

# PROCESSAMENTO DE LINGUAGEM NATURAL

## Similaridade Textual



# TÓPICOS

1. **Introdução**
2. **Métricas baseadas em Caracteres**
3. **Métricas baseadas em Termos**

# SIMILARIDADE TEXTUAL

- Verificar o quão próximos são dois fragmentos de texto a partir do **significado** e **estrutura**

Frase 1: '**Os**' '*gatos*' '**comem**' '**os**' '*ratos*'

Frase 2: '**Os**' '*gatos*' '**comem**' '**os**' '*insetos*'

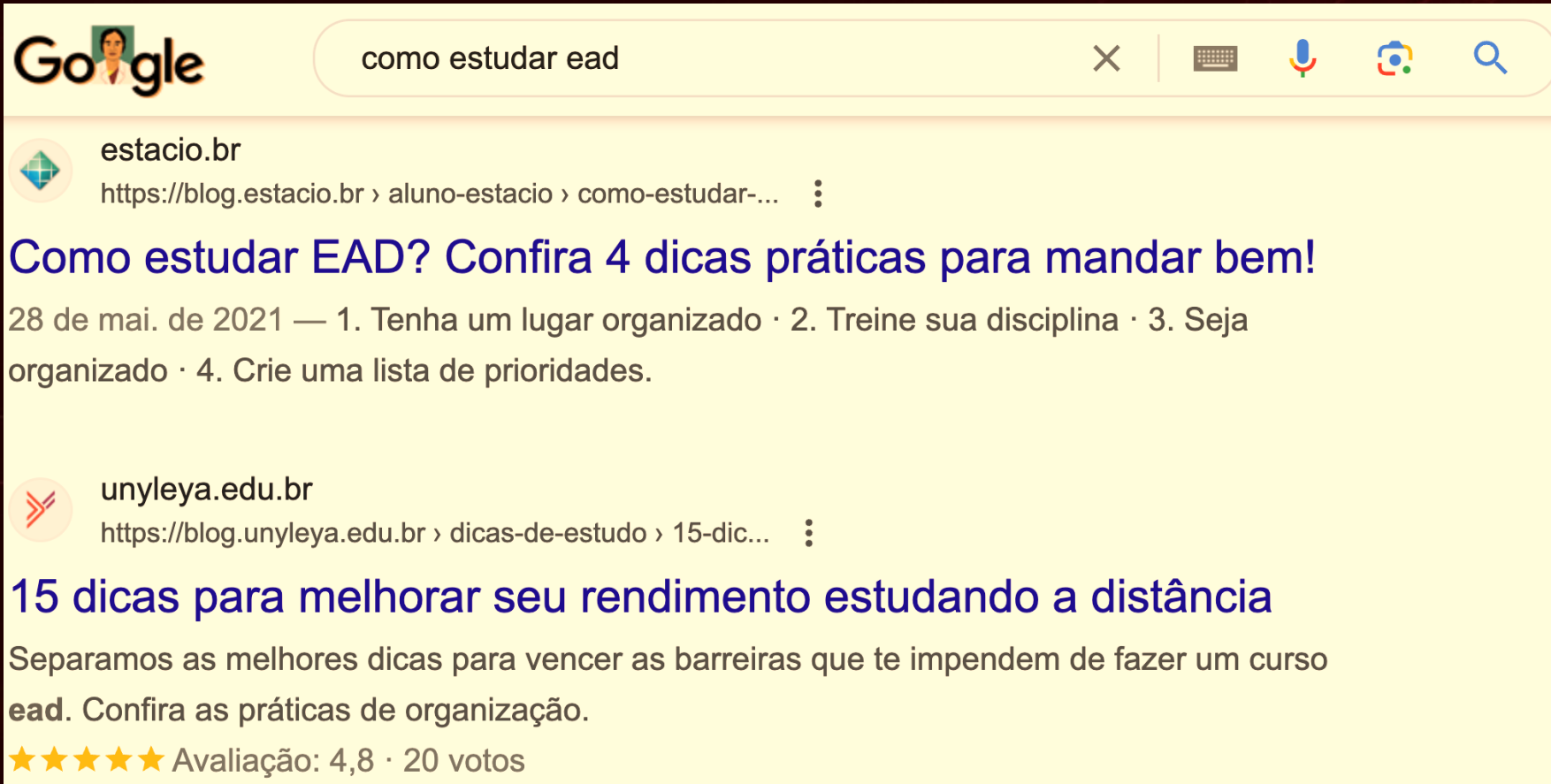
1

2

3

- **Similaridade Semântica**
  - Carro / Automóvel
- **Similaridade Léxica**
  - Carro / Barro

