MCZA017-13
Processamento de Linguagem Natural

Casamento aproximado entre strings

Semântica e similaridade de palavras - Parte I

Prof. Jesús P. Mena-Chalco
jesus.mena@ufabc.edu.br

2Q-2019

Casamento aproximado entre *strings*

String matching

No contexto de correção ortográfica

- “Graffe”

É mais próximo a?

- Graf
- Graft
- Grail
- Giraffe

String matching

No contexto de biologia computacional

- As sequências de aminoacidos

```
AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC
```

- Pode ser alinhado a:

```
-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC
```

Distância de Hamming

Utilizado para detectar erros nas transmissões binárias de comprimento fixo.

A distancia de Hamming é a quantidade de bits usado na mudança de uma transmissão para a recepção.

Distância de Hamming

```
def hamming(s1, s2):
    if len(s1) == len(s2):
        cont = 0
        for i in range(0, len(s2)):
            if s1[i]!=s2[i]:
                cont = cont + 1
        return cont
    else:
        return -1

if __name__ == "__main__":

    print( hamming("PLN", "PNL") )
    print( hamming("UFABC", "UFRJ")  )
    print( hamming("UFABC", "UFRRJ") )
```

Distância de Levenshtein

- É usada para **medir a quantidade de diferenças** entre duas strings.
- Usando-se de operaçoes como **inserção, exclusão e substituição**, esta distância métrica define o número mínimo de edições para transformar uma string em outra.
- Por exemplo, a distância de Levenshtein entre “**casa**” e “**pata**” é 2
  - Pois não há maneira de o fazer com menos de 2 edições.
  - A conversão de “casa” para “pata” é obtida substituindo-se “c” por “p”, logo substituindo-se “s” por “t”.

Distância de Levenshtein

$$lev_{a,b}(i,j) \begin{cases} max(i,j) & ,se \quad min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) \\ lev_{a,b}(i,j-1) \\ lev_{a,b}(i-1,j-1)_{a_i \neq b_i} \end{cases} & ,caso \quad contrario \end{cases}$$

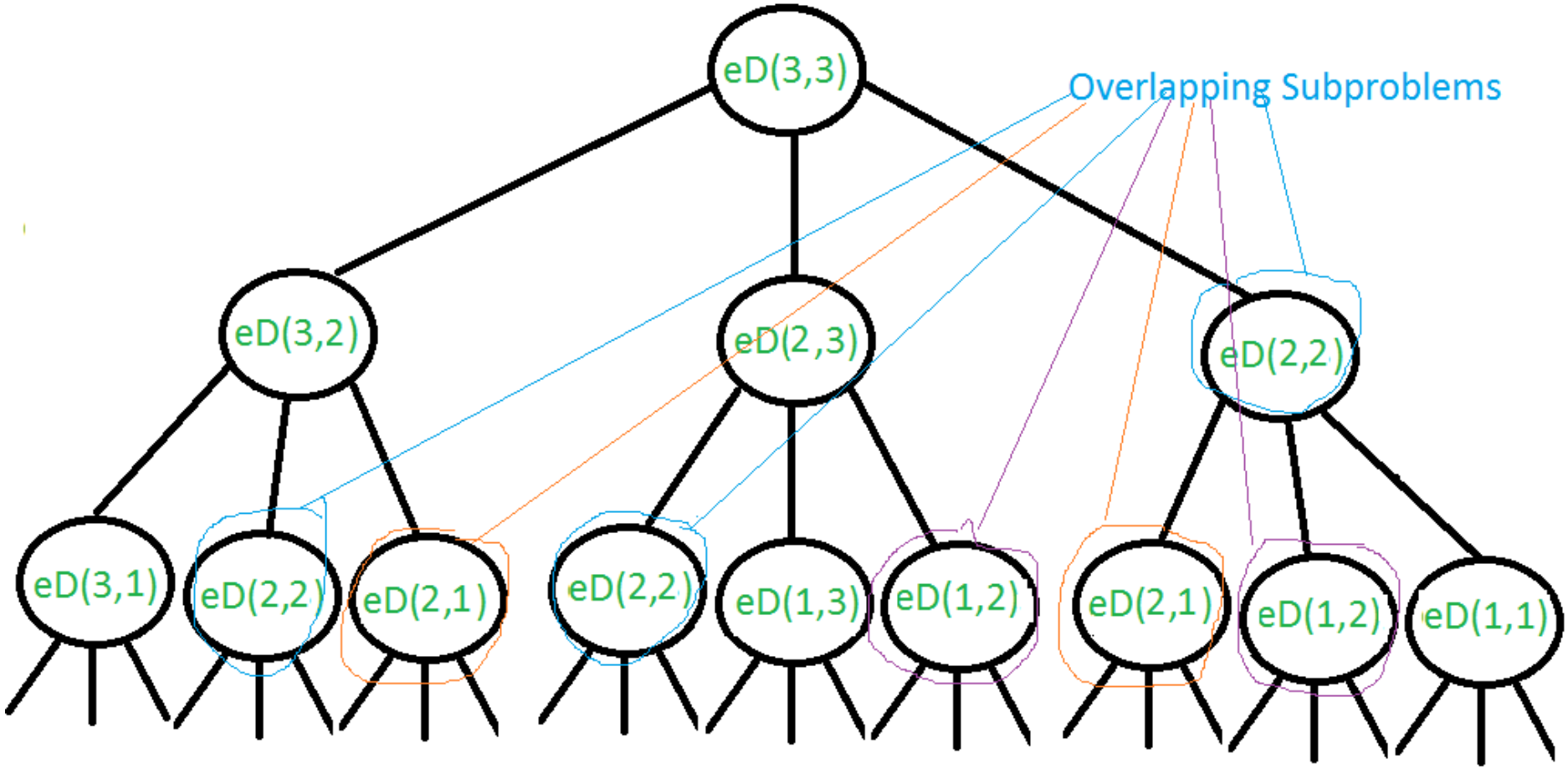
- Sejam **a** e **b** duas strings.
- Sejam **i** e **j** o comprimento de **a** e **b**, respectivamente.

- Se o comprimento de uma string for zero, então a distância será igual ao comprimento da outra string.

Distância de Levenshtein

```
def Levenshtein(s1,s2):
    if len(s1) > len(s2):
        #s1,s2 = s2,s1
        temp = s1
        s1 = s2
        s2 = temp

    #caso base
    if len(s1)==0 or len(s2)==0:
        if len(s1)>len(s2):
            return len(s1)
        else:
            return len(s2)

    #def. de custo
    if s1[-1] == s2[-1]:
        custo = 0
    else:
        custo = 1

    # determina o melhor caminho na comparacao
    primeiroTermo = Levenshtein(s1[:-1],s2) + 1
    segundoTermo  = Levenshtein(s1,s2[:-1]) + 1
    terceiroTermo = Levenshtein(s1[:-1],s2[:-1]) + custo

    return min( primeiroTermo, segundoTermo, terceiroTermo)
```

Distância de Levenshtein

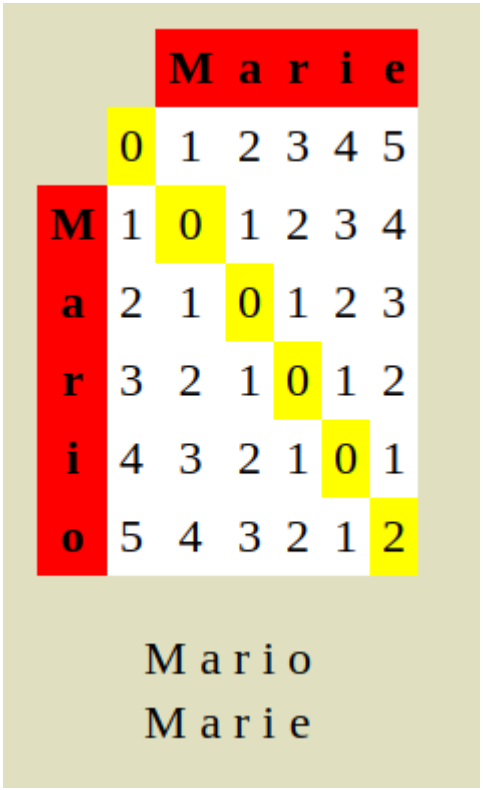


Worst case recursion tree when m = 3, n = 3.
Worst case example str1="abc" str2="xyz"

