

Modelos de Linguagem Neurais

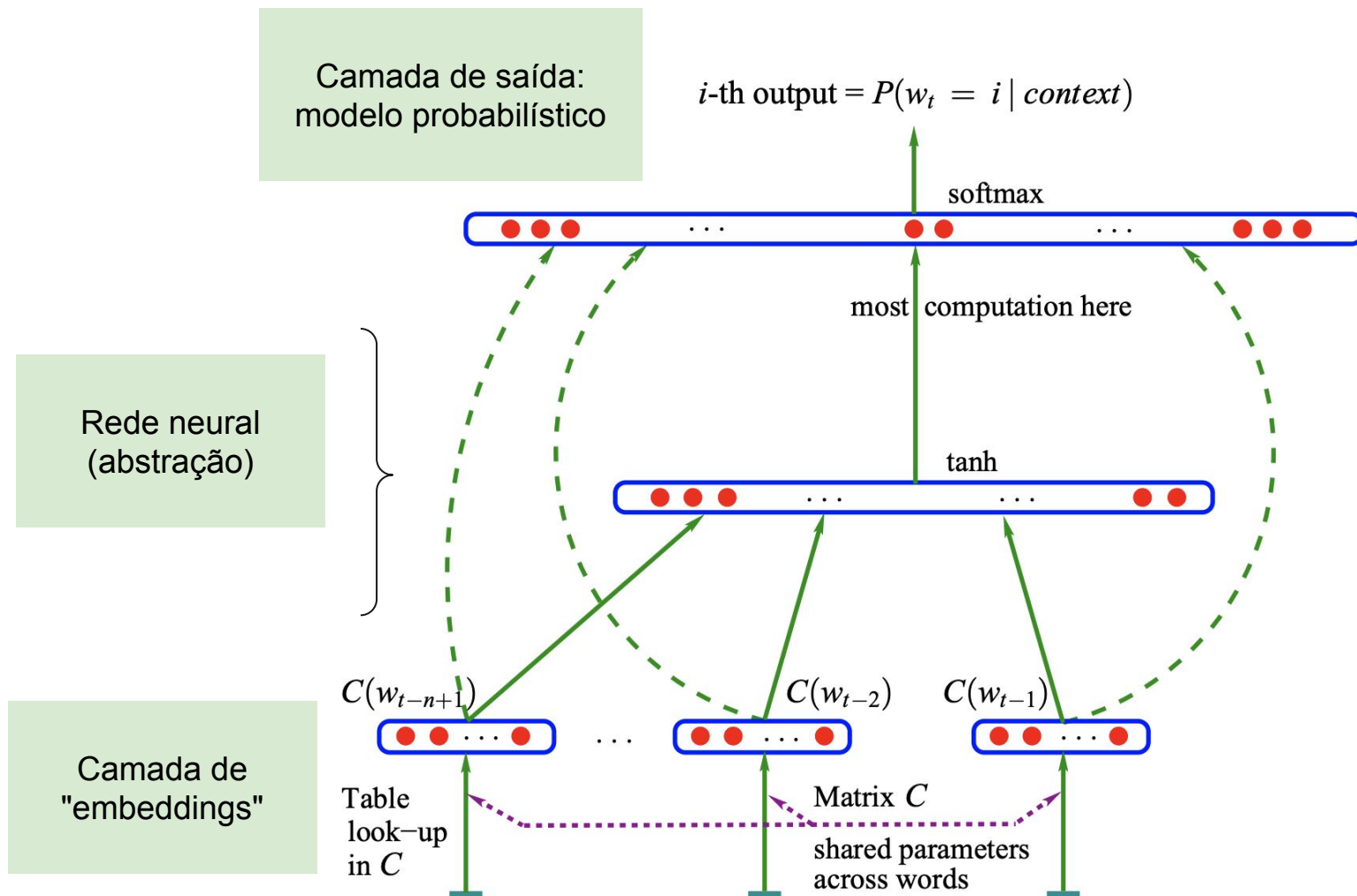
Semântica vetorial e embeddings estáticos



Profa Aline Paes
alinepaes@ic.uff.br

Entender o **ontem** fazer o **agora** para inovar no **amanhã**

Modelo de Linguagem Neural



Como representar palavras no computador?

- Um algoritmo para processar um texto : uma sequência de caracteres (strings)
- Um vocabulário: uma lista de strings

1	a
2	ab
3	aba
...	...
11494	lugar
...	...

Indexação de palavras

Eu	fui	ao	cinema
1235	2456	20	459

Semântica vetorial

Eu	fui	ao	cinema
1235	2456	20	459

...	
20	
...	
459	
....	
1235	
...	
2456	
...	

Semântica vetorial

Eu	fui	ao	cinema
1235	2456	20	459

...	
20	
...	
459	
....	
1235	
...	
2456	
...	

Semântica vetorial

Eu	fui	ao	cinema
----	-----	----	--------

1235	2456	20	459
------	------	----	-----

...	
20	
...	
459	
....	
1235	
...	
2456	
...	

Semântica vetorial

Eu	fui	ao	cinema
1235	2456	20	459

...	
20	
...	
459	
....	
1235	
...	
2456	
...	

Semântica de palavras

- Como representar “significado” de palavras no computador?

Semântica de palavras

- Como representar “significado” de palavras no computador?
 - Recuperando de um recurso linguístico

Semântica de palavras

- Wordnet

```
from nltk.corpus import wordnet as wn
poses = { 'n': 'noun', 'v': 'verb', 's': 'adj (s)', 'a': 'adj', 'r': 'adv' }
for synset in wn.synsets("good"):
    print("{}: {}".format(poses[synset.pos()],
        ", ".join([l.name() for l in synset.lemmas()])))
```

```
noun: good
noun: good, goodness
noun: good, goodness
noun: commodity, trade_good, good
adj: good
adj (sat): full, good
adj: good
adj (sat): estimable, good, honorable, respectable
adj (sat): beneficial, good
adj (sat): good
adj (sat): good, just, upright
...
adverb: well, good
adverb: thoroughly, soundly, good
```