# SIMILARIDADE TEXTUAL



A screenshot of a Google search interface. The search bar at the top contains the text 'como estudar ead'. Below the search bar, two search results are displayed. The first result is from 'estacio.br' with the URL 'https://blog.estacio.br > aluno-estacio > como-estudar-...'. The title of the article is 'Como estudar EAD? Confira 4 dicas práticas para mandar bem!'. The snippet below the title reads: '28 de mai. de 2021 — 1. Tenha um lugar organizado · 2. Treine sua disciplina · 3. Seja organizado · 4. Crie uma lista de prioridades.' The second result is from 'unyleya.edu.br' with the URL 'https://blog.unyleya.edu.br > dicas-de-estudo > 15-dic...'. The title of the article is '15 dicas para melhorar seu rendimento estudando a distância'. The snippet below the title reads: 'Separamos as melhores dicas para vencer as barreiras que te impendem de fazer um curso ead. Confira as práticas de organização.' At the bottom of the second result, there is a star rating of five stars and the text 'Avaliação: 4,8 · 20 votos'.

Google

como estudar ead

estacio.br  
https://blog.estacio.br > aluno-estacio > como-estudar-...

**Como estudar EAD? Confira 4 dicas práticas para mandar bem!**

28 de mai. de 2021 — 1. Tenha um lugar organizado · 2. Treine sua disciplina · 3. Seja organizado · 4. Crie uma lista de prioridades.

unyleya.edu.br  
https://blog.unyleya.edu.br > dicas-de-estudo > 15-dic...

**15 dicas para melhorar seu rendimento estudando a distância**

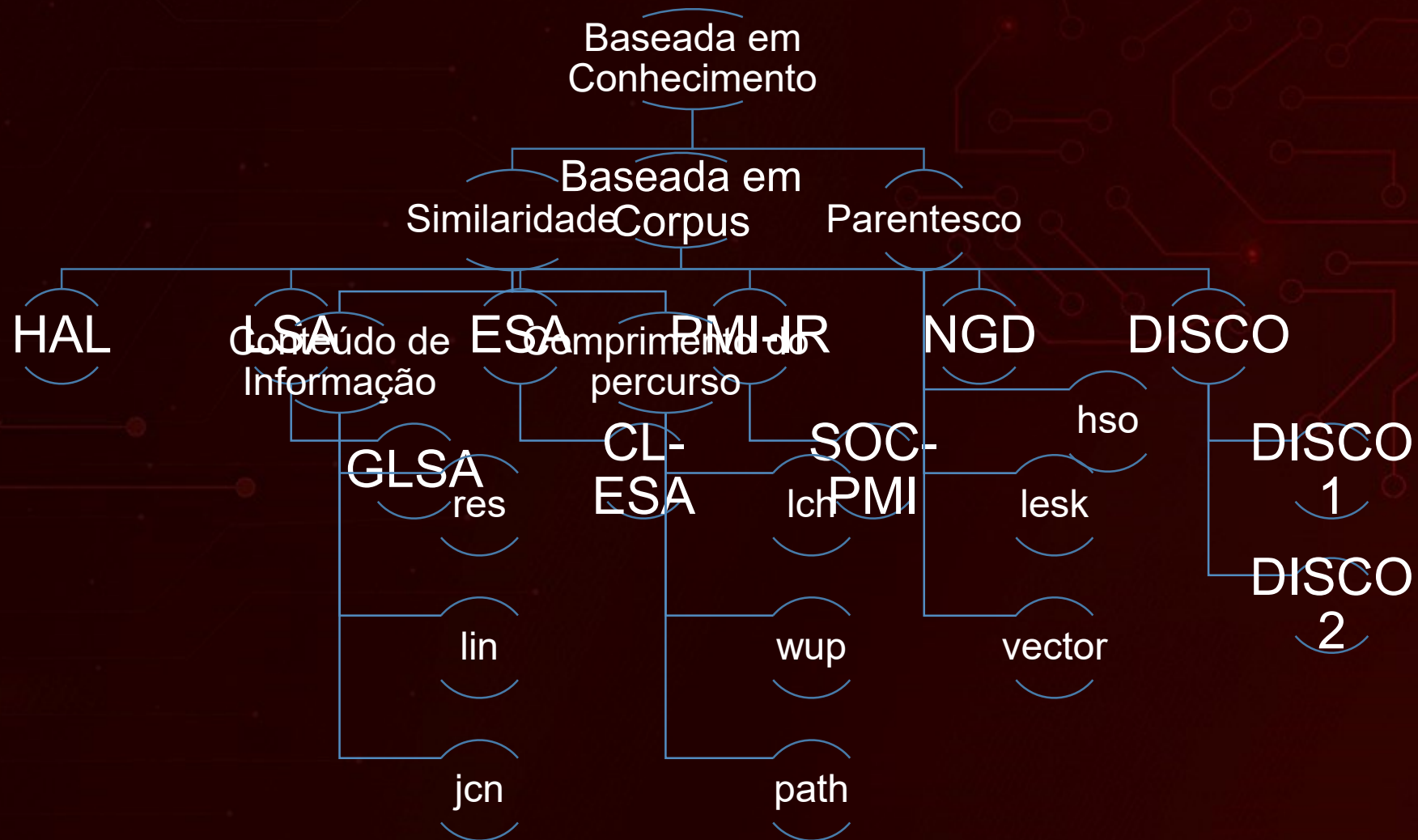
Separamos as melhores dicas para vencer as barreiras que te impendem de fazer um curso ead. Confira as práticas de organização.