https://www.geeksforgeeks.org/edit-distance-dp-using-memoization/

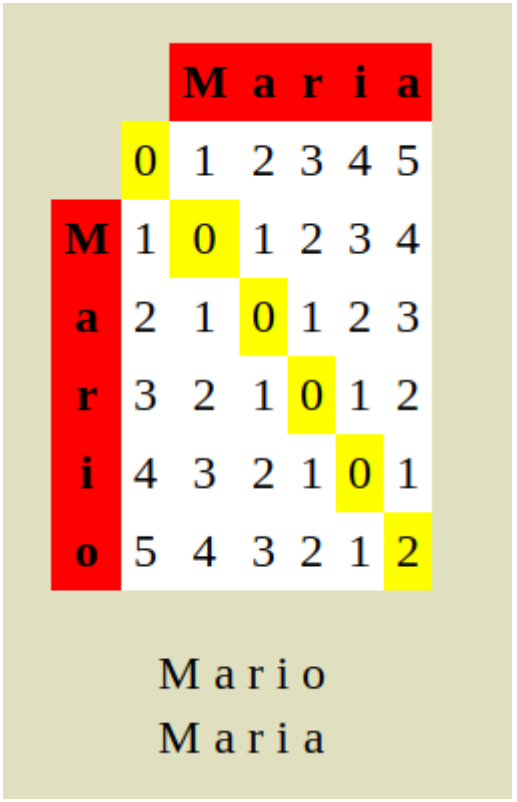
Distância de Levenshtein

Pesos
+1 = Inserção
+1 = Eliminação
+2 = Substituição



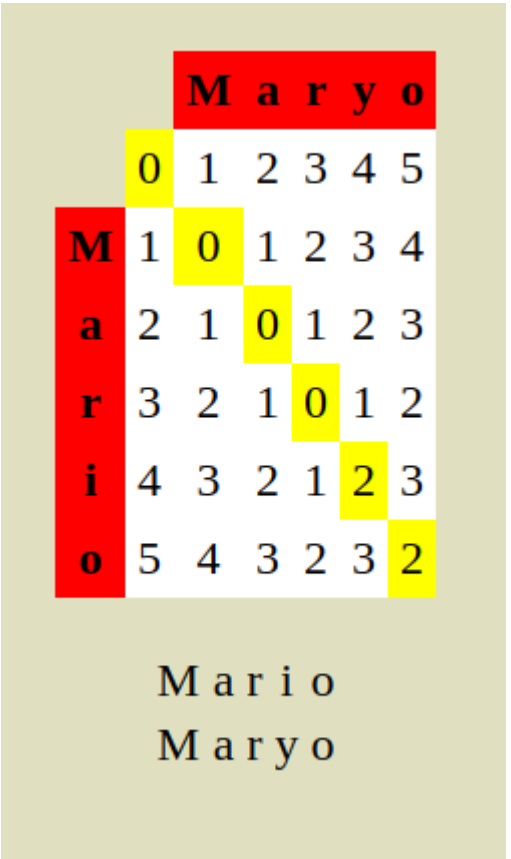
		M	a	r	i	e
	0	1	2	3	4	5
M	1	0	1	2	3	4
a	2	1	0	1	2	3
r	3	2	1	0	1	2
i	4	3	2	1	0	1
o	5	4	3	2	1	2

Mario
Marie



		M	a	r	i	a
	0	1	2	3	4	5
M	1	0	1	2	3	4
a	2	1	0	1	2	3
r	3	2	1	0	1	2
i	4	3	2	1	0	1
o	5	4	3	2	1	2

Mario
Maria



		M	a	r	y	o
	0	1	2	3	4	5
M	1	0	1	2	3	4
a	2	1	0	1	2	3
r	3	2	1	0	1	2
i	4	3	2	1	2	3
o	5	4	3	2	3	2

Mario
Maryo

http://www.let.rug.nl/~kleiweg/lev/

Distância de Levenshtein

Pesos
+1 = Inserção
+1 = Eliminação
+2 = Substituição

		D	U	D	U
	0	1	2	3	4
E	1	2	3	4	5
D	2	1	2	3	4
U	3	2	1	2	3

E D U
D U D U

E D U
D U D U

E D U
D U D U

E D U
D U D U

		E	I	D	U
	0	1	2	3	4
E	1	0	1	2	3
D	2	1	2	1	2
U	3	2	3	2	1

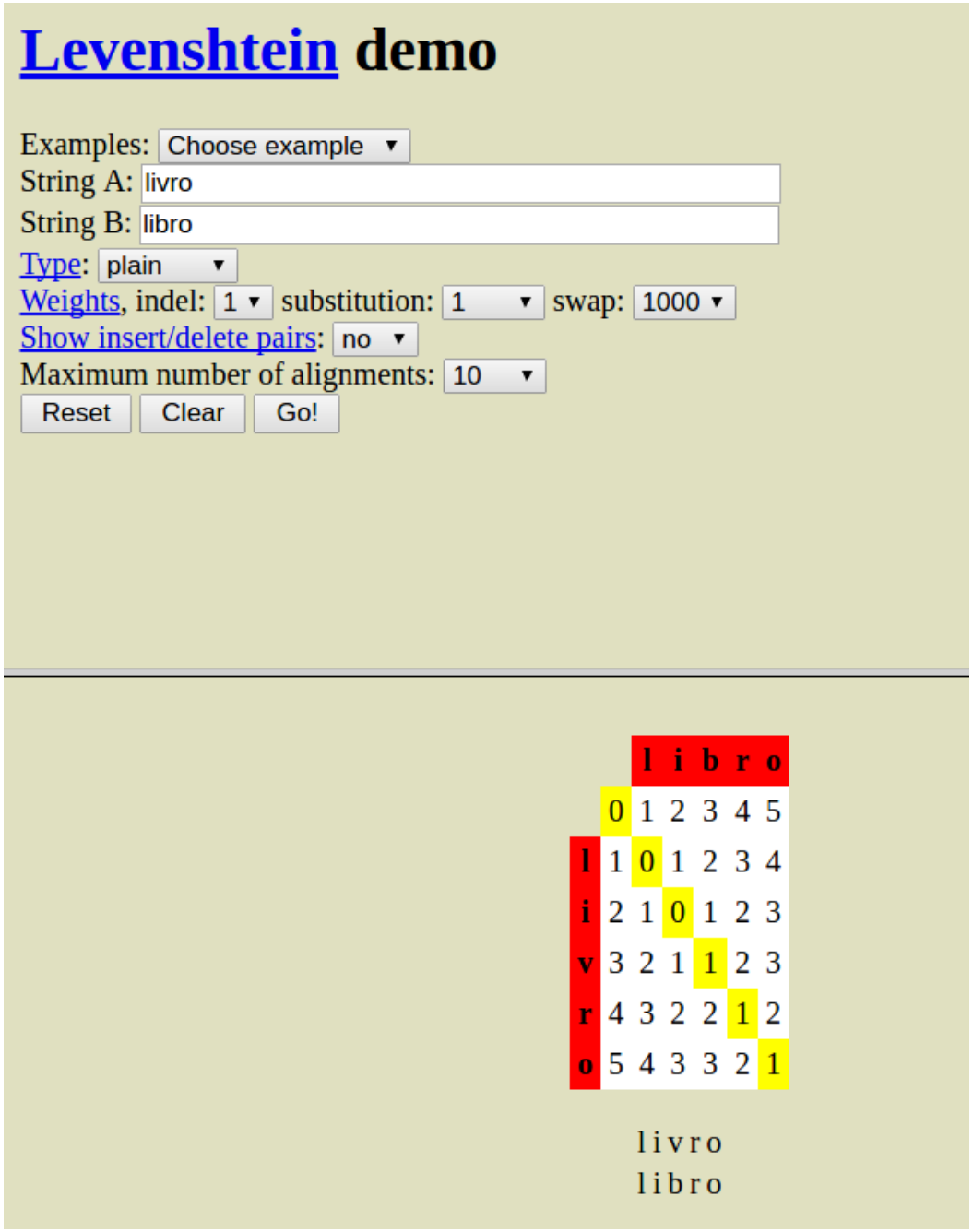
E D U
E I D U

		E	D	U	A	R	D	O
	0	1	2	3	4	5	6	7
E	1	0	1	2	3	4	5	6
D	2	1	0	1	2	3	4	5
U	3	2	1	0	1	2	3	4

E D U
E D U A R D O

http://www.let.rug.nl/~kleiweg/lev/

http://www.let.rug.nl/kleiweg/lev/



String A: Luis Paulo Roberto Cozta

String B: Luiz Paulo Roberto Cotia

		L	u	i	z		P	a	u	l	o		R	o	b	e	r	t	o		C	o	t	i	a
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
L	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
u	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
i	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
s	4	3	2	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	5	4	3	2	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P	6	5	4	3	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	7	6	5	4	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
u	8	7	6	5	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
l	9	8	7	6	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
o	10	9	8	7	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	11	10	9	8	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14
R	12	11	10	9	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13
o	13	12	11	10	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12
b	14	13	12	11	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11
e	15	14	13	12	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	10
r	16	15	14	13	13	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9
t	17	16	15	14	14	13	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8
o	18	17	16	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7
	19	18	17	16	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5	6
C	20	19	18	17	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5
o	21	20	19	18	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	2	3	4
z	22	21	20	19	18	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	2	3	4
t	23	22	21	20	19	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	3	4
a	24	23	22	21	20	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3	3

ELSEVIER

Information and Control

Volume 64, Issues 1–3, January–March 1985, Pages 100-118

Algorithms for approximate string matching *

Esko Ukkonen

Department of Computer Science, University of Helsinki, Tukholmankatu 2, SF-00250 Helsinki, Finland

Available online 5 May 2005.

