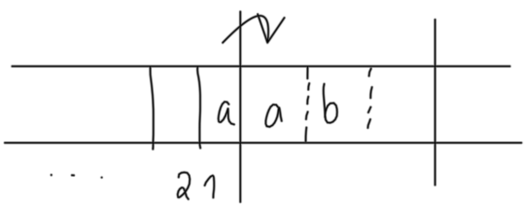
Algoritmos universais = LZ77 e LZ78



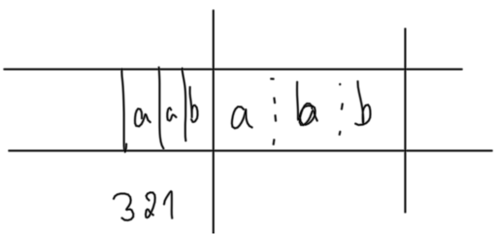
Descodificação



(0, 0, "a")



(1, 1, "b")



(2, 2, 'b')

LZ 78

a | a b | a b b | b | a a | a b b a ...

1 - a
2 - ab
3 - abb
4 - b
5 - aa
6 - abba
⋮

(0, 'a'), (1, 'b'), (2, 'b'),
(0, 'b'), (1, 'a'), (3, 'a').

Descodificação

a ab abb abb

1 - a
2 - ab
3 - abb