★★★★★ Avaliação: 4,8 · 20 votos



# SIMILARIDADE TEXTUAL

([Gomma & Fahmy, 2013](#))



# SIMILARIDADE TEXTUAL

([Gomma & Fahmy, 2013](#))

- Baseada em *Strings*
- Baseada em termos
  - *Block Distance, Cosine Distance, Dice's Coefficient, Euclidean Distance, Jaccard Similarity, Matching Coefficient, Overlap Coefficient*
- Baseada em caracteres
  - LCS, Damerau-Levenshtein, Jaro, Jaro-Winkler, Needleman-Wunsch, Smith-Waterman, N-gram

# MÉTRICAS BASEADAS EM *STRINGS*

- Distância/Similaridade baseada em *string* é a mais antiga, simples e utilizada, operando sobre sequências de *strings* e composição de caractere.
- **Baseadas em Termos**
  - Distância depende do conjunto de palavras (*tokens*) contidas em *s* e *t*.
- **Baseadas em caracteres**
  - Menor número de operações necessárias para transformar *s* em *t*.

# TÓPICOS

1. Introdução
2. **Métricas baseadas em Termos**
3. Métricas baseadas em Caracteres



# MÉTRICAS BASEADAS EM TERMOS

- Distância entre  $s$  e  $t$  é baseada no conjunto de palavras que aparecem em  $s$  e  $t$ .
- A ordem das palavras não é relevante
  - Ex.: “William Cohen” = “Cohen, William” e “Joyce James” = “James Joyce”
- Normalmente, as palavras são ponderadas e as mais comuns contam menos
  - Ex.: “Silva” conta menos que “Basgalupp”

# JACCARD

$S$		William	Cohen	CM	Univ		Pgh
$T$	Dr.	William	Cohen	CM		University	
$ S \cup T $	Dr.	William	Cohen	CM	Univ	University	Pgh
$ S \cap T $		William	Cohen	CM			

$$Jaccard\ Score = \frac{|S \cup T|}{|S \cap T|} = \frac{3}{7}$$

# MÉTRICAS BASEADAS EM TERMOS

- **Vantagens:**

- Explora informações de frequência
- Eficiente: encontrar  $\{t : \text{sim}(t,s) > k\}$  é sublinear!
- Ordem das palavras é ignorada (William Cohen vs Cohen, William)

- **Desvantagens:**

- Sensível a erros de digitação (William Cohon)
- Sensível a abreviações (Univ. vs University)
- Ordem das palavras é ignorada (James Joyce vs Joyce James)

# TÓPICOS

1. Introdução
2. Métricas baseadas em Termos
3. Métricas baseadas em Caracteres



# ***MINIMUM EDIT DISTANCE***

- **Número mínimo de operações de edição (inserção, exclusão e substituição) necessárias para transformar uma *string* em outra**
- **Útil para tarefas como:**
  - **Correção ortográfica**
  - **Resolução de correferência**
  - **Identificação de variantes linguísticas ou ortográficas**
  - **Identificação de cognatos**

# LEVENSHTEIN

- Custos das operações = 1

1 – “ABADAC”  
2 – “CADA”

Passo 1 – Exclusão da primeira letra ‘A’, gerando “BADAC”;

Passo 2 – Substituição de ‘C’ por ‘B’, gerando “CADAB”;

Passo 3 – Exclusão da letra ‘B’, gerando “CADA”.