Show less

https://doi.org/10.1016/S0019-9958(85)80046-2

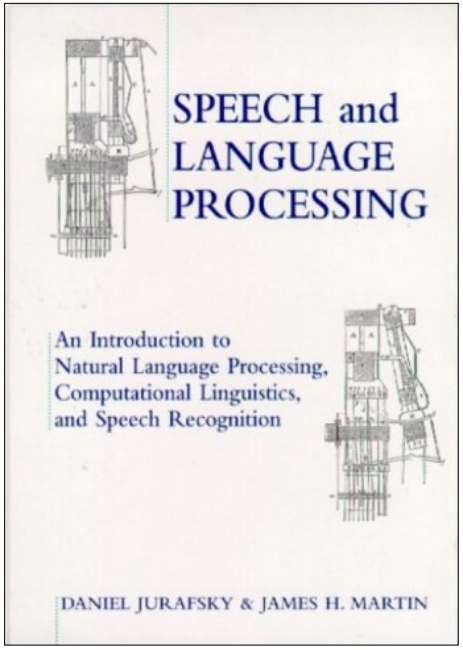
The edit distance between strings $a_1 \ldots a_m$ and $b_1 \ldots b_n$ is the minimum cost s of a sequence of editing steps (insertions, deletions, changes) that convert one string into the other. A well-known tabulating method computes s as well as the corresponding editing sequence in time and in space $O(mn)$ (in space $O(\min(m, n))$ if the editing sequence is not required). Starting from this method, we develop an improved algorithm that works in time and in space $O(s \cdot \min(m, n))$. Another improvement with time $O(s \cdot \min(m, n))$ and space $O(s \cdot \min(s, m, n))$ is given for the special case where all editing steps have the same cost independently of the characters involved. If the editing sequence that gives cost s is not required, our algorithms can be implemented in space $O(\min(s, m, n))$. Since $s = O(\max(m, n))$, the new methods are always asymptotically as good as the original tabulating method. As a by-product, algorithms are obtained that, given a threshold value t, test in time $O(t \cdot \min(m, n))$ and in space $O(\min(t, m, n))$ whether $s \leqslant t$. Finally, different generalized edit distances are analyzed and conditions are given under which our algorithms can be used in conjunction with extended edit operation sets, including, for example, transposition of adjacent characters.

```
def is_edit_le(s1, s2, le):
    '''Return the edit distance if <= le; return le+1 otherwise.
    m = len(s1)
    n = len(s2)
    p = -1
    r = p - min(m, n)
    f_kp = _Fkp()
    while (n-m, p) not in f_kp or f_kp[n-m, p] != m:
        p += 1
        if p > le:
            # The number of edit operations is larger than the limit
            # le
            return p
        r += 1
        if r <= 0:
            for k in range(-p, p+1):
                _fill_f_kp(k, p, f_kp, s1, s2)
        else:
            for k in range(max(-m, -p), -r+1):
                _fill_f_kp(k, p, f_kp, s1, s2)
            for k in range (r, min(n, p) + 1):
                _fill_f_kp(k, p, f_kp, s1, s2)
    return p
```

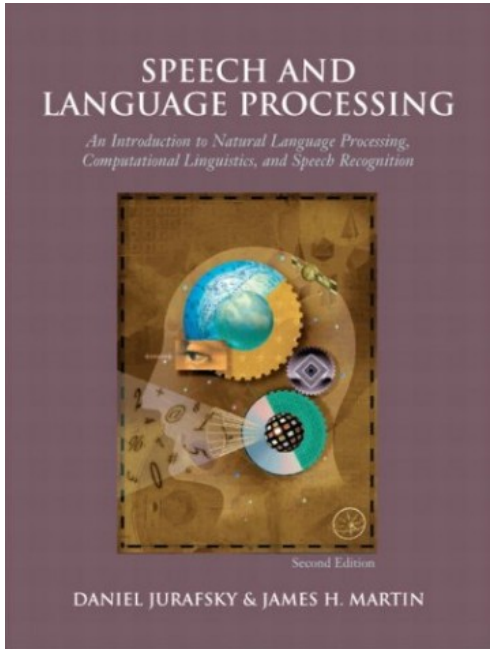
Bibliografia

Daniel Jurafsky & James H. Martin.

Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition**.** Pearson/Prentice Hall.



2000



2009

2019?

Stanford University

University of Colorado, Boulder

Bibliografia – Capítulo 6

Speech and Language Processing (3rd ed. draft)

Dan Jurafsky and James H. Martin

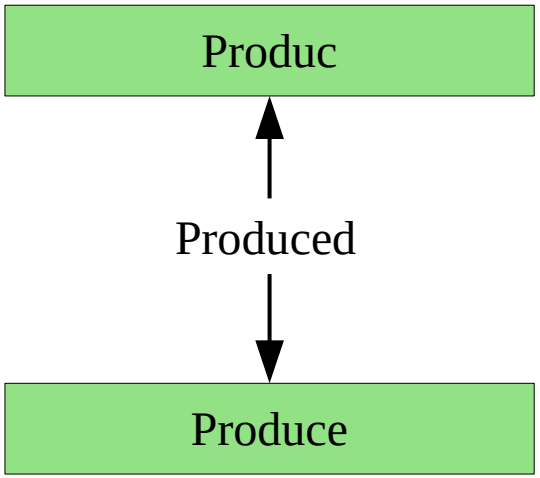
	Chapter	Slides	Relation to 2nd ed.
1:	Introduction		[Ch. 1 in 2nd ed.]
2:	Regular Expressions, Text Normalization, and Edit Distance	Text [pptx] [pdf] Edit Distance [pptx] [pdf]	[Ch. 2 and parts of Ch. 3 in 2nd ed.]
3:	Language Modeling with N-Grams	LM [pptx] [pdf]	[Ch. 4 in 2nd ed.]
4:	Naive Bayes Classification and Sentiment	NB [pptx] [pdf] Sentiment [pptx] [pdf]	[new in this edition]
5:	Logistic Regression		
6:	Vector Semantics	Vector1 [pptx] [pdf] Vector2 [pptx] [pdf]	

https://web.stanford.edu/~jurafsky/slp3/

Cinco definições importantes sobre ‘significado’ de palavras

Da aula 04: Stemming x Lemmatization

- *Stemming* (a ação de reduzir em **stems**)
  - **Stem: Parte de uma palavra**
  - Stemmer: O artefato (programa)



- *Lemmatization* (a ação de reduzir em **Lemmas**)
  - **Lemma: Forma básica da palavra**
  - Lemmatizer: O artefato (programa)

Lemma e Wordform

Lemma: é a forma básica da palavra (sem inflexão).

Wordform: é uma palavra com inflexão.

wordform	Lemma
Banks	Bank
Sung	Sing
Durmiu	dormir
Bancos	Banco

Diferentes significados?

Um determinado **lemma** pode ter significados diferentes.

Diferentes significados?

Um determinado **lemma** pode ter significados diferentes.

Exemplo:

- “… um **banco** pode manter investimentos dos correntistas ...” Inst. Financeira
- “… os métodos implementados em um **banco** de dados...” Artefato
- “… trocaram de cor o **banco** de madeira ...” Assento

1) Homônimos

São palavras que **compartilham a mesma forma** mas com significados diferentes (origens diferentes).

- **Banco**: Instituição financeira.
- **Banco**: Artefato para armazenamento de dados.
- **Banco**: Assento.

Homônimos podem ser:

- Homógrafos, i.e., mesma forma de **escrita** (banco/banco)
- Homófonos, i.e., mesma forma de **fala** (Concerto/conserto)

Homônimos criam problemas em PLN

- **Em recuperação de informação**

  “banco quebrado” (a instituição ou o assento?)

- **Em tradução de textos**

  bat: “morcego”

  bat: “bastão”

- **Em aplicações text-to-speech** (a pronuncia é diferente)

  bass (instumento musical)

  bass (peixe)

2) Polissemia (muitos significados)

É a propriedade de uma palavra tem de apresentar vários significados.

Uma palavra polissêmica tem significados relacionados. (origens similares):

- **Letra**: Elemento básico de um alfabeto.
- **Letra**: Texto de uma canção.
- **Letra**: Caligrafia de uma determinado indivíduo.

- **Vela: ...** de um barco
- **Vela: ...** para iluminar
- **Vela: ...** de vigilante

Relações sistemáticas (metonímia)

Muitos tipos de polissemia **são sistemáticos**:

- Rádio
- Universidade
- Escola
- Hospital

Prédio ↔ Organização

Relações sistemáticas (metonímia)

Muitos tipos de polissemia são sistemáticos:

- Rádio
- Universidade
- Escola
- Hospital

Prédio ↔ Organização

Outros tipos de relações sistemáticas:

- Eu amo J. K. Rowling
- Eu amo (as obras de) J. K. Rowling

Autor ↔ Trabalhos de autor

- Maracujá tem lindas flores
- Ontem experimentei maracujá

Árvore ↔ Fruto

Como determinar se uma palavra tem mais de um significado?

Como determinar se uma palavra tem mais de um significado?

Usando o teste “Zeugma” (figura de linguagem ou estilo)

- ... construirá uma **universidade** de mármore …
- … pedirá à **universidade** de João ...