polissemia

```
from nltk.corpus import wordnet as wn
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

Wordnet

- Problemas

- Não aborda **contexto e contextoS**
 - *Commodity* é sinônimo de *good*
 - Só é correto em alguns contextos
- Não é adaptada **automaticamente**
 - Novas palavras ou novos significados podem não estar lá
- Não dispõe de mecanismos para computar **similaridade**
 - Não apenas para sinônimos
 - Gato e cachorro

Semântica vetorial

- One hot
 - Vetor de dimensão $|V|$ que associa '1' para apenas uma posição e o resto é '0'
 - Cada palavra terá '1' em uma posição diferente
- hotel = [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
- pousada = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 - Vetores ortogonais
 - Não capturam similaridade ou semântica
 - Representação esparsa

Você sabe o que é um **tezgüino**?



Você sabe o que é um tezgüino?

Uma garrafa de tezgüino está na mesa.

Todo mundo gosta de tezgüino.

Tezgüino te faz ficar bêbado.

Tezgüino é feito de milho.

Você sabe o que é um **tezgüino**?

Uma garrafa de **tezgüino** está na mesa.

Todo mundo gosta de **tezgüino**.

Tezgüino te faz ficar bêbado.

Tezgüino é feito de milho.

E agora? Você sabe?

Que outras palavras cabem aqui?

Uma garrafa de _____ está na mesa.
Todo mundo gosta de _____.
_____ te faz ficar bêbado.
_____ é feito de milho.

	(1)	(2)	(3)	(4)
tezgüino	1	1	1	1
água	1	0	0	0
Óleo de motor	0	0	0	1
bolo	0	1	0	1
vinho	1	1	1	0

Contextos

Que outras palavras cabem aqui?

Uma garrafa de _____ está na mesa.
Todo mundo gosta de _____.
_____ te faz ficar bêbado.
_____ é feito de milho.

	(1)	(2)	(3)	(4)
tezgüino	1	1	1	1
água	1	0	0	0
Óleo de motor	0	0	0	1
bolo	0	1	0	1
vinho	1	1	1	0

Contextos

Qual é o
mais
parecido?

Que outras palavras cabem aqui?

Uma garrafa de _____ está na mesa.
Todo mundo gosta de _____.
_____ te faz ficar bêbado.
_____ é feito de milho.

	(1)	(2)	(3)	(4)
tezgüino	1	1	1	1
água	1	0	0	0
Óleo de motor	0	0	0	1
bolo	0	1	0	1
vinho	1	1	1	0

Contextos

Qual é o
mais
próximo?

Hipótese distribucional

Capturar significado e capturar contexto são essencialmente a mesma coisa

Semântica distribucional

- *“The meaning of a word is its use in the language”* (Wittgenstein, 1953).