**3 operações**

# LEVENSHTEIN - ALTERNATIVA

- Levenshtein também propôs uma versão alternativa em que:
  - Substituição tem custo = 2

1 – “ABADAC”  
2 – “CADA”

Passo 1 – Exclusão da primeira letra ‘A’, gerando “BADAC”; **+1**

Passo 2 – Substituição de ‘C’ por ‘B’, gerando “CADAB”; **+2**

Passo 3 – Exclusão da letra ‘B’, gerando “CADA”. **+1**

**4 operações**

## EXEMPLO: LEVENSHTEIN

- **distancia("William Cohen", "Willliam Cohon")**

[illegible]



## EXEMPLO: LEVENSHTEIN

- **distancia("William Cohen", "Willliam Cohon")**

[illegible]

# EXEMPLO: LEVENSHTTEIN (1)

- $D(i, j)$  = melhor alinhamento de  $s_1..s_i$  para  $t_1..t_j$

$$= \min \left\{ \begin{array}{ll} D(i-1, j-1), \text{ se } s_i = t_j & // \text{ copiar} \\ D(i-1, j-1) + 1, \text{ se } s_i \neq t_j & // \text{ substituir} \\ D(i-1, j) + 1 & // \text{ inserir} \\ D(i, j-1) + 1 & // \text{ remover} \end{array} \right.$$

## EXEMPLO: LEVENSHTTEIN (2)

- $D(i, j)$  = melhor alinhamento de  $s_1..s_i$  para  $t_1..t_j$

$$= \min \begin{cases} D(i-1, j-1) + d(s_i, t_j) & // \text{substituir/copiar} \\ D(i-1, j) + 1 & // \text{inserir} \\ D(i, j-1) + 1 & // \text{remover} \end{cases}$$

- (simplificação:  $D(c, d) = 0$  se  $c = d$ , 1 caso contrário)
- além disso,  $D(i, 0) = i$  (para  $i$  inserções) e  $D(0, j) = j$

## EXEMPLO: LEVENSHTTEIN (3)

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + d(s_i, t_j) & // \text{ substituir/copiar} \\ D(i-1, j) + 1 & // \text{ inserir} \\ D(i, j-1) + 1 & // \text{ remover} \end{cases}$$

	C	O	H	E	N
M	1	2	3	4	5
C	1	2	3	4	5
C	2	2	3	4	5
O	3	2	3	4	5
H	4	3	2	3	4
N	5	4	3	3	3

$= D(s, t)$



## EXEMPLO: LEVENSHTTEIN (4)

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + d(s_i, t_j) & // \text{substituir/copiar} \\ D(i-1, j) + 1 & // \text{inserir} \\ D(i, j-1) + 1 & // \text{remover} \end{cases}$$

- Caminho indica de onde veio o valor mínimo. Pode ser usado para encontrar as operações e o melhor alinhamento (pode ser mais de um)

	C	O	H	E	N
M	1	2	3	4	5
C	1	2	3	4	5
C	2	2	3	4	5
O	3	2	3	4	5
H	4	3	2	3	4
N	5	4	3	3	3

=  $D(s, t)$

# NEEDLEMAN-WUNCH

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + d(s_i, t_j) & // \text{ substituir/copiar} \\ D(i-1, j) + G & // \text{ inserir} \\ D(i, j-1) + G & // \text{ remover} \end{cases}$$

$G$  = "gap cost"