Como determinar se uma palavra tem mais de um significado?

Usando o teste “Zeugma” (figura de linguagem ou estilo)

- ... construirá uma **universidade** de mármore …
- … pedirá à **universidade** de João ...

Teste:

Se a construção não faz sentido (coerente), provavelmente a palavra seja polissêmica:

“construirá uma universidade de mármore e de João?”

3) Sinônimos

Palavras que tem o **mesmo significado** em alguns ou todos os contextos.

- Caderno — Caderneta
- Carro — Automóvel
- Sofá — Divá
- Agua — H_2O
- Computador — PC

Duas palavras são sinônimas se:

- Ambas podem ser substituídas **em todas** as situações.
- Ambas têm o mesmo significado proposicional.

4) Antônimos

Palavras que tem **significado oposto** em relação a uma característica.

- escuro — claro
- quente — frio
- curto — longo
- para cima — para baixo
- rápido — lento

5) Hiponímia e Hiperonímia

sub super

Indicam relação hierarquica de significados entre palavras.

Uma palavra A é **hiponímia** de B, se o significado de A é mais específico que B:

- Carro é uma hiponímia de Automóvel
- Sandália é uma hiponímia de Calçado

Se modo inverso:

- Automóvel é uma hiperonímia de Carro
- Calçado é uma hiperonímia de Sandália

Wordnet:
Um repositório (tesauro) muito útil em PLN

Wordnet – wordnet.princeton.edu

A Wordnet é uma base de dados (1985) usada na área de linguística computacional, em inglês.

Wordnet está organizado em base de relações (hierárquicas).

Usado para **desambiguar o significado** das palavras.

Category	Unique Strings
Noun	117,798
Verb	11,529
Adjective	22,479
Adverb	4,481

← *Versão 3.0, contem mais substantivos*

É um tesauro, isto é, um dicionário de 'ideias comuns'

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: university Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

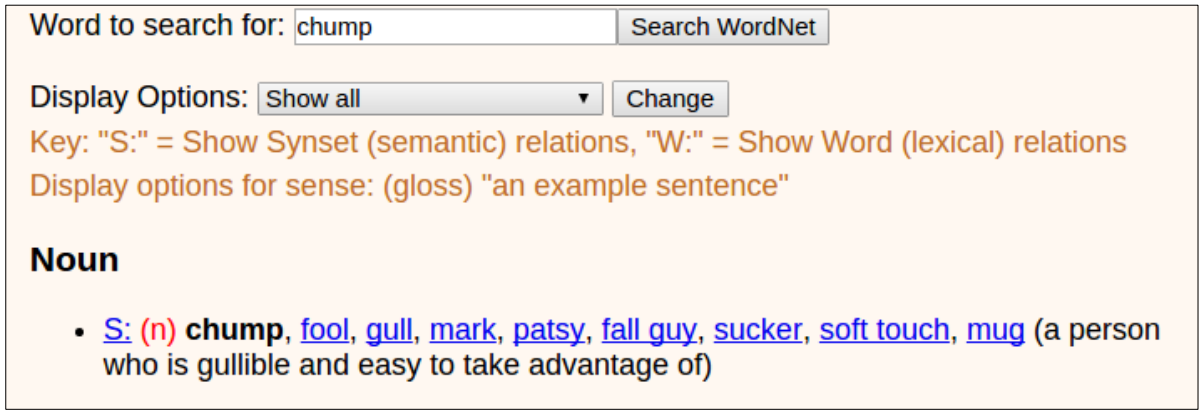
Noun

- S: (n) **university** (the body of faculty and students at a university)
- S: (n) **university** (establishment where a seat of higher learning is housed, including administrative and living quarters as well as facilities for research and teaching)
  - ***direct hyponym*** / *full hyponym*
    - S: (n) city university (an urban university in a large city)
    - S: (n) Oxbridge (general term for an ancient and prestigious and privileged university (especially Oxford University or Cambridge University))
    - S: (n) redbrick university ((British informal) a provincial British university of relatively recent founding; distinguished from Oxford University and Cambridge University)
    - S: (n) Brown University, Brown (a university in Rhode Island)
    - S: (n) Cambridge University, Cambridge (a university in England)
    - S: (n) Carnegie Mellon University (an engineering university in Pittsburgh)
    - S: (n) Columbia University, Columbia (a university in New York City)
    - S: (n) Cooper Union, Cooper Union for the Advancement of Science and Art (university founded in 1859 by Peter Cooper to offer free courses in the arts and sciences)
    - S: (n) Cornell University (a university in Ithaca, New York)
    - S: (n) Duke University (a university in Durham, North Carolina)
    - S: (n) Harvard University, Harvard (a university in Massachusetts)
    - S: (n) Johns Hopkins (a university in Baltimore)
    - S: (n) Massachusetts Institute of Technology, MIT (an engineering university in Cambridge)


WordNet Search - 3.1
- WordNet home page - Glossary - Help
Word to search for: bass
Search WordNet
Display Options: (Select option to change)
Change
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"
Noun
S: (n) bass (the lowest part of the musical range)
S: (n) bass, bass part (the lowest part in polyphonic music)
S: (n) bass, basso (an adult male singer with the lowest voice)
S: (n) sea bass, bass (the lean flesh of a saltwater fish of the family Serranidae)
S: (n) freshwater bass, bass (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
S: (n) bass, bass voice, basso (the lowest adult male singing voice)
S: (n) bass (the member with the lowest range of a family of musical instruments)
S: (n) bass (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)
Adjective
S: (adj) bass, deep (having or denoting a low vocal or instrumental range) "a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"

Synset = Synonym set

É um conjunto de sinônimos (próximos) a uma palavra



Synset = Synonym set

Hierarquia de hiperonomios

Noun

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, bass part (the lowest part in polyphonic music)
- S: (n) **bass**, basso (an adult male singer with the lowest voice)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) singer, vocalist, vocalizer, vocaliser (a person who sings)
      - S: (n) musician, instrumentalist, player (someone who plays a musical instrument (as a profession))
        - S: (n) performer, performing artist (an entertainer who performs a dramatic or musical work for an audience)
          - S: (n) entertainer (a person who tries to please or amuse)
            - S: (n) person, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
              - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                - S: (n) living thing, animate thing (a living (or once living) entity)
                  - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                    - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                      - S: (n) physical entity (an entity that has physical existence)
                        - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
              - S: (n) causal agent, cause, causal agency (any entity that produces an effect or is responsible for events or results)
                - S: (n) physical entity (an entity that has physical existence)

Wordnet – diferentes iniciativas

http://www.globalwordnet.org/gwa/wordnet_table.html

Language	Resource name	Developer(s)	Contact	Distributor/License
Afrikaans	Afrikaans WordNet	North-West University, South Africa	Gerhard van Huyssteen	
Albanian	AlbaNet	Vlora University, Vlora, Albania	Ervin Ruci	http://fjalnet.com/shqip.xml
Arabic	Arabic WordNet	Arabic WordNet	Horacio Rodriguez	http://www.globalwordnet.org/AWN/ Free download of XML formatted DB
Arabic/English/Malaysian/Indonesian/Finnish/Hebrew/Japanese/Persian/Thai/French	Open Multilingual Wordnet	Linguistics and Multilingual Studies, Nanyang Technological University	Francis Bond	download (various open licenses)
Asian wordnet	Asian WordNet	NICT, Kyoto, Japan	Virach Sornlertlamvanich, Hitoshi Isahara	Open Source download
Malaysian/Indonesian	Wordnet Bahasa	Linguistics and Multilingual Studies, Nanyang Technological University	wn-msa-devel@lists.sourceforge.net	Open Source download (MIT license)
Bantu languages	African WordNet	University of South Africa (UNISA) in Pretoria	Sonja Bosch	.
Basque	BasquWordNet	University of the Basque Country	Eneko Agirre (eneko@si.ehu.es) aradiaz@si.ehu.es	browse online only at http://ixa2.si.ehu.es/cgi-bin/mcr/public/wei.consult.perl
Spanish-Catalan-Basque	.	Consortium of Spanish Universities	German Rigau e-mail:german.rigau AT ehu.es	browse online only at http://garraf.epsevg.upc.es/cgi-bin/wei4/public/wei.consult.perl
Bulgarian	BulNet	Institute of Bulgarian Language Bulgarian Academy of Sciences, Sofia, Bulgaria	Prof. Sv. Koeva	ELDA/ELRA
Portuguese	WordNet.PT - Portuguese WordNet	Centro de Linguística da Universidade de Lisboa	Palmira Marrafa e-mail: Palmira.Marrafa@netcabo.pt	available online only
Portuguese	OpenWN-PT (Brazilian Portuguese Wordnet	FGV/EMAp, Rio de Janeiro, Brazil	Alexandre Rademaker	download

Wordnet – em português

http://wnpt.brlcloud.com/wn/search?term=banana

banana Search

[Doc | Source | Activity | Stats | Login | API version **46-pointers-solr**]

15 results found for 'banana'

RDF Type:
- NounSynset (14)
- CoreConcept (2)
- BaseConcept (1)
- VerbSynset (1)