- *“You shall know a word by the company it keeps”* (Harris ,1954)
- Palavras que ocorrem em contextos similares tendem a ter significados similares: hipótese distribucional (Joos, 1950; Harris, 1954; Firth, 1957)

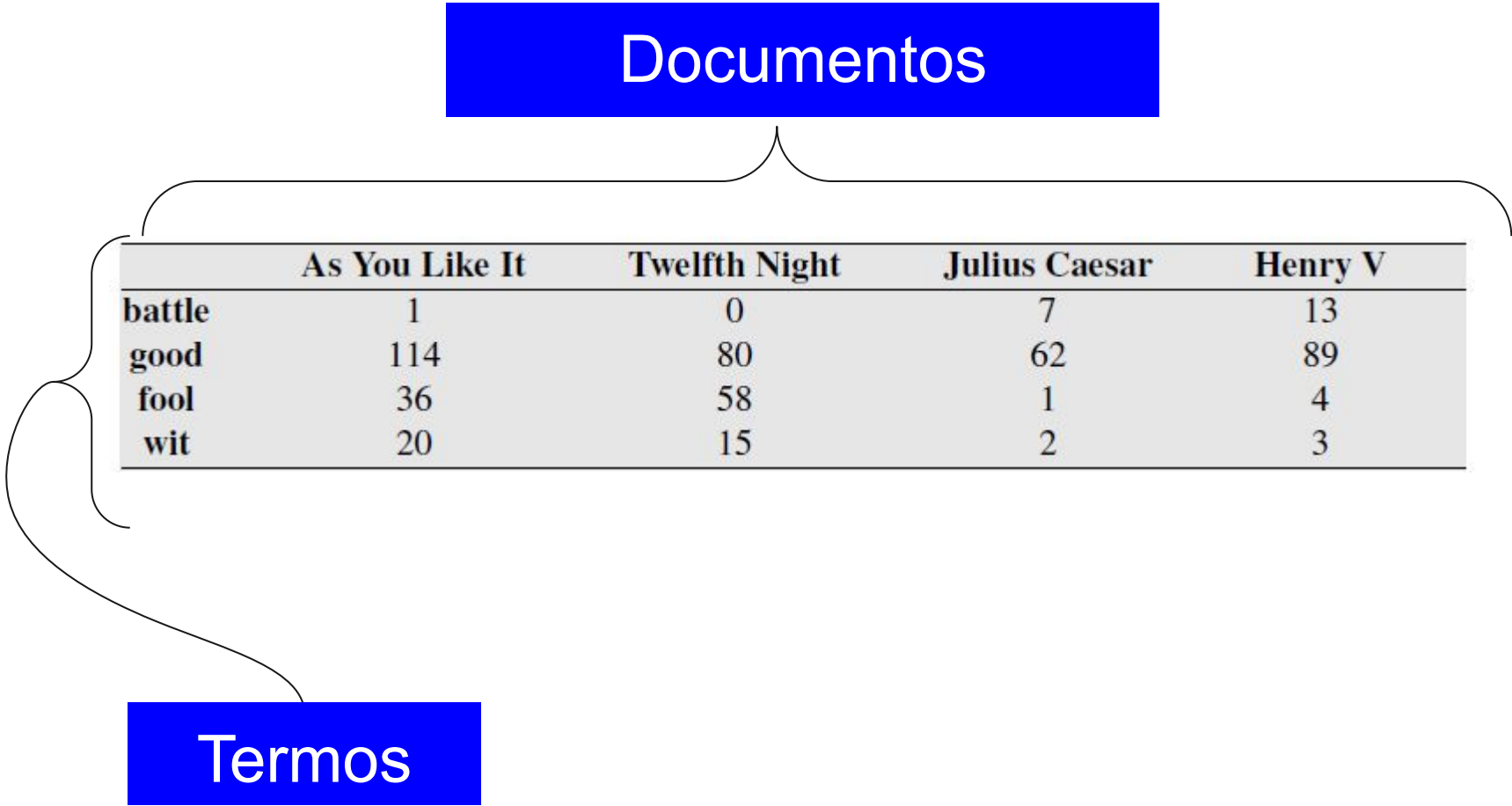
Contexto e valor

O que é o contexto?

Como atribuir valor para a palavra
que reflita contextos?

Matriz de coocorrência (termo-documento)

Documentos

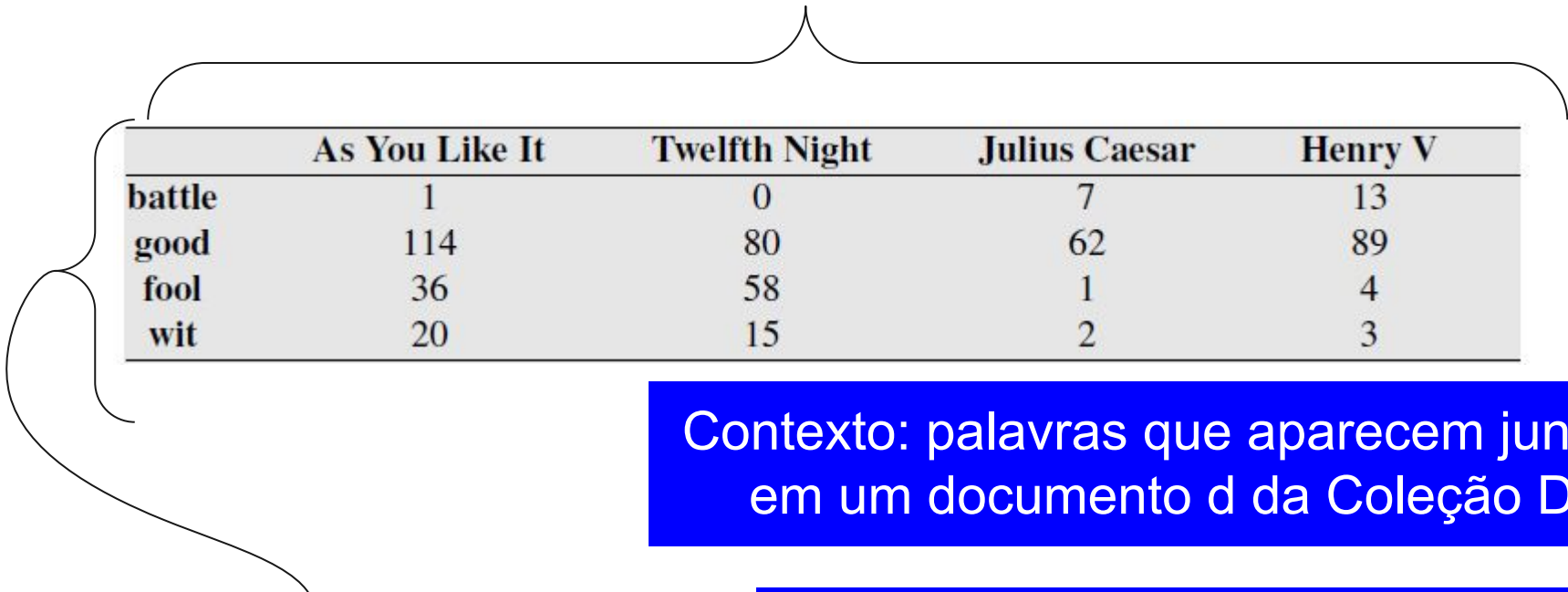


	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Termos

Matriz de coocorrência (termo-documento)

Documentos



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

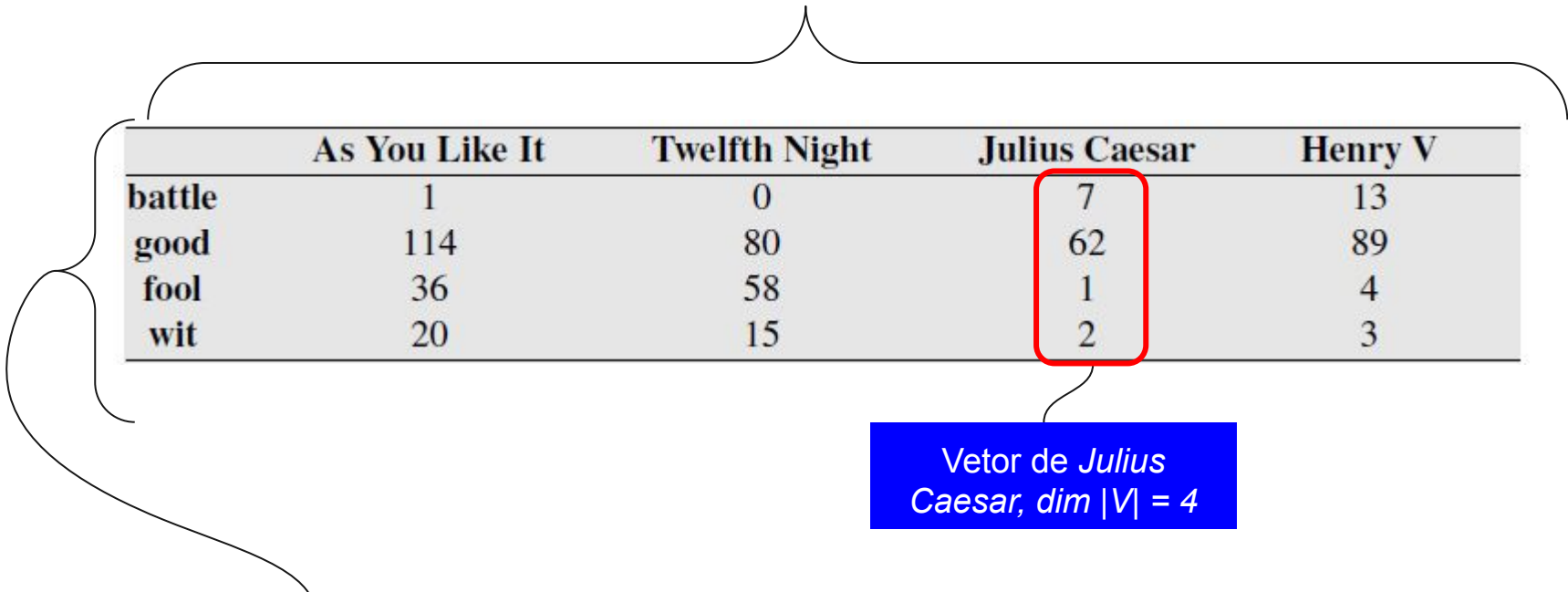
Contexto: palavras que aparecem juntas
em um documento d da Coleção D

Termos

Valor: quantas vezes elas
aparecem juntas

Documentos como vetores

Documentos



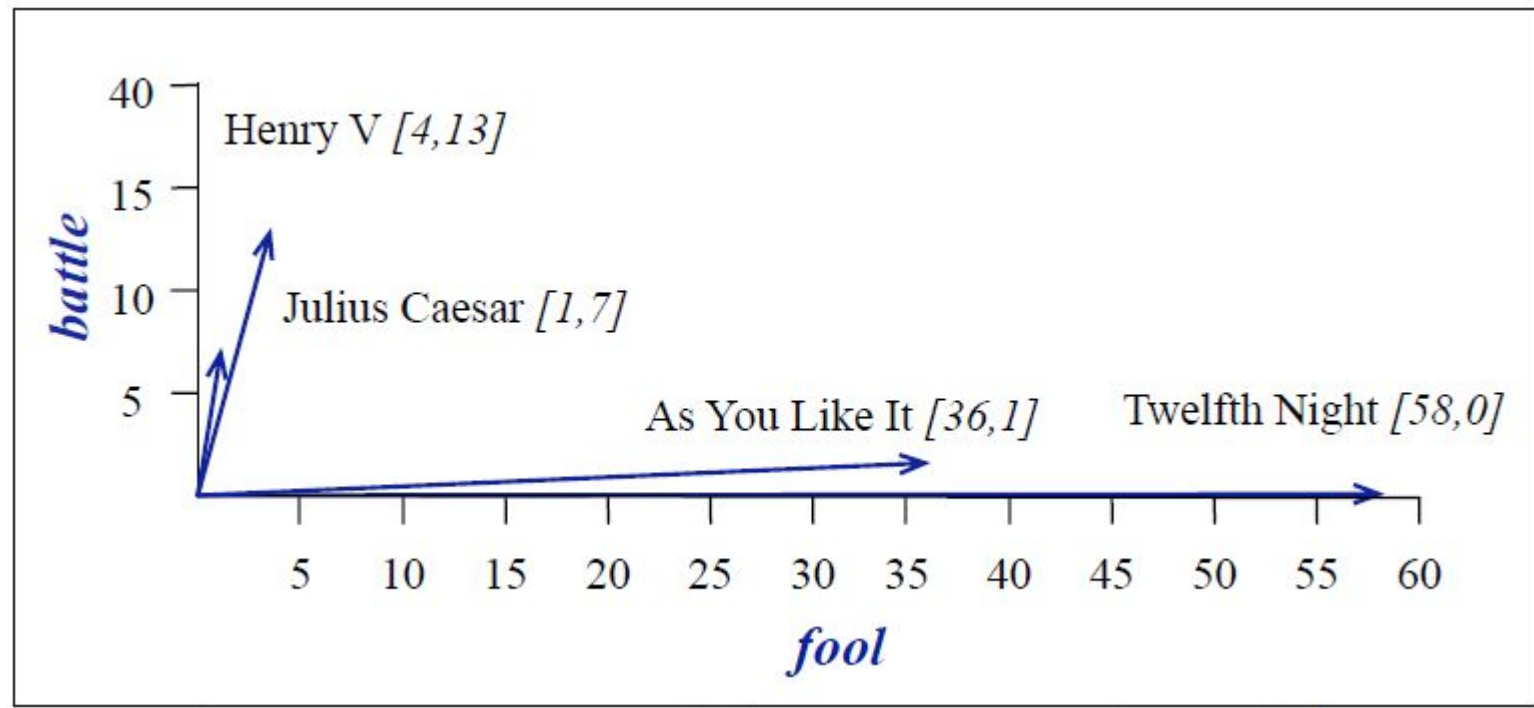
	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Vetor de *Julius Caesar*, $\dim |V| = 4$

Termos

Espaço vetorial: coleção de vetores, caracterizado pela dimensão

Visualização em 2D



Palavras como vetores

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Matriz de coocorrência (termo-termo)

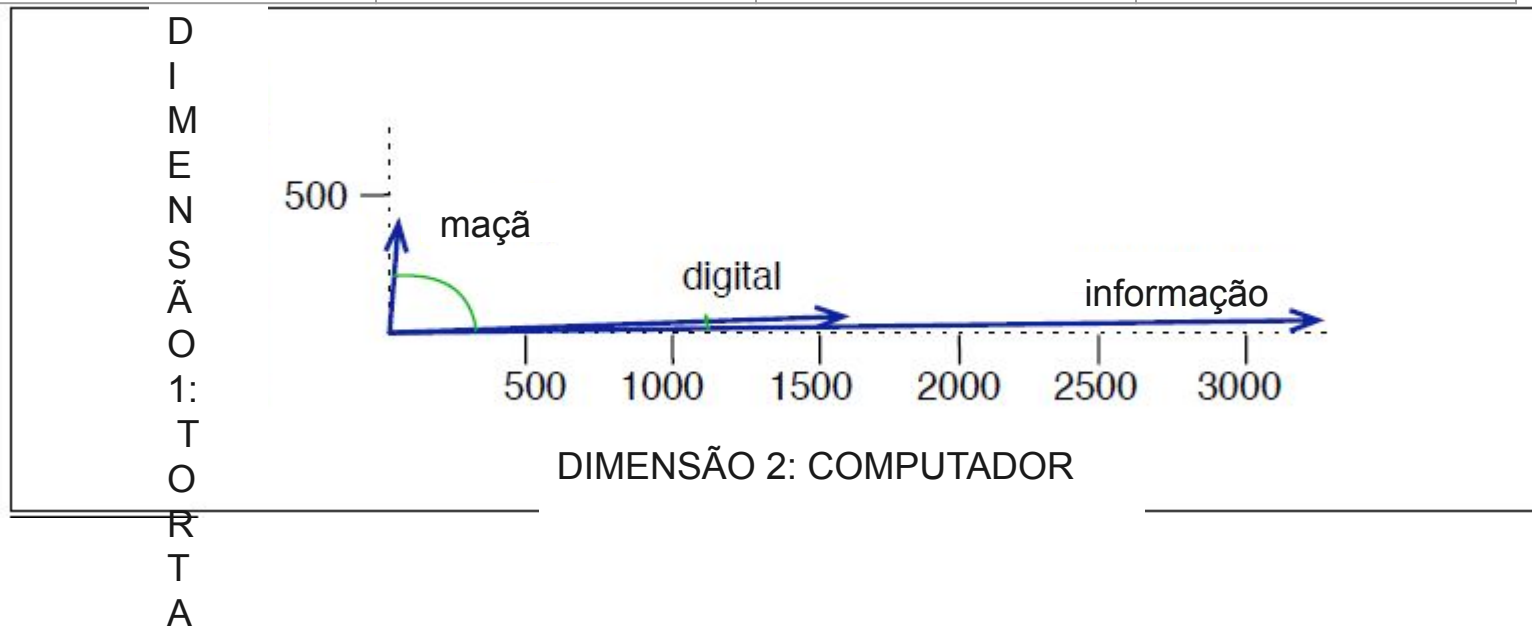
- Matriz $|V| \times |V|$
 - Cada célula representa quantas vezes as palavras na [linha,coluna] aparecem juntas
 - No mesmo documento
 - Ou em uma janela

Matriz de coocorrência (termo-termo)

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	

Visualização em 2D

	torta	dados	computador
maçã	442	8	2
digital	5	1683	1670
informação	5	3982	3325



Calculando similaridade

$$\text{dot product}(\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^N v_i w_i = v_1 w_1 + v_2 w_2 + \dots + v_N w_N$$

- Tende a ser alto apenas quando os dois vetores têm valores altos nas mesmas dimensões
- Vetores que têm zeros em diferentes dimensões (ortogonais) terão um produto escalar de 0, representando sua forte dissimilaridade.

Calculando similaridade

$$\text{dot product}(\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^N v_i w_i = v_1 w_1 + v_2 w_2 + \dots + v_N w_N$$

- Tende a ser alto apenas quando os dois vetores têm valores altos nas mesmas dimensões