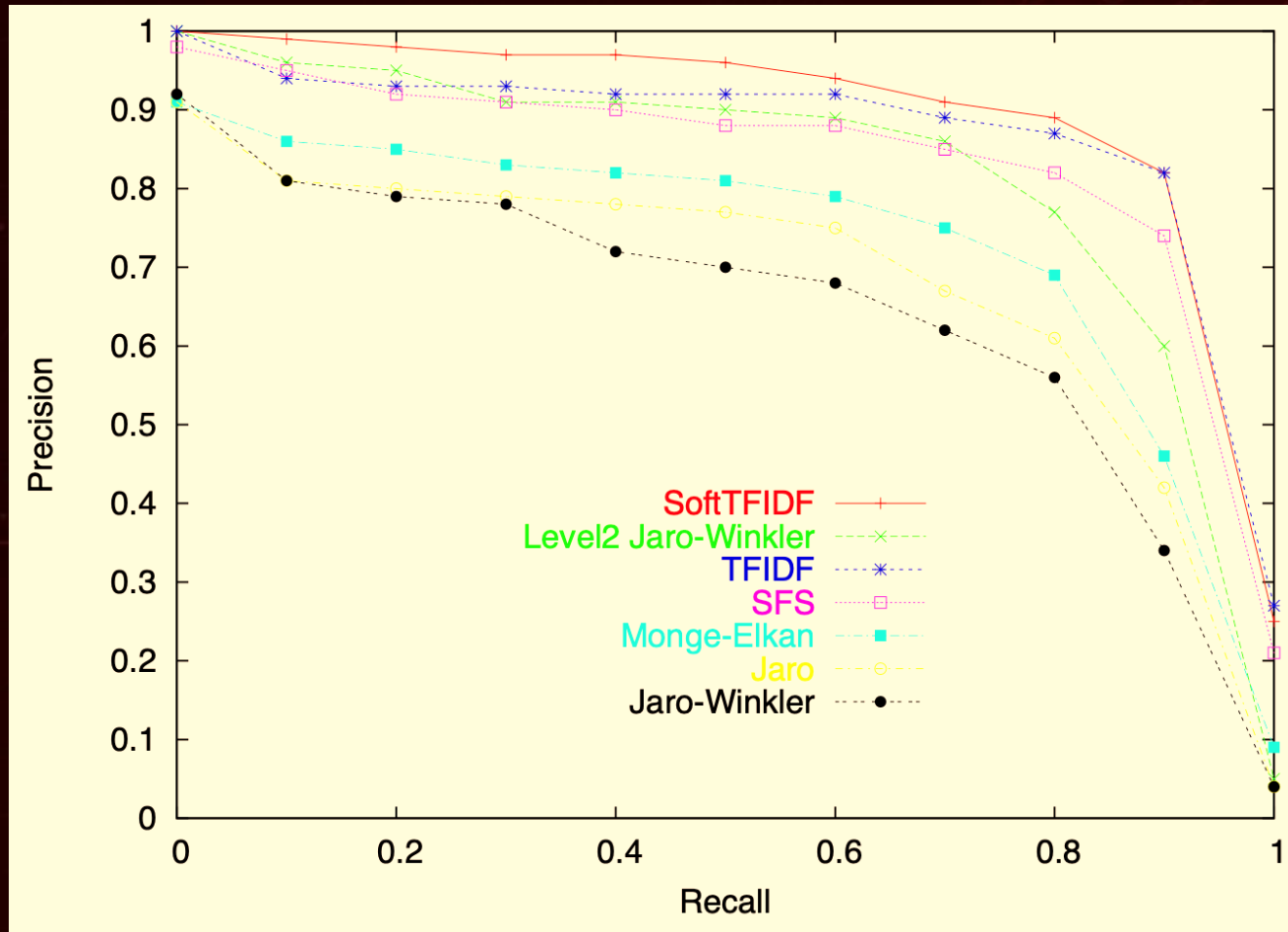
$d(c, d)$  é uma função de distância arbitrária em caracteres (ex.: relacionada à frequência de *typos*)

William Cohen



Wukkuan Cigeb

# RESULTADOS EXPERIMENTAIS



# O QUE VIMOS?

- **Introdução**
- **Métricas baseadas em Caracteres**
- **Métricas baseadas em Termos**



# PRÓXIMA VIDEOAULA

- Prática: *Corpus* e Similaridade Textual

# REFERÊNCIAS

- **Curso de Processamento de Linguagem Natural**
  - **Profa. Helena Caseli (UFSCar)**
- **Curso de Processamento de Linguagem Natural**
  - **Prof. Thiago Pardo (ICMC-USP)**
- **Curso de Linguística Computacional**
  - **Prof. Thiago Castro Ferreira (UFMG)**
- **Carnegie Mellon University**
  - **Prof. Willian W. Cohen**