Lexicographer file:
- noun.plant (8)
- noun.food (6)
- verb.consumption (1)

 # words (pt_BR):
- 1 (6)
- 2 (6)
- 0 (3)

 # words (en):
- 2 (8)
- 1 (5)
- 3 (2)

Frame:
- Somebody ----s something (1)

1. 07684938-n banana_bread | **cuca, bolo de banana**
   - *(moist bread containing banana pulp)*
2. 07753592-n banana | **banana, bananeira**
   - *(elongated crescent-shaped yellow fruit with soft sweet flesh)*
3. 07616748-n banana_split | **banana split, Banana Split**
   - *(a banana split lengthwise and topped with scoops of ice cream and sauces and nuts and whipped cream)*
4. 07738570-n banana_skin, banana_peel | **casca de banana**
   - *(the skin of a banana (especially when it is stripped off and discarded); "he slipped on a banana skin and almost fell")*
5. 12352990-n Musa_paradisiaca, plantain_tree, plantain | **plantago, Banana-da-terra**
   - *(a banana tree bearing hanging clusters of edible angular greenish starchy fruits; tropics and subtropics)*
6. 12352639-n dwarf_banana, Musa_acuminata | **Banana-maçã**
   - *(low-growing Asian banana tree cultivated especially in the West Indies for its clusters of edible yellow fruit)*
7. 12353203-n Musa_paradisiaca_sapientum, edible_banana | **banana**
   - *(widely cultivated species of banana trees bearing compact hanging clusters of commercially important edible yellow fruit)*
8. 07746749-n ceriman, monstera
   - *(tropical cylindrical fruit resembling a pinecone with pineapple-banana flavor)*
9. 12352287-n banana, banana_tree | **banana, bananeira**
   - *(any of several tropical and subtropical treelike herbs of the genus Musa having a terminal crown of large entire leaves and usually bearing hanging clusters of elongated fruits)*
10. 01168468-v eat | **comer**
    - *(take in solid food; "She was eating a banana"; "What did you eat for dinner last night?")*

Similaridade entre palavra?

Similaridade entre palavras

Duas palavras são similares se **ambas compartilham** o mesmo significado.

As palavras similares mantem uma relação de significado.

- Instituição financeira:

  Banco é similar a **fundo**

- Objeto:

  Caderno é similar a **caderneta**

Porque é importante avaliar similaridade?

A similaridade de palavras pode ser útil em diferentes tipos de aplicações, como por exemplo:

- **Recuperação de Informação (IR)**

  Busca por elementos similares

- **Detecção de plágio**

  Busca por regiões similares

- **Agrupamento de textos**

  Busca por conjuntos de textos similares

  …

Porque é importante avaliar similaridade?

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand

Similaridade entre palavras e palavras correlatas

Versão mais flexível:

A similaridade entre palavras pode ser estimadada por uma medida de proximidade de significado: “Quase sinônimos”

- **Carro** é similar a **Bicicleta**

Exemplo de palavras correlatas:

- **Carro** está relacionado com **Gasolina**

Algoritmos

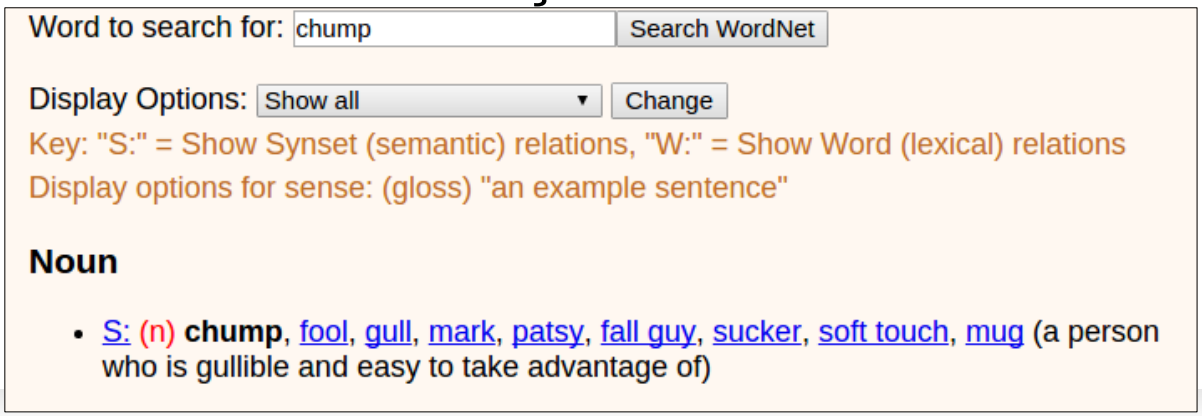
Duas abordagens para identificar similaridade entre palavras:

(1) Algoritmos baseados em tesauro:

Duas palavras são similares se uma é hiponímia de outra

- Carro é uma hiponímia de Automóvel
- Sandália é uma hiponímia de Calçado

Ou se compartilham a mesma definição



Algoritmos

Duas abordagens para identificar similaridade entre palavras:

(2) Algoritmos baseados em distribuição de palavras:

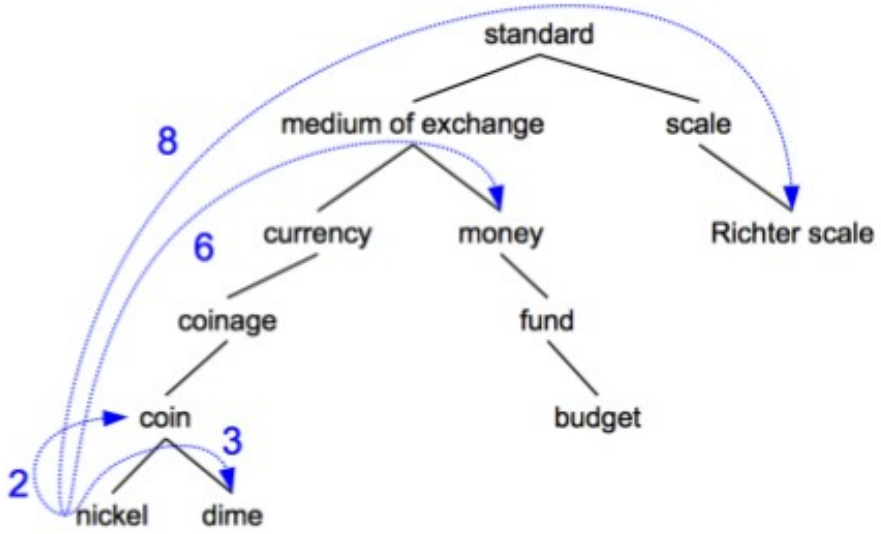
Não precisam de um tesauro, mas de um corpus grande no qual sejam evidenciados diferentes pares de palavras...

(1) Algoritmos de similaridade de palavras basedos em tesauro(s)

Similaridade usando tesauro

Denomiado de “*path based similarity*”:

Assumindo que as palavras tem comprimento igual a 1 para si mesmos



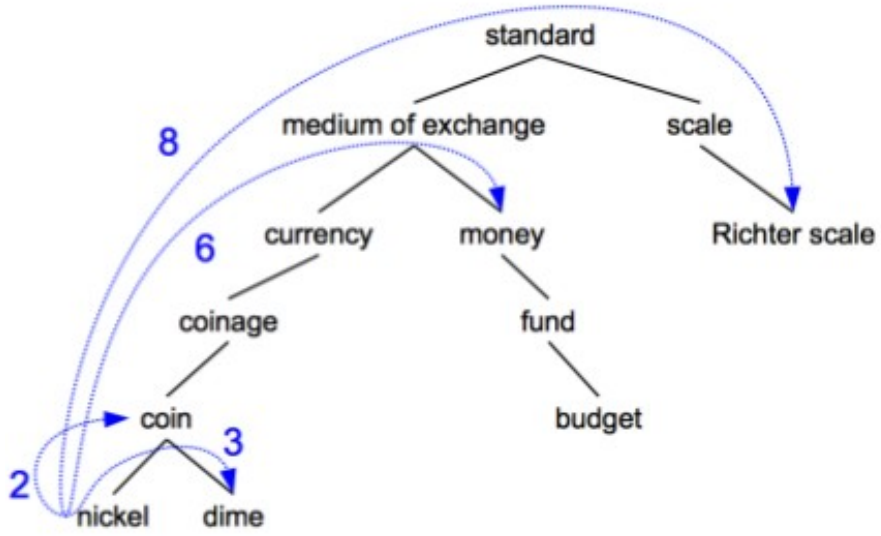
Duas palavras são similares se ambas estão na mesma hierarquia (ou bem próximas).

Pensamento computacional:

distância do menor caminho entre eles.

Formalizando as medidas

- Pathlen(c_1, c_2) = 1 + comprimento do caminho entre c_1 e c_2 na árvore de hiponímia.