- Vetores que têm zeros em diferentes dimensões (ortogonais) terão um produto escalar de 0, representando sua forte dissimilaridade.
- **Favorece vetores longos**

Calculando similaridade

$$\text{dot product}(\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^N v_i w_i = v_1 w_1 + v_2 w_2 + \dots + v_N w_N$$

- O produto interno bruto será maior para palavras frequentes
 - Palavras mais frequentes têm vetores maiores
 - Tendem a coocorrer com mais palavras
 - Têm valores de coocorrência mais altos

Similaridade de cosseno

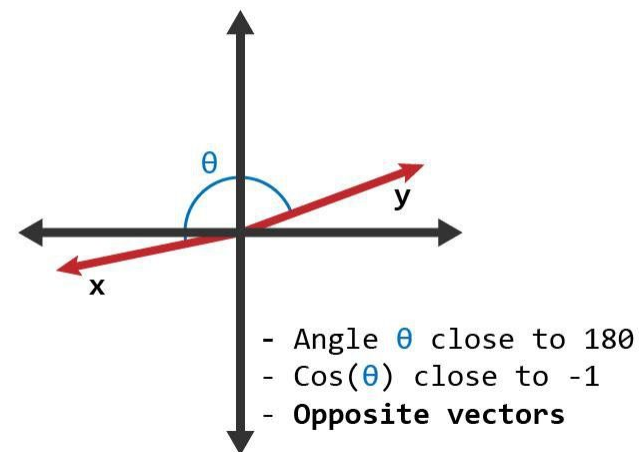
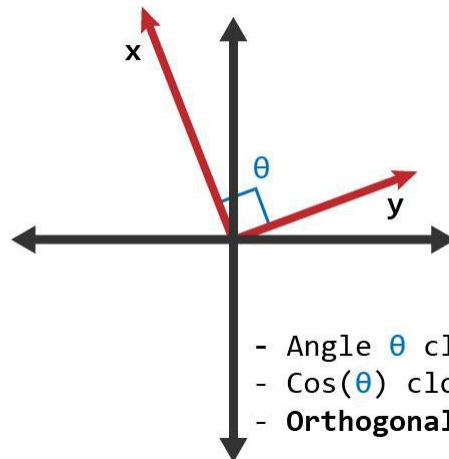
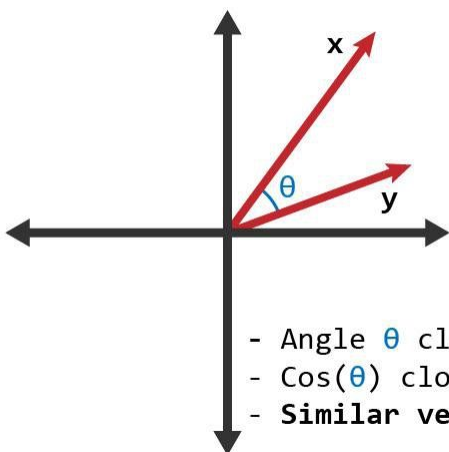
- Produto interno normalizado = cosseno do ângulo entre os dois vetores

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta$$
$$\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} = \cos \theta$$

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}| |\mathbf{w}|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

Similaridade de cosseno

- Produto interno normalizado = cosseno do ângulo entre os dois vetores

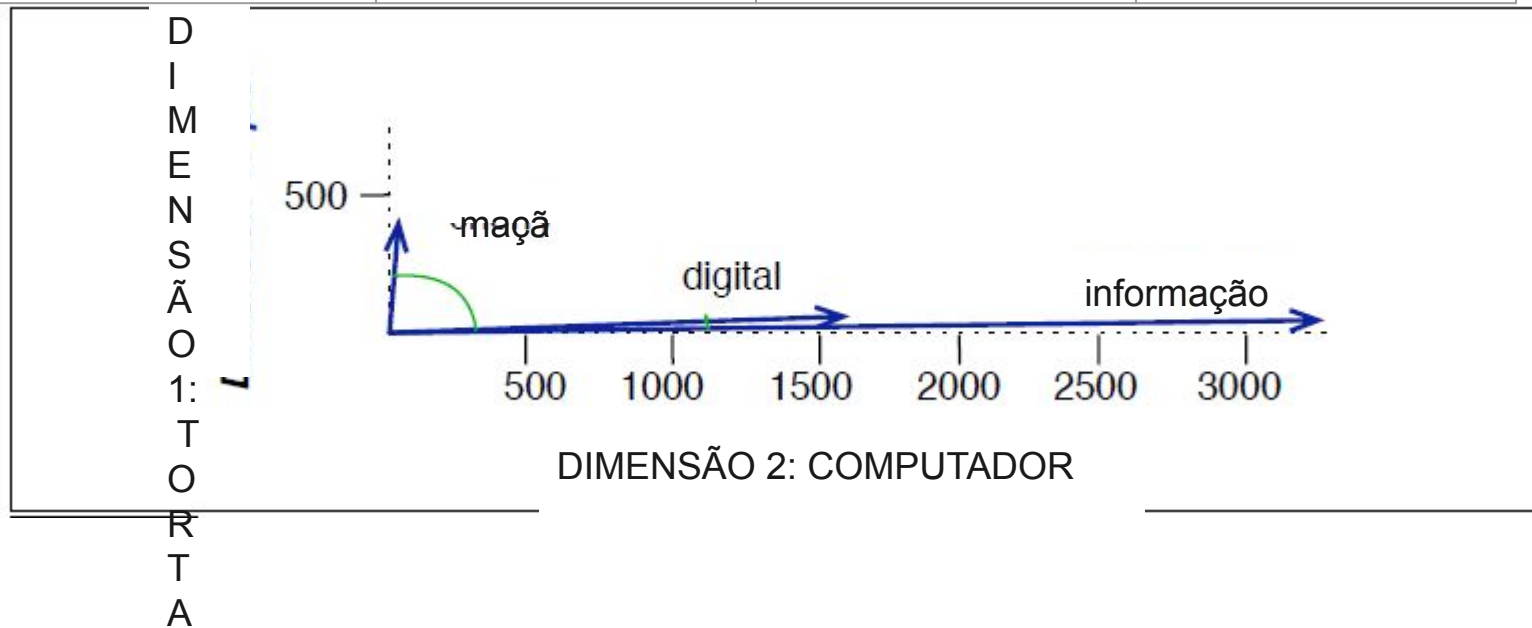


Pergunta

- A similaridade de cosseno dos valores vistos até agora estarão entre -1 e 1?
 - SIM
 - Não
 - Depende

Similaridade de cosseno

	torta	dados	computador
maçã	442	8	2
digital	5	1683	1670
informação	5	3982	3325



Similaridade de cosseno

	torta	dados	computador
maçã	442	8	2
digital	5	1683	1670
informação	5	3982	3325

$$\cos(\text{maçã}, \text{informação}) = \frac{442 * 5 + 8 * 3982 + 2 * 3325}{\sqrt{442^2 + 8^2 + 2^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .017$$

$$\cos(\text{digital}, \text{informação}) = \frac{5 * 5 + 1683 * 3982 + 1670 * 3325}{\sqrt{5^2 + 1683^2 + 1670^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .996$$

Matriz de coocorrência (termo-termo)

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	

Vetores esparsos e
com valores absolutos
distantes, alta
dimensionalidade e
contexto longo

TF-IDF

Contexto: palavras que aparecem juntas em um documento d , dada uma coleção D

Valor: $\text{tf-idf}(w, d, D) = \text{tf}(w, d) * \text{idf}(w, D)$

$$\text{tf-idf}(w, d, D) = \text{tf}(w, d) * \text{idf}(w, D) = N(w, d) * \text{idf}(w, D)$$

$$\text{tf-idf}(w, d, D) = N(w, d) * \log \frac{|D|}{|d \in D : w \in d|}$$

TF-IDF

Contexto: palavras que aparecem
juntas em um documento d , dada uma
coleção D

Valor: $\text{tf-idf}(w,d,D) = \text{tf}(w,d) * \text{idf}(w,D)$

$$\text{tf-idf}(w$$

Vetores ainda podem ser esparsos, com uma janela longa de contexto, e alta dimensionalidade

tf-

$$) * \text{idf}(w, D)$$
 $v \in d|$

Positive Pointwise Mutual Information (PPMI)

Contexto: palavras que aparecem juntas em uma janela de tamanho L

$$\text{PPMI}(w,c) = \max(0, \text{PMI}(w,c))$$

$$\text{PMI}(w, c) = \log \frac{P(w, c)}{P(w)P(c)} = \log \frac{N(w, c)|V|}{N(w)N(c)}$$

Positive Pointwise Mutual Information (PPMI)

Contexto: palavras que aparecem juntas em uma janela de tamanho L

$$\text{PPMI}(w, c) = \max(0, \text{PMI}(w, c))$$

$$\text{PMI}(w, c) =$$

Vetores ainda podem ser esparsos, e com alta dimensionalidade

$$\frac{N(w, c) |V|}{N(w) N(c)}$$

Semântica vetorial

- Matrizes baseadas em contagem são esparsas e com alta dimensionalidade
 - Ruim para usar com algoritmos de ML
 - Mais difícil verificar similaridades

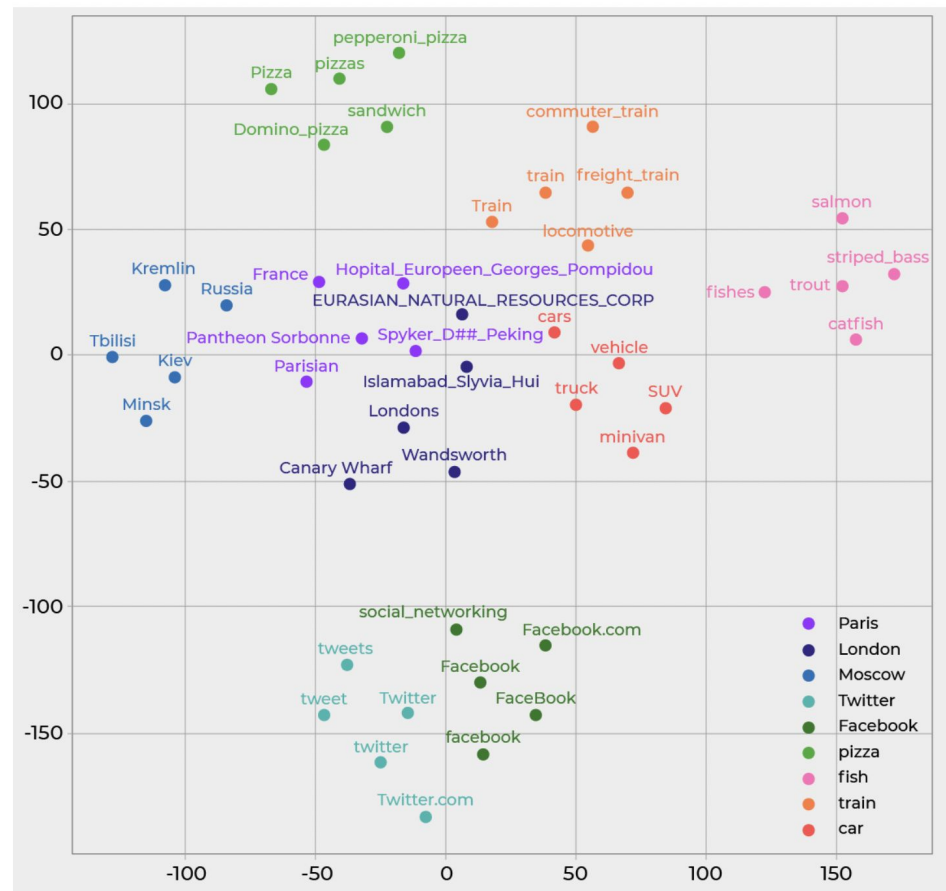
Semântica vetorial

- Matrizes baseadas em contagem são esparsas e com alta dimensionalidade
 - Ruim para usar com algoritmos de ML
 - Mais difícil verificar similaridades
- Como obter um vetor denso de baixa dimensionalidade?
- E que ainda capture contexto?

0.286
0.792
-0.177
-0.107
0.109
-0.542
0.349
0.271

Espaço de similaridades

- **Palavras similares** são representados por vetores que estão próximos no espaço vetorial criado



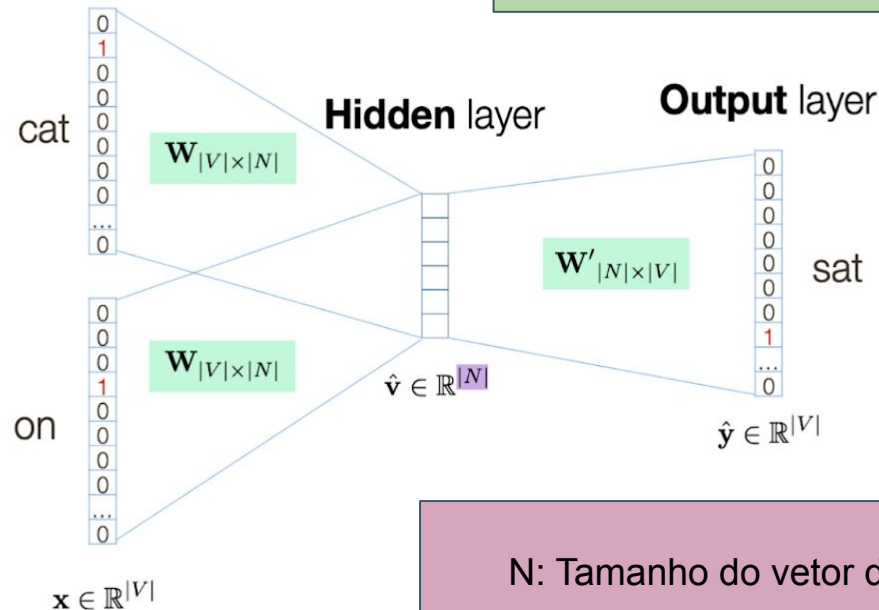
Word similarity according to word2vec

Word2Vec

- Pacote de SW com dois algoritmos
 - Continuous BOW (CBOW)
 - SkipGram
- Ao invés de contar quantas vezes cada palavra ocorre perto de outra, **treinamos um classificador** em uma tarefa de previsão binária : “*a palavra X é provável de aparecer com a palavra Y*”?
- A predição não é importante -- os pesos do classificador, sim -- serão os **embeddings**

De onde vêm os vetores?

Input layer



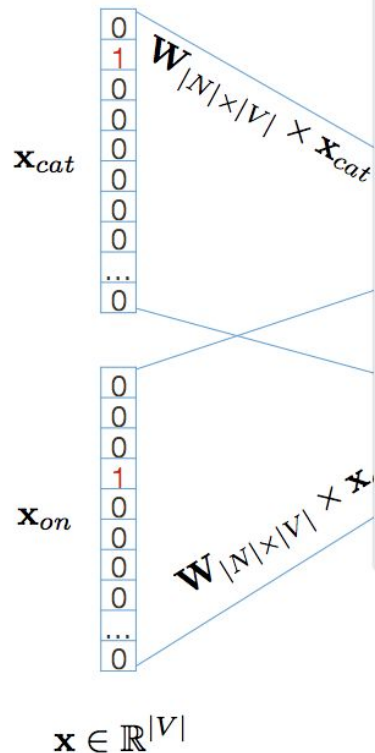
\mathbf{W} : Parâmetros/pesos a serem *aprendidos* por uma rede neural

N : Tamanho do vetor de palavras