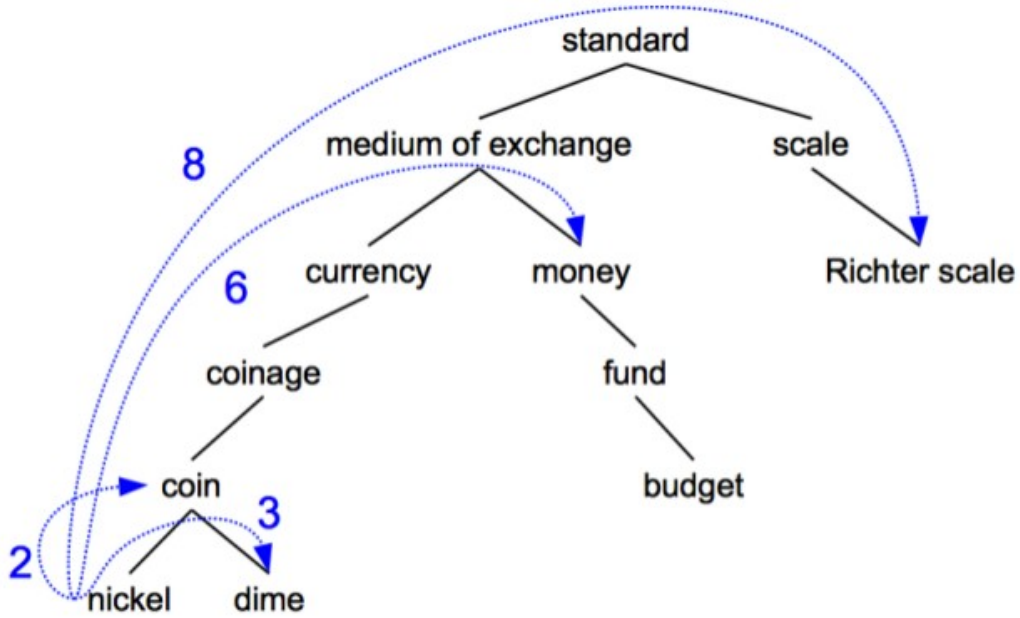


$$\text{simpath}(c_1,c_2) = \frac{1}{\text{pathlen}(c_1,c_2)}$$

$$\text{wordsim}(w_1,w_2) = \max_{c_1 \in \text{senses}(w_1), c_2 \in \text{senses}(w_2)} \text{sim}(c_1,c_2)$$

Exemplo

$$\text{simpath}(c_1,c_2) = \frac{1}{\text{pathlen}(c_1,c_2)}$$



simpath(*nickel,coin*) = 1/2 = .5

simpath(*fund,budget*) = 1/2 = .5

simpath(*nickel,currency*) = 1/4 = .25

simpath(*nickel,money*) = 1/6 = .17

simpath(*coinage,Richter scale*) = 1/6 = .17

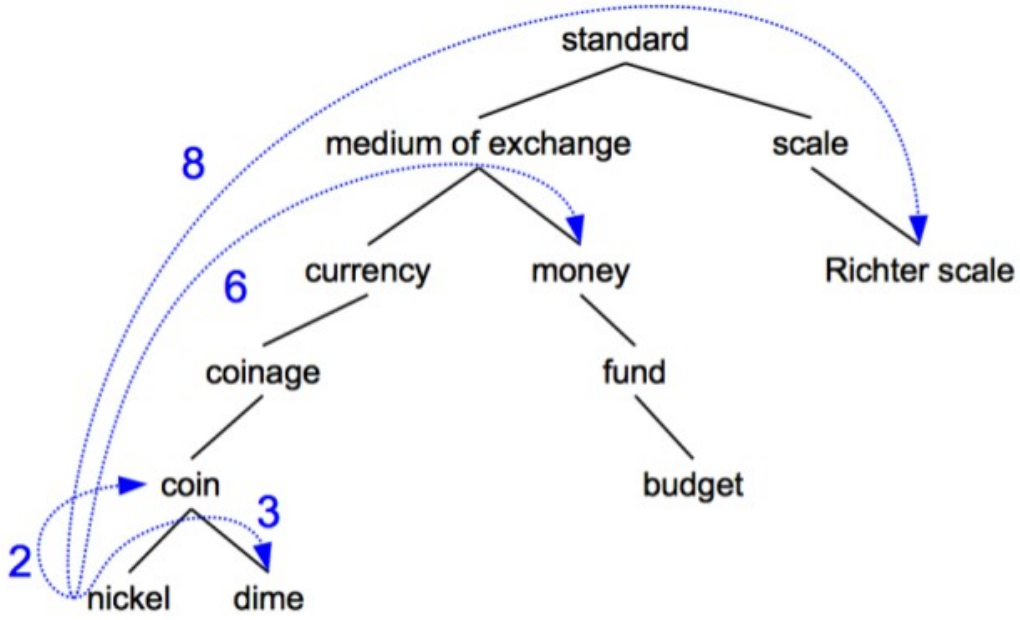
Problema

Podemos discutir um problema dessa abordagem:

Assumimos que cada aresta representa distância uniforme.

simpath(nickel, money) == simpath(nickel, standard)

Os vértices em hierarquias superiores são mais abstratos!



Problema

Podemos discutir um problema dessa abordagem:

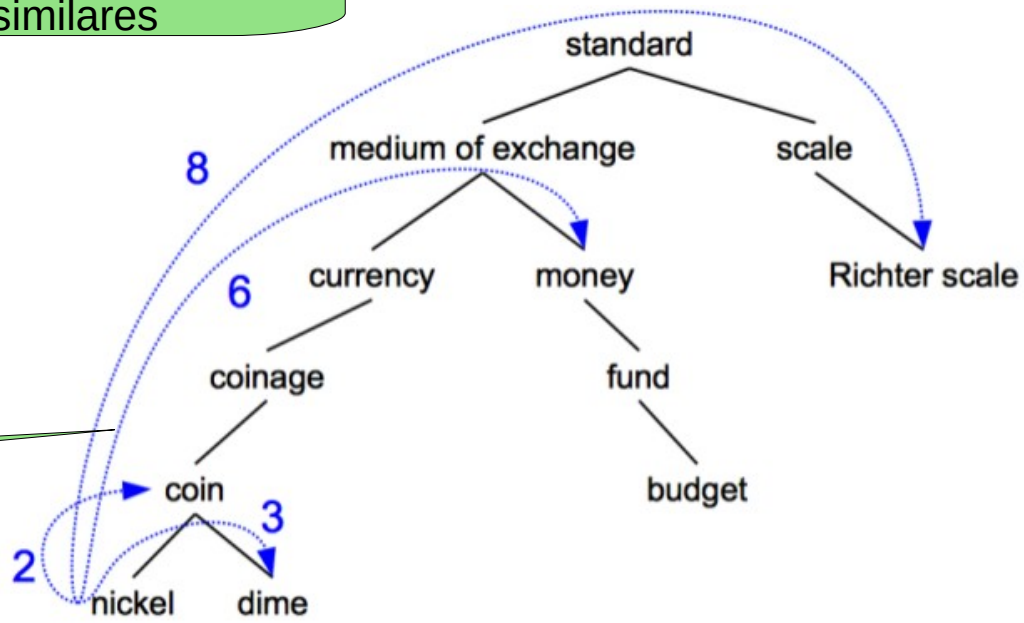
Assumimos que cada aresta representa distância uniforme.

simpath(nickel, money) == simpath(nickel, standard)

Palavras conectadas por um vértice abstrado deveriam ser menos similares

Os vértices em hierarquias superiores são mais abstratos!

Deveria ser possível representar o custo de cada aresta de forma independente



Contornando o problema

Computation and Language

Using Information Content to Evaluate Semantic Similarity in a Taxonomy

Philip Resnik

(Submitted on 29 Nov 1995)

This paper presents a new measure of semantic similarity in an IS-A taxonomy, based on the notion of information content. Experimental evaluation suggests that the measure performs encouragingly well (a correlation of r = 0.79 with a benchmark set of human similarity judgments, with an upper bound of r = 0.90 for human subjects performing the same task), and significantly better than the traditional edge counting approach (r = 0.66).

Comments:	6 pages, 2 postscript figures, uses ijcai95.sty
Subjects:	**Computation and Language (cs.CL)**
Journal reference:	Proceedings of the 14th International Joint Conference on Artificial Intelligence
Cite as:	**arXiv:cmp-lg/9511007**
	(or **arXiv:cmp-lg/9511007v1** for this version)

Submission history

From: Philip Resnik [view email]
[v1] Wed, 29 Nov 1995 19:32:04 GMT (13kb)

Utiliza um corpus para "captar" da melhor forma a distância entre 2 conceitos ou 2 palavras

Contornando o problema

An Information-Theoretic Definition of Similarity

1988

Dekang Lin
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

Abstract

Similarity is an important and widely used concept. Previous definitions of similarity are tied to a particular application or a form of knowledge representation. We present an information-theoretic definition of similarity that is applicable as long as there is a probabilistic model. We demonstrate how our definition can be used to measure the similarity in a number of different domains.