Slides originais: <https://www.cs.ubc.ca/~lsigal/532L/Lecture7.pdf>

De onde vêm os vetores?

Input layer



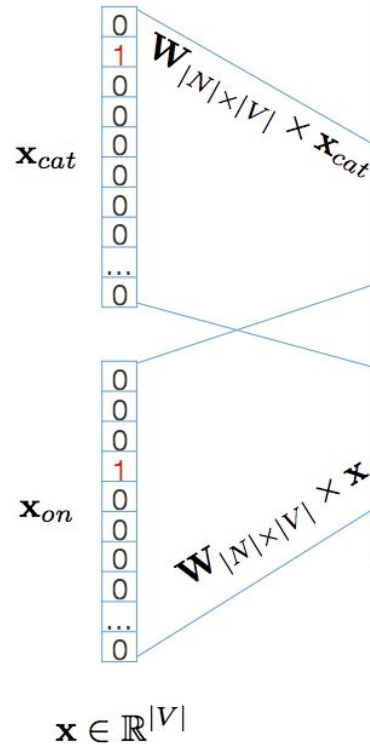
$$\mathbf{W}_{|V| \times |N|}^T \times \mathbf{x}_{cat} = \mathbf{v}_{cat}$$

0.1	2.4	1.6	1.8	0.5	0.9	3.2
0.5	2.6	1.4	2.9	1.5	3.6	6.1
...
...
0.6	1.8	2.7	1.9	2.4	2.0	1.2

$$\times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.4 \\ 2.6 \\ \dots \\ \dots \\ 1.8 \end{bmatrix}$$

De onde vêm os vetores?

Input layer



$$\mathbf{W}_{|V| \times |N|}^T \times \mathbf{x}_{on} = \mathbf{v}_{on}$$

0.1	2.4	1.6	1.8	0.5	0.9	3.2
0.5	2.6	1.4	2.9	1.5	3.6	6.1
...
...
0.6	1.8	2.7	1.9	2.4	2.0	1.2

$$\times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.8 \\ 2.9 \\ \dots \\ \dots \\ 1.9 \end{bmatrix}$$

Word2Vec

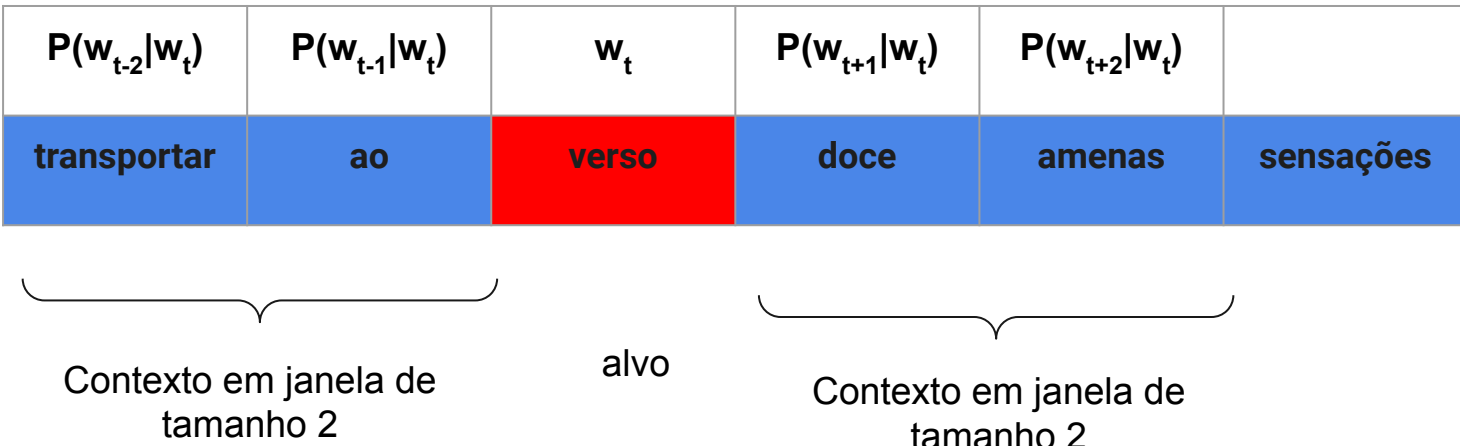
- Ideia geral

- A partir de um corpus grande
- Todas as palavras do vocabulário são representadas como um **vetor (embeddings)**
- Vá em cada posição t do texto, que tem uma palavra **alvo** e palavras no **contexto**
 - Exemplo positivo: (alvo, contexto)
 - Exemplo negativo: amostra de outras palavras
- Use a **similaridade** dos vetores para calcular a probabilidade da palavra x dada a palavra y
 - com um classificador binário
- Continue ajustando os vetores para **maximizar** essa probabilidade

Word2vec

- Como computar $P(\text{contexto} \mid \text{alvo})$?
 - $P(w_{t+j} \mid w_t)$

$P(w_{t-2} \mid w_t)$	$P(w_{t-1} \mid w_t)$	w_t	$P(w_{t+1} \mid w_t)$	$P(w_{t+2} \mid w_t)$	
transportar	ao	verso	doce	amenas	sensações


Contexto em janela de tamanho 2 alvo Contexto em janela de tamanho 2

Word2vec

- Como computar $P(\text{contexto} \mid \text{alvo})$?
 - $P(w_{t+j} \mid w_t)$

	$P(w_{t-2} \mid w_t)$	$P(w_{t-1} \mid w_t)$	w_t	$P(w_{t+1} \mid w_t)$	$P(w_{t+2} \mid w_t)$
transportar	ao	verso	doce	amenas	sensações



Contexto em janela de
tamanho 2

alvo



Contexto em janela de
tamanho 2

Objetivo (SkipGram)

- Para cada posição $t = 1, \dots, T$, predizer o contexto dentro de uma janela de tamanho m , dada a palavra central w_t

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^T \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} | w_t; \theta)$$

Objetivo