1 Introduction

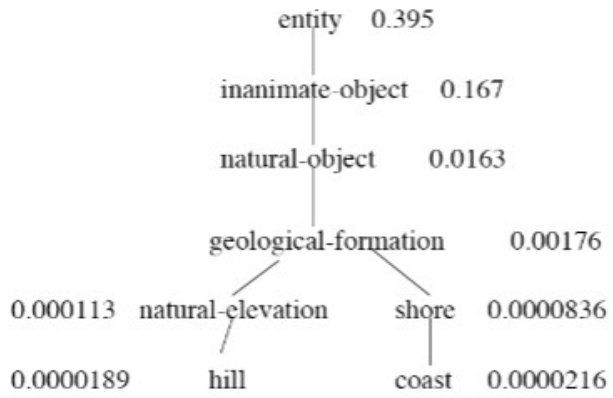
Similarity is a fundamental and widely used concept. Many similarity measures have been proposed, such as information content [Resnik, 1995b], mutual information [Hindle, 1990], Dice coefficient [Frakes and Baeza-Yates, 1992], cosine coefficient [Frakes and Baeza-Yates, 1992], distance-based measurements [Lee et al., 1989; Rada et al., 1989], and feature contrast model [Tversky, 1977]. McGill *etc.* surveyed and compared 67 similarity measures used in information retrieval [McGill et al., 1979].

A problem with previous similarity measures is that each particular measure. Almost all of the comparisons and evaluations of previous similarity measures have been based on empirical results.

This paper presents a definition of similarity that achieves two goals:

Universality: We define similarity in information-theoretic terms. It is applicable as long as the domain has a probabilistic model. Since probability theory can be integrated with many kinds of knowledge representations, such as first order logic [Bacchus, 1988] and semantic networks [Pearl, 1988], our definition of similarity can be applied to many different domains where very different similarity measures had previously been proposed. Moreover, the universality of the definition also allows the measure to be used in domains where no similarity measure has previously been proposed, such as the similarity between ordinal values.

Theoretical Justification: The similarity measure is not defined directly by a formula. Rather, it is derived from a set of assumptions about similarity. In other words, if the assumptions are deemed reasonable, the similarity measure necessarily follows.



Utiliza um corpus para "captar" da melhor forma a distância entre 2 conceitos ou 2 palavras

Contornando o problema

$$\mathrm{sim}_{\mathrm{path}}(c_1, c_2) = \frac{1}{pathlen(c_1, c_2)}$$

$$\mathrm{sim}_{\mathrm{resnik}}(c_1, c_2) = -\log P(LCS(c_1, c_2))$$

$$\mathrm{sim}_{\mathrm{lin}}(c_1, c_2) = \frac{2\log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)}$$

$$\mathrm{sim}_{\mathrm{jiangconrath}}(c_1, c_2) = \frac{1}{\log P(c_1) + \log P(c_2) - 2\log P(LCS(c_1, c_2))}$$

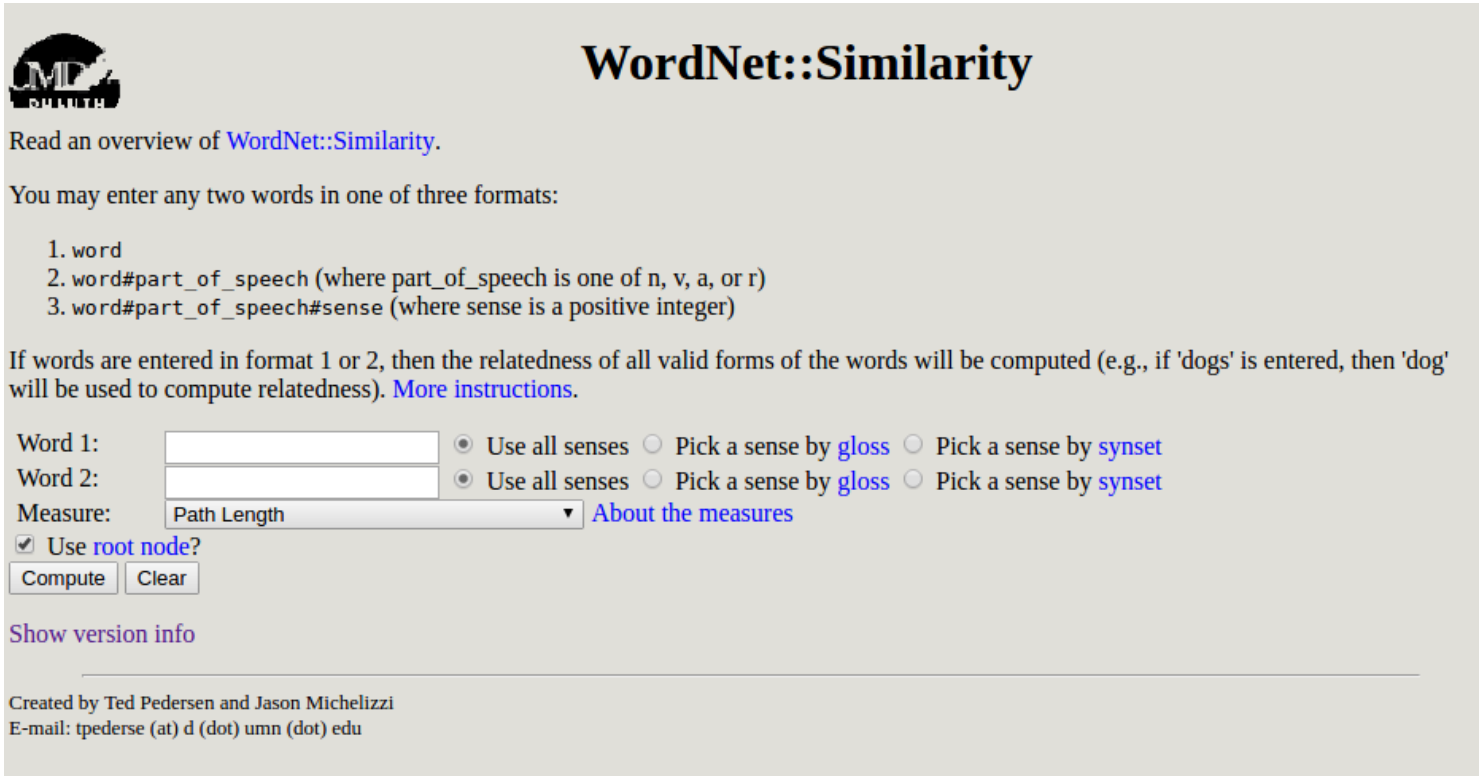
$$\mathrm{sim}_{eLesk}(c_1, c_2) = \sum_{r,q \in RELS} \mathrm{overlap}(\mathrm{gloss}(r(c_1)), \mathrm{gloss}(q(c_2)))$$

Interfaces

NLTK oferece métodos para cálculo de similaridade de palavras baseada em wordnet.

Por outro lado existem outras iniciativas on line:

http://maraca.d.umn.edu/cgi-bin/similarity/similarity.cgi



Interfaces

The relatedness of dog#n#1 and cat#n#1 using path is 0.2.

View relatedness of all senses (without traces)

View relatedness of all senses (with traces)

View traces

You may enter any two words in one of three formats:

1. `word`
2. `word#part_of_speech` (where part_of_speech is one of n, v, a, or r)
3. `word#part_of_speech#sense` (where sense is a positive integer)

If words are entered in format 1 or 2, then the relatedness of all valid forms of the words will be computed (e.g., if '
will be used to compute relatedness). More instructions.

Word 1: dog — Use all senses / Pick a sense by gloss / Pick a sense by synset

Word 2: cat — Use all senses / Pick a sense by gloss / Pick a sense by synset

Measure: Path Length — About the measures

Use root node?

Compute Clear

Show version info

Lin

The relatedness value returned by the lin measure is a number equal to 2 * IC(lcs) / (IC(synset1) + IC(synset2)
information content of x. One can observe, then, that the relatedness value will be greater-than or equal-to zero

If the information content of any of either synset1 or synset2 is zero, then zero is returned as the relatedness sc
the information content of a synset would be zero only if that synset were the root node, but when the frequenc
value of zero as the information content because of a lack of better alternatives.

Adapted Lesk (Extended Gloss Overlaps)

The Extended Gloss Overlaps measure (lesk) works by finding overlaps in the glosses of the two synsets. The
the squares of the overlap lengths. For example, a single word overlap results in a score of 1. Two single word
A two word overlap (i.e., two consecutive words) results in a score of 4. A three word overlap results in a score

Gloss Vector

The Gloss Vector measure (vector) works by forming second-order co-occurrence vectors from the glosses or
The relatedness of two concepts is determined as the cosine of the angle between their gloss vectors. In order t
issues presented by extremely short glosses, this measure augments the glosses of concepts with glosses of adj
WordNet relations.

Gloss Vector (pairwise)

The Gloss Vector (pairwise) measure (vector_pairs) is very similar to the "regular" Gloss Vector measure, exce
glosses of concepts with adjacent glosses. The regular Gloss Vector measure first combines the adjacent glosse
gloss" and creates a single vector corresponding to each of the two concepts from the two "super-glosses". The
on the other hand, forms separate vectors corresponding to each of the adjacent glosses (does not form a single
separate vectors will be created for the hyponyms, the holonyms, the meronyms, etc. of the two concepts. The
the individual cosines of the corresponding gloss vectors, i.e. the cosine of the angle between the hyponym vec
angle between the hlonym vectors, and so on. From empirical studies, we have found that the regular Gloss Ve
than the pairwise Gloss Vector measure.