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^T \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} | w_t; \theta)$$

- Objetivo: minimizar log-verossimilhança negativa (média)
- Equivale a maximizar a acurácia preditiva

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$

Objetivo da função

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$

transportar

ao

verso

doce

amenas

sensações

$$J_{t,j}(\theta) = -\log P(\text{transportar} | \text{verso}) = \log \frac{u_{\text{transportar}}^\top v_{\text{verso}}}{\sum_{w \in V} u_w^\top v_{\text{verso}}} = -u_{\text{transportar}}^\top v_{\text{verso}} + \log \sum_{w \in V} u_w^\top v_{\text{verso}}$$

aumenta similaridade entre v_{verso} e $u_{\text{transportar}}$
diminui a similaridade entre todas as outras

Como treinar os vetores?

- Os parâmetros θ são os vetores v_w e u_w para cada palavra no vocabulário
- Tais vetores serão aprendidos a partir de um grande volume de textos, com o objetivo de otimizar a função de custo
- O método de otimização será o gradiente descendente

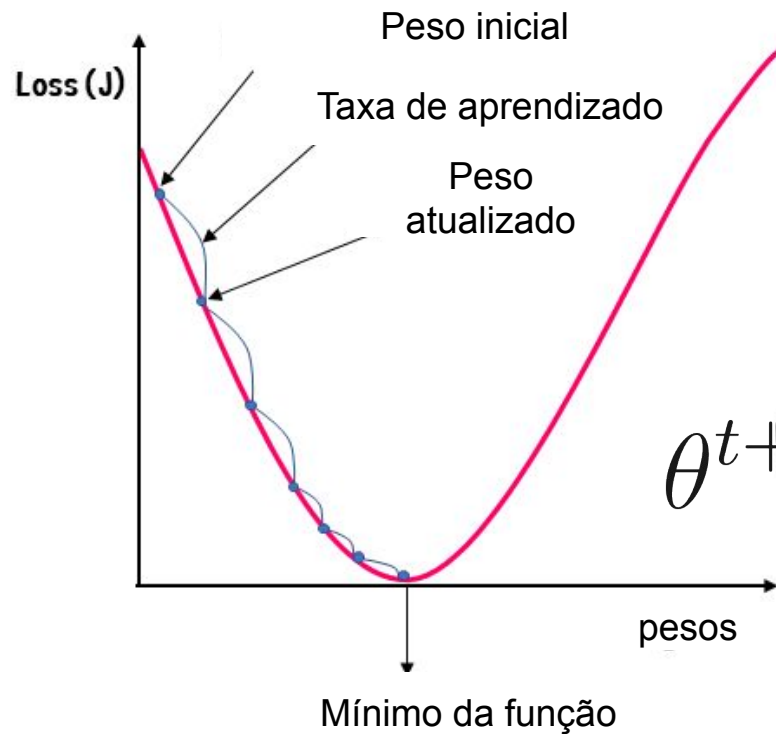
$$\theta^{t+1} = \theta^t - \alpha \nabla_{\theta} J(\theta)$$

Como treinar os vetores?

- Os parâmetros θ são os vetores v_w e u_w para cada palavra no vocabulário
- Tais vetores são aprendidos a partir de um grande conjunto de dados de treinamento, otimizando a função de custo. **Essencialmente, uma rede neural**
- O método de otimização será o gradiente descendente

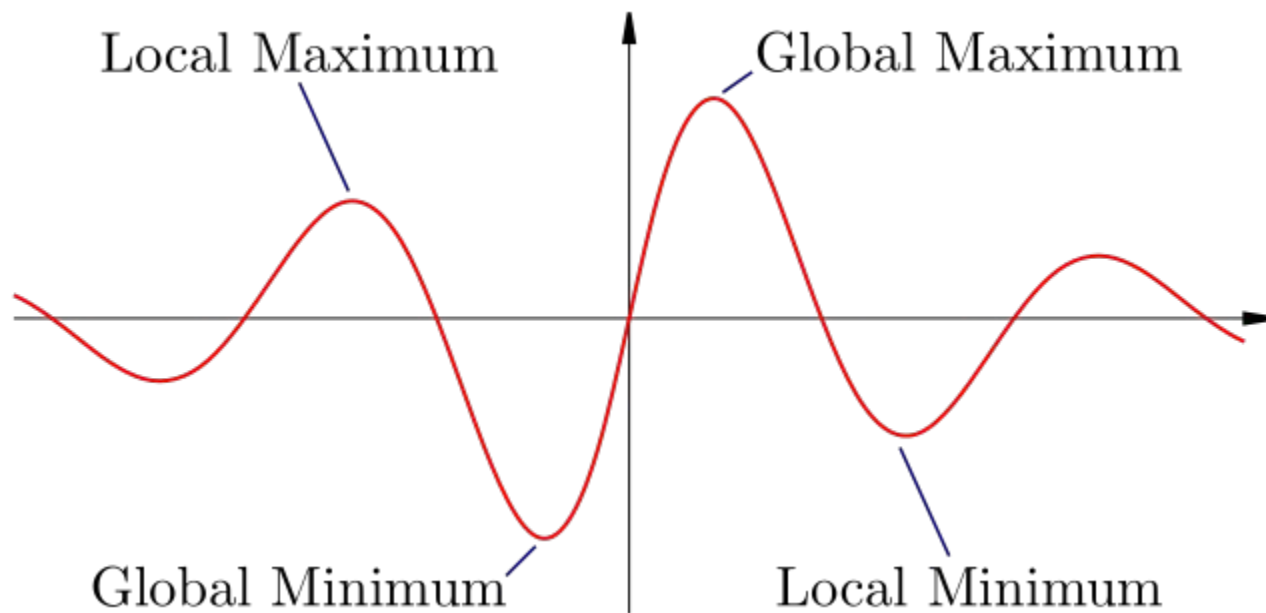
$$\theta^{t+1} = \theta^t - \alpha \nabla_{\theta} J(\theta)$$

Gradiente descendente

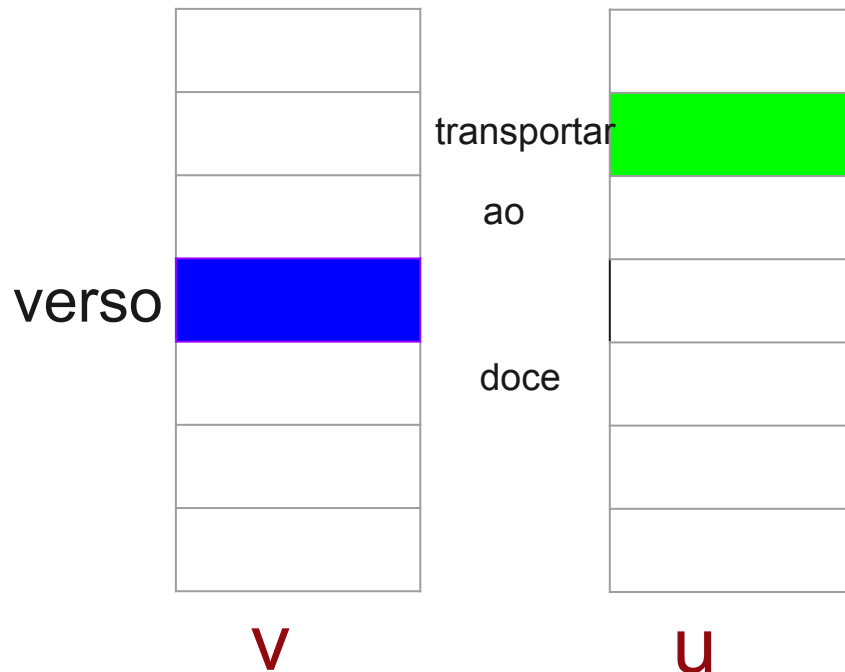


$$\theta^{t+1} = \theta^t - \alpha \nabla_{\theta} J(\theta)$$

Gradiente descendente



Voltando ao word2vec: uma palavra por vez



1 - produto interno de v_{verso} com cada elemento em u

2 - exponencial

3 - soma

4 - computa a loss para esse passo

5 - avalia o gradiente e atualiza v_{verso} e u_w

Amostragem de negativos

- Exemplos positivos : (target, contexto) na janela

- o verso, ao
- o verso, transportar
- o verso, doce
- o verso, amenas

transportar	ao	verso	doce	amenas	sensações
-------------	----	-------	------	--------	-----------

- Exemplos negativos: (target, palavra fora da janela)

- o verso, sensações
- o verso, mais
- o verso, ninho
- o verso, que
- o

Amostragem de negativos

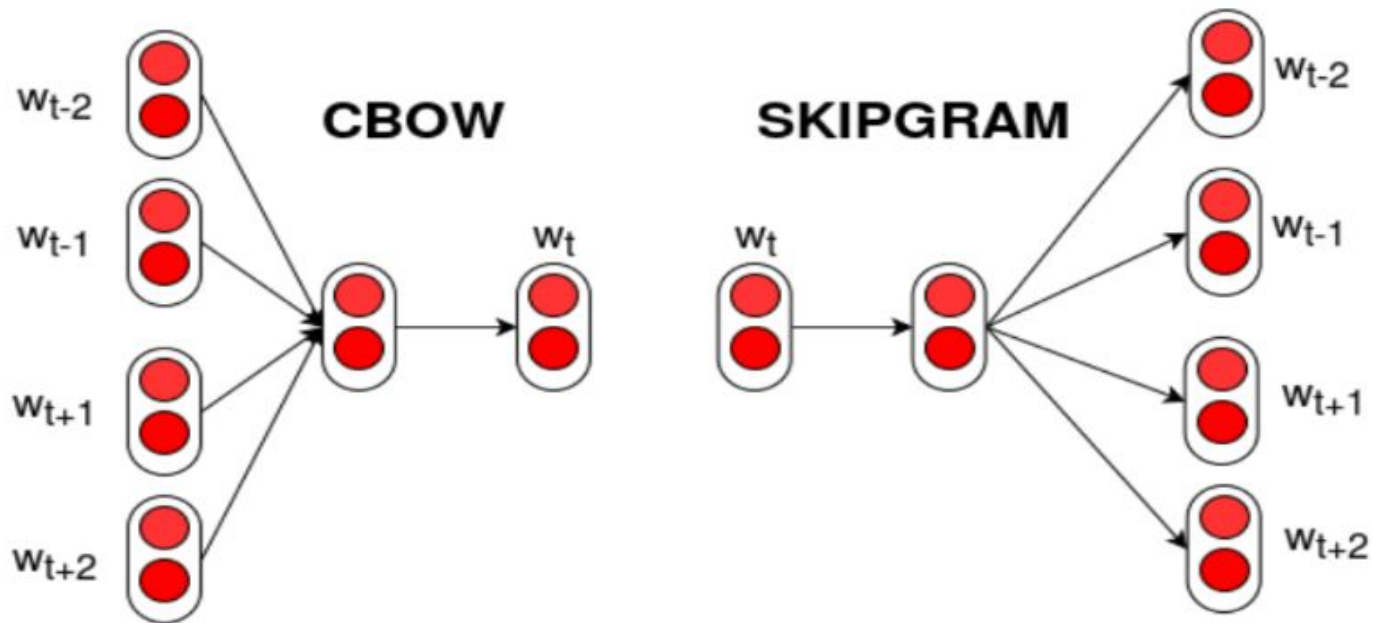
- Fator de normalização é muito custoso

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

Amostragem de negativos

- Exemplos negativos: (target, palavra aleatória)
 - Para cada exemplo positivo, são criados K exemplos negativos
- Palavra aleatória
 - não pode ser a target
 - escolhida de acordo com sua frequência unigrama ponderada
 - Para evitar que uma palavra muito frequente seja muito mais escolhida
 - $P(a) = 0.99; P(b) = 0.01$
 - $P_{0.75}(a) = 0.97; P_{0.75}(b) = 0.03$

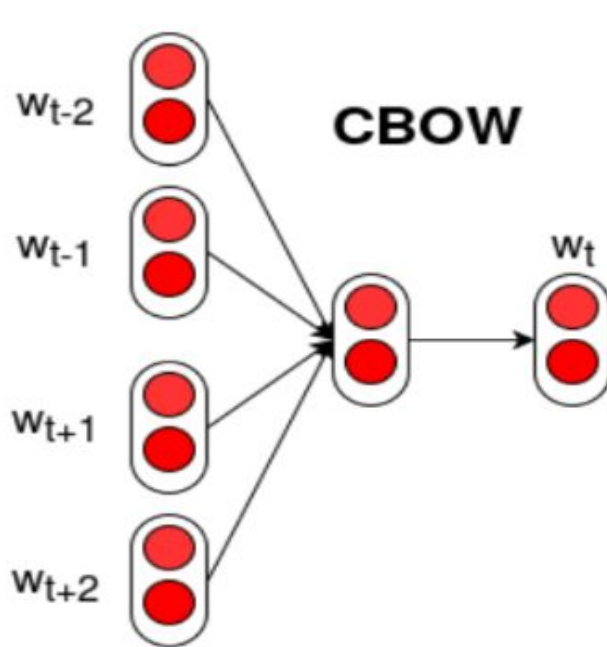
Word2Vec: skipgram e CBOW



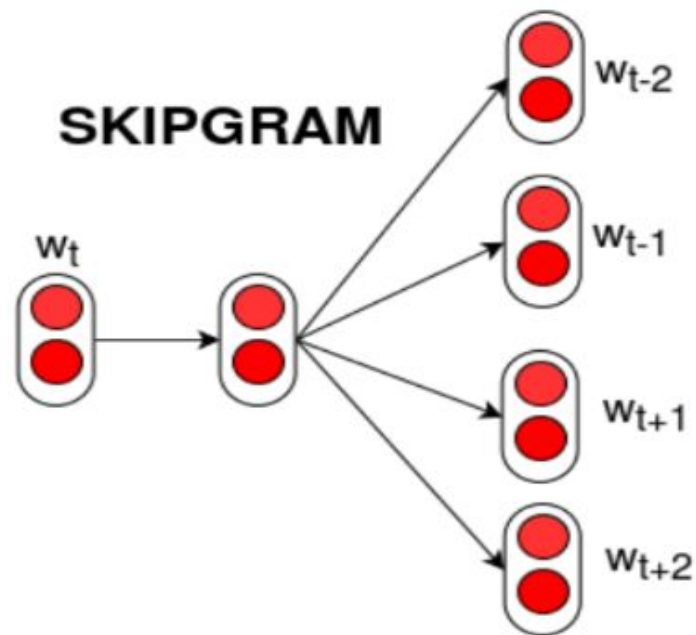
$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_t | w_{t+j})$$

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

Skipgram vs CBOW



$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_t | w_{t+j})$$



$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

Avaliação

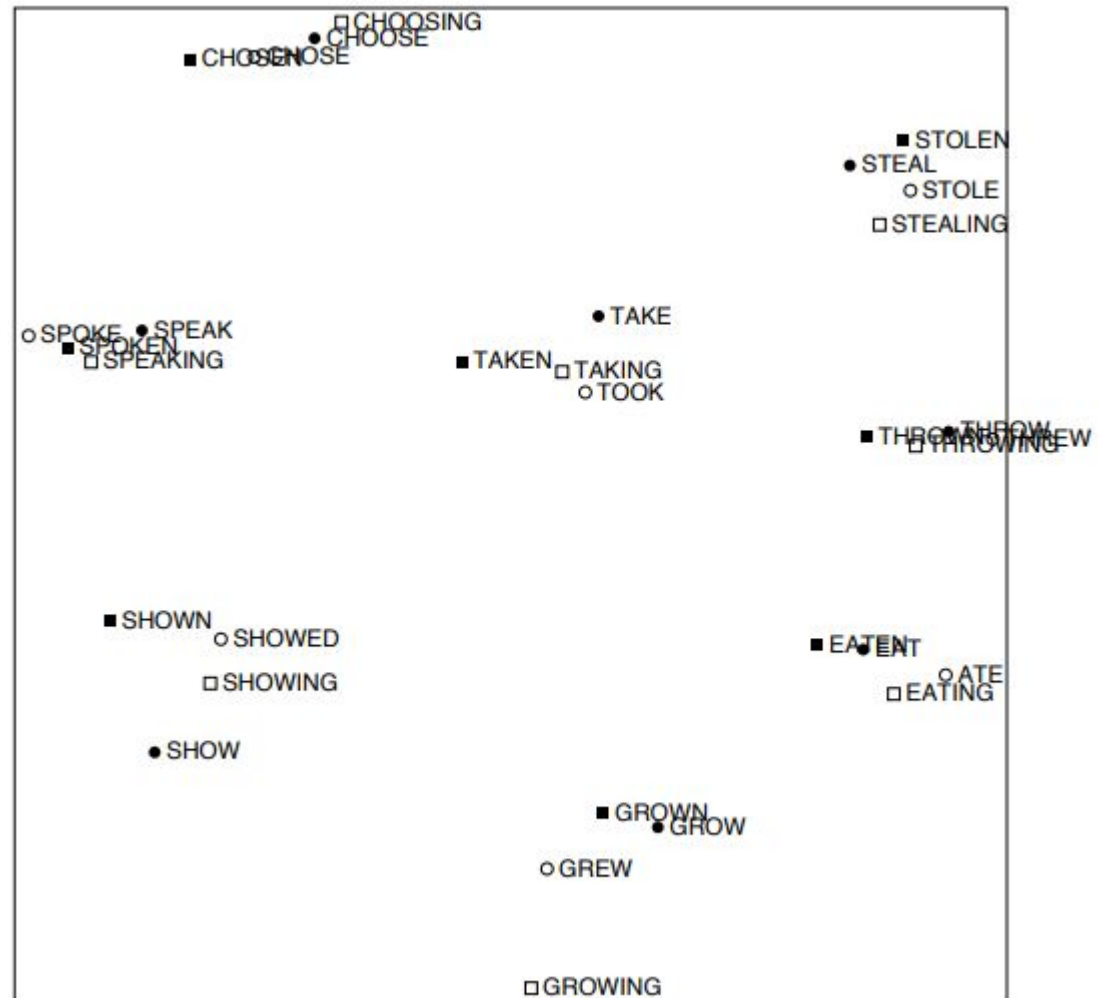
- **Intrínseca**

- Avaliar em uma tarefa intermediária
- Não fica claro o comportamento em uma tarefa real

- **Extrínseca**

- Avaliação em uma tarefa real
- Pode ser complicado de obter métricas rapidamente
- Não fica claro se o problema era da tarefa alvo ou do modelo de linguagem

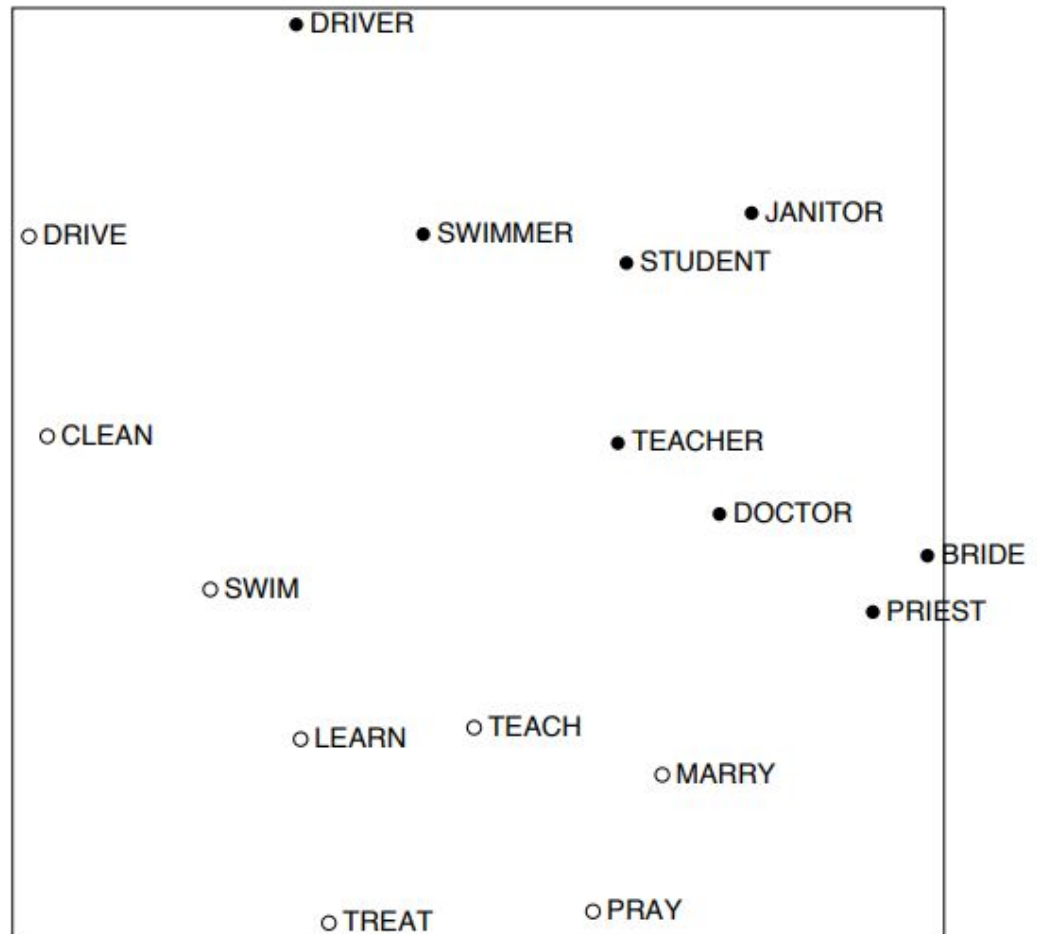
Avaliação intrínseca



An Improved Model of
Semantic Similarity Based on
Lexical Co-Occurrence
Rohde et al. ms., 2005

Avaliação

An Improved Model of
Semantic Similarity Based on
Lexical Co-Occurrence
Rohde et al. ms., 2005

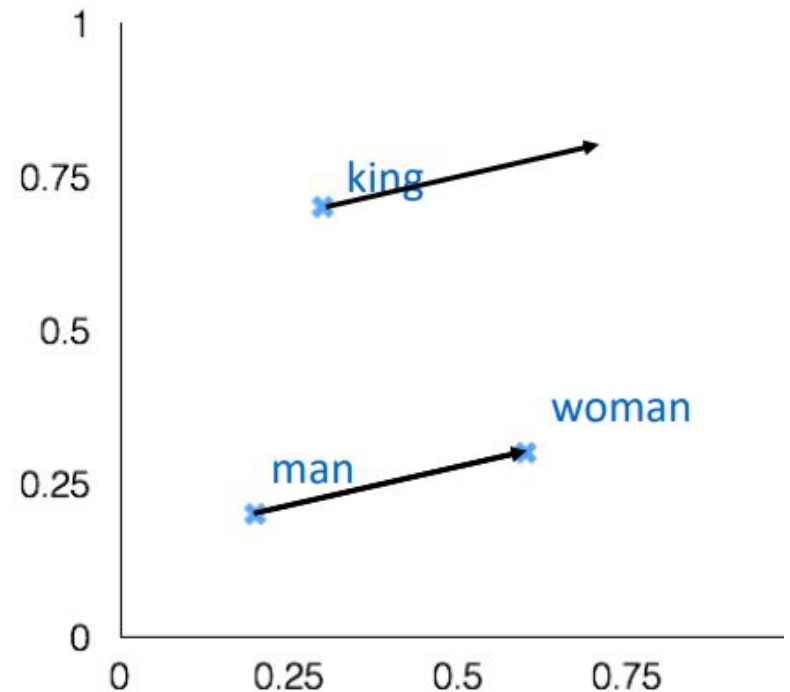


Avaliação

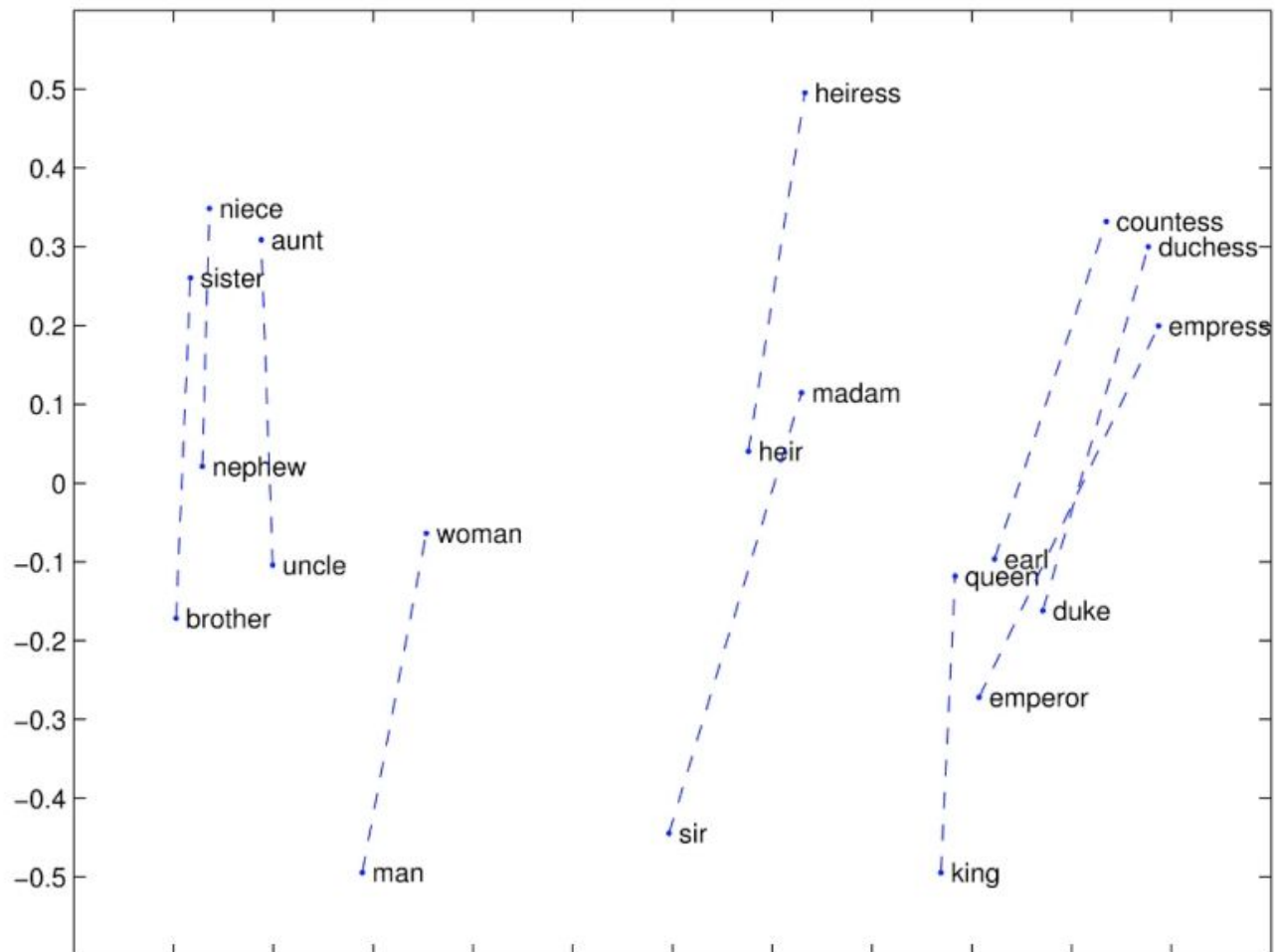
Relações sintáticas e semânticas capturadas com similaridade de cosseno

a:b :: c:?

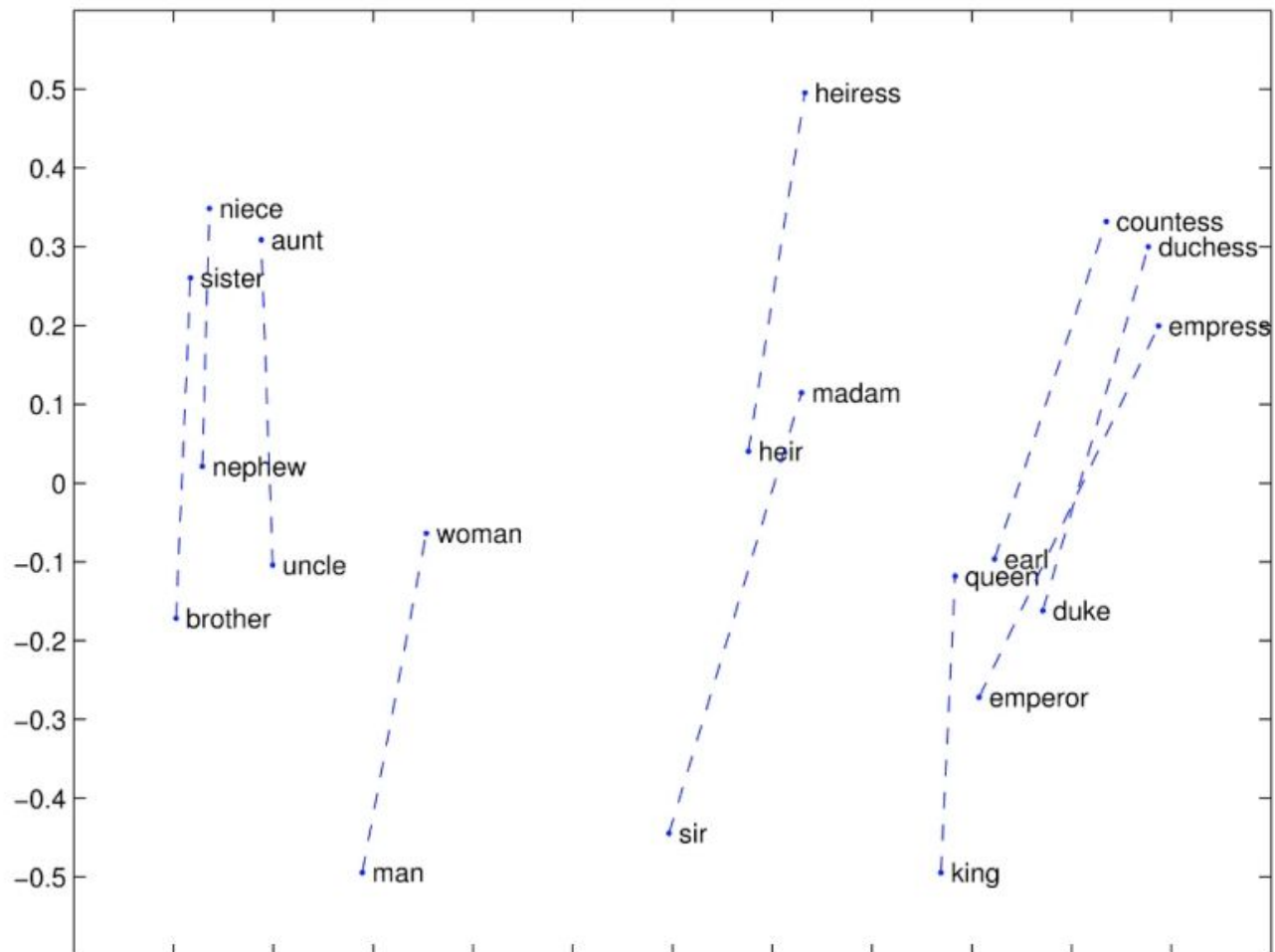
man:woman :: king:?



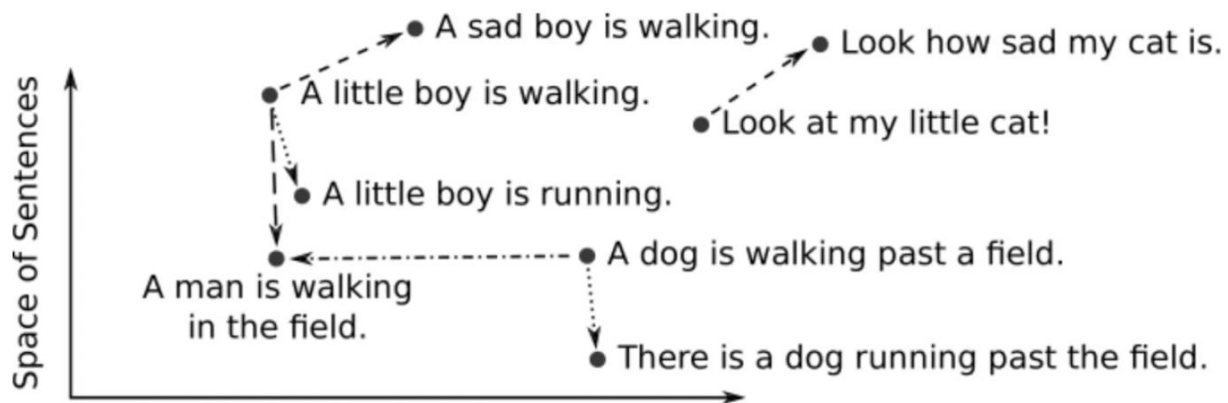
Avaliação



Avaliação - Notebook 1



E as sentenças?

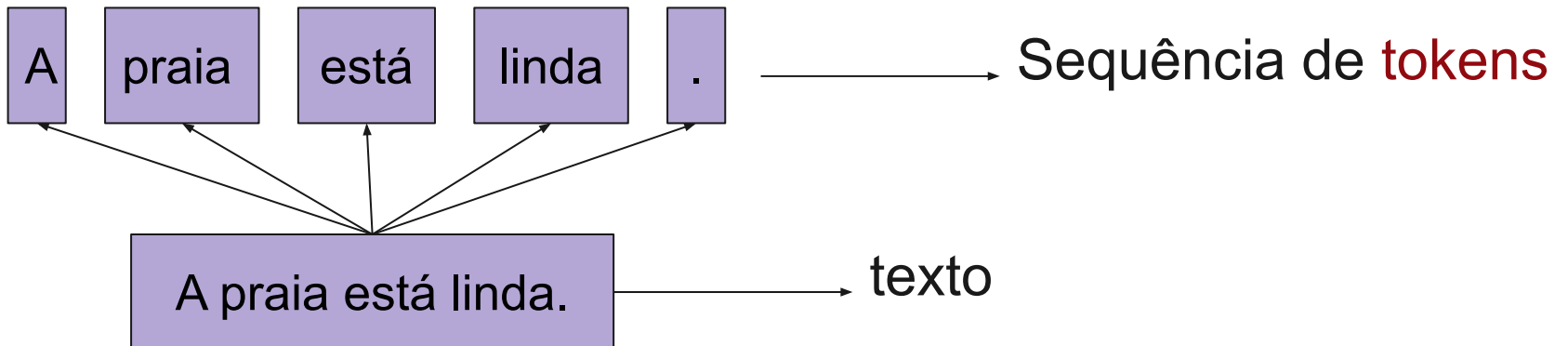


Como resolver uma tarefa (extrínseca)?

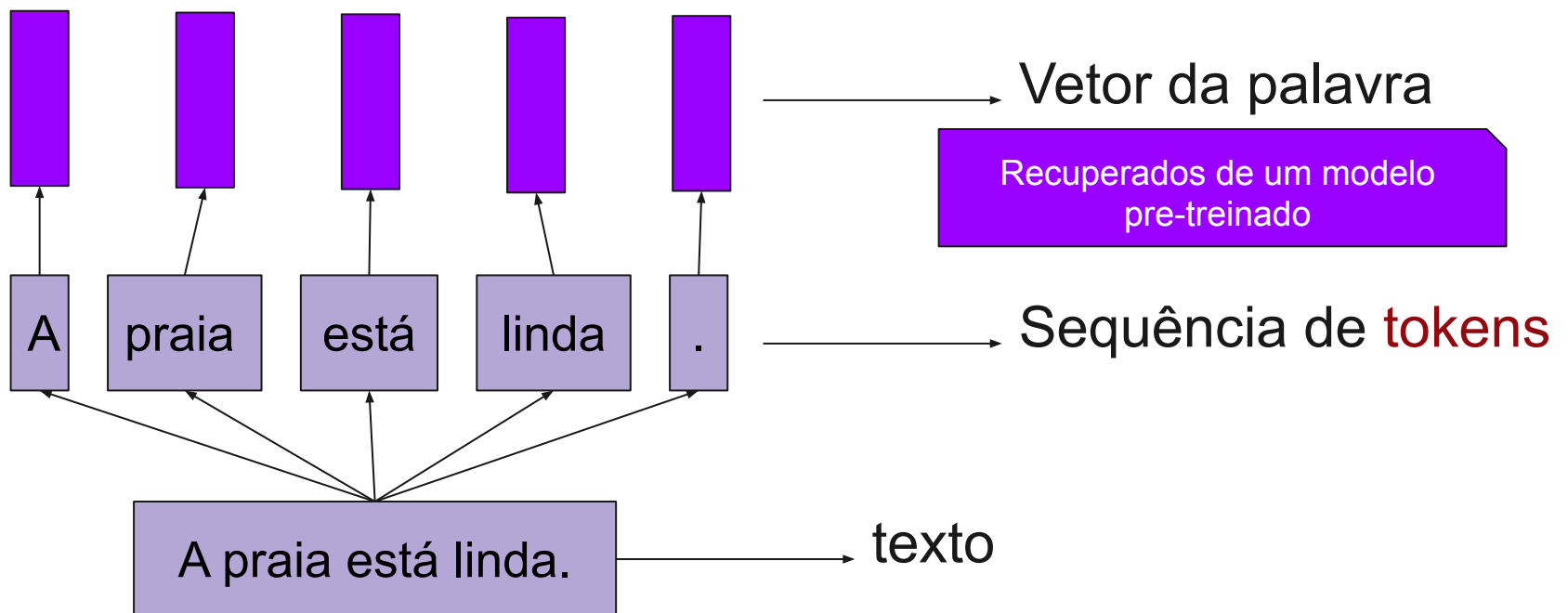
A praia está linda.

texto

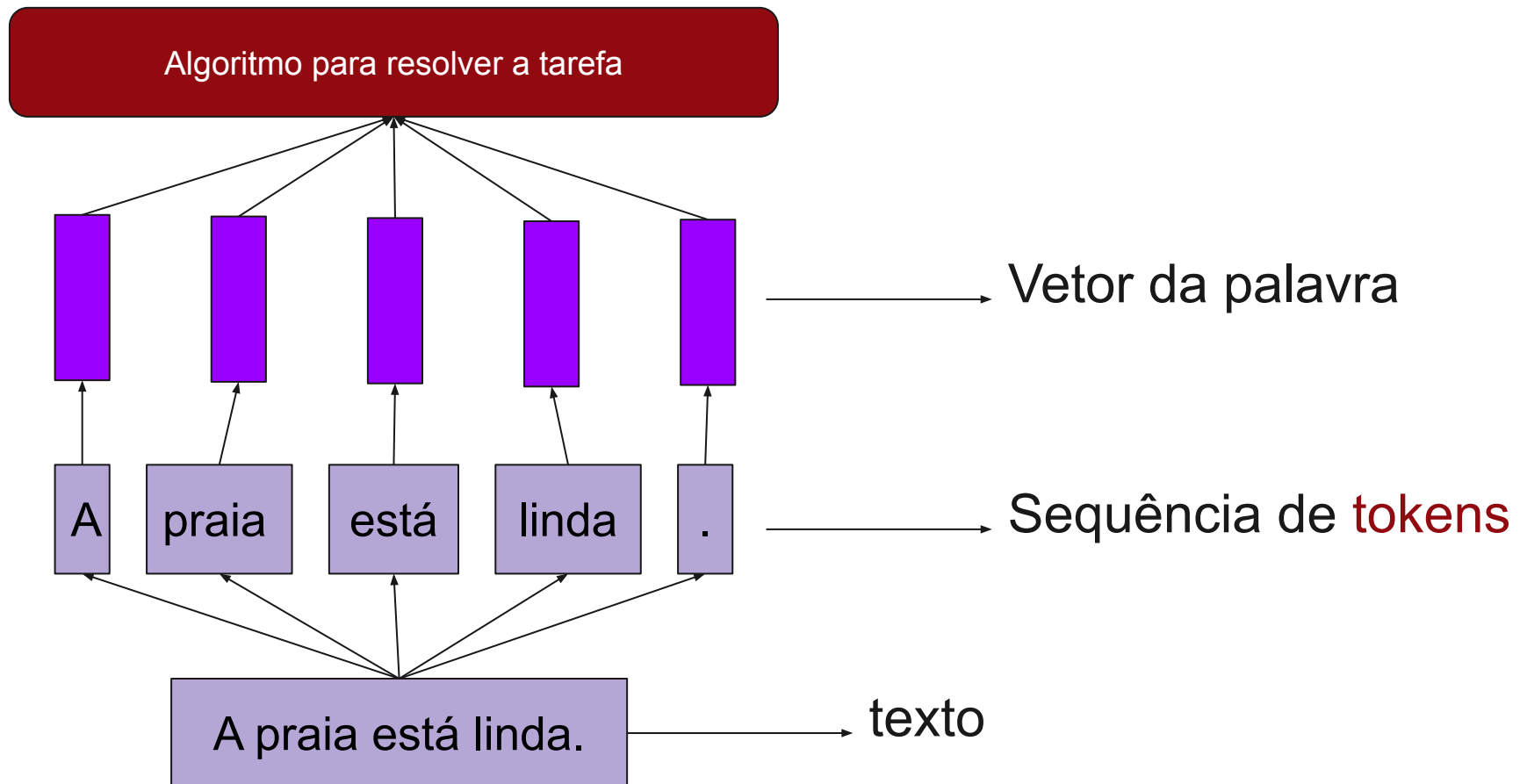
Como resolver uma tarefa?