Hirst & St-Onge

This measure (hso) works by finding lexical chains linking the two word senses. There are three classes of rela
strong, strong, and medium-strong. The maximum relatedness score is 16.

Random

The relatedness values are simply randomly generated numbers. This is intended only to be used as a baseline.

(2) Algoritmos de similaridade de palavras basedos em distribuição de palavras

Porque?

Por que é necessário este tipo de abordagem?

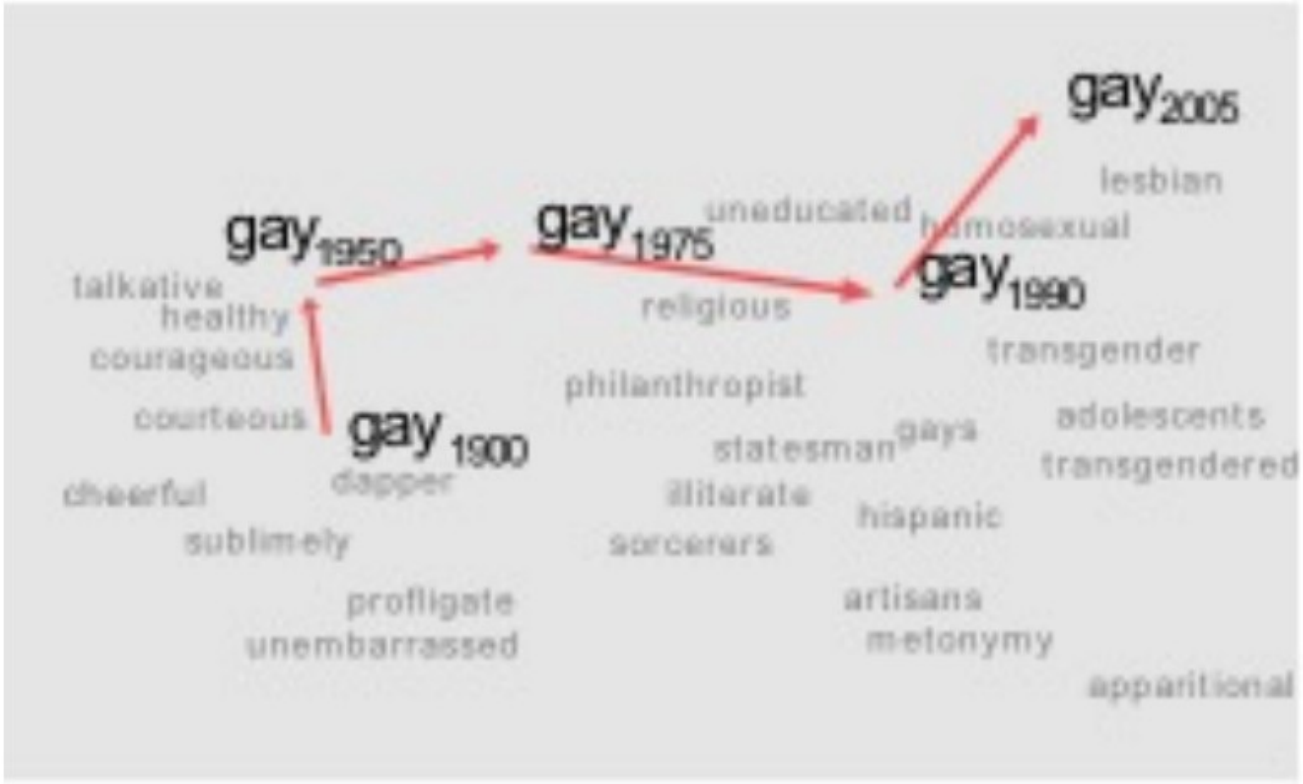
- As métricas apresentadas nos slides anteriores são **dependentes** de um tesauro.
- Dependem da **completude** das palavras (dicionário), ie. não são flexíveis.
- No Tesauro algumas relações não estão representadas.
- Adjetivos e verbos são menos representados nos tesauros:

Category	Unique Strings
Noun	117,798
Verb	11,529
Adjective	22,479
Adverb	4,481

Wordnet Versão 3.0, contem mais substantivos

Por que é necessário este tipo de abordagem?

Kulkarni, Al-Rfou, Perozzi, Skiena 2015



A semantica muda/evolue ao longo do tempo

Abordagem baseada em distribuição de palavras

Obras similares estão "geralmente" próximas

Em PLN:

Palavras que estão em contextos similares, tendem a ser semanticamente similares

Abordagem baseada em distribuição de palavras

Na literatura isso é conhecido como:

- ***Distributional semantics.***
- ***Vector semantics.***

O significado de uma palavra é calculada **a partir da distribuição de palavras** que estão ao redor dela.

As palavras são representadas como um **vetor de números**.

Abordagem baseada em distribuição de palavras

Zellig Harris (1954): "**oculist** and **eye-doctor** … occur in almost the same environments…. **If A and B have almost identical environments we say that they are synonyms**.

Firth (1957): "You shall know a word by the company it keeps!"

Abordagem baseada em distribuição de palavras

A bottle of ***tesgüino*** is on the table

Everybody likes ***tesgüino***

Tesgüino makes you drunk

We make ***tesgüino*** out of corn.

Abordagem baseada em distribuição de palavras

A bottle of ***tesgüino*** is on the table
Everybody likes ***tesgüino***
Tesgüino makes you drunk
We make ***tesgüino*** out of corn.

Podemos não saber o que é “tesguino” (certamente não estará presente em algum tesauro), mas pelo contexto podemos intuir que trata-se de uma bebida alcoólica.

--> Duas palavras serão similares se ambas estão em contextos similares.

Matriz termo-documento

Matriz: termo-documento

Dois **documentos** são similares se os vetores são similares

V

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

A dimensão do vetor é
o tamanho do vocabulário: $N^{|V|}$

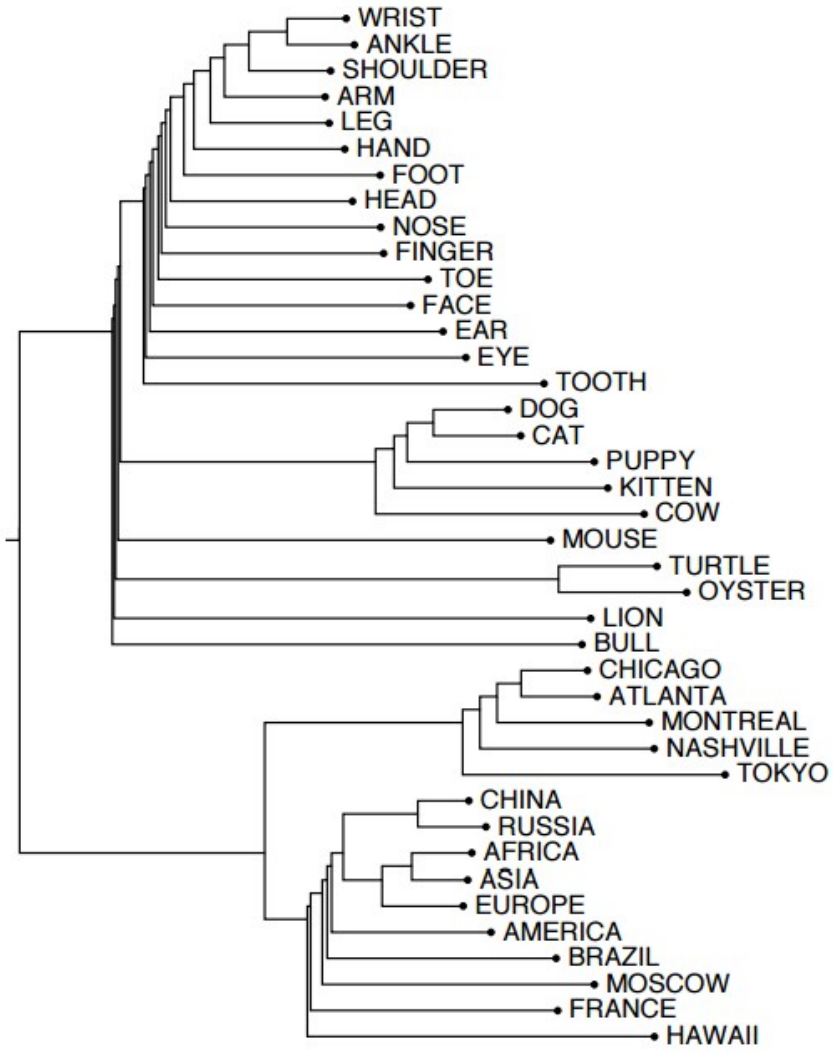
Matriz: termo-documento

Duas **palavras** são similares se os vetores são similares

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

A dimensão do vetor é o número de documentos: $N^{|D|}$

Agrupamento hierárquico



Capturar significa relacional

vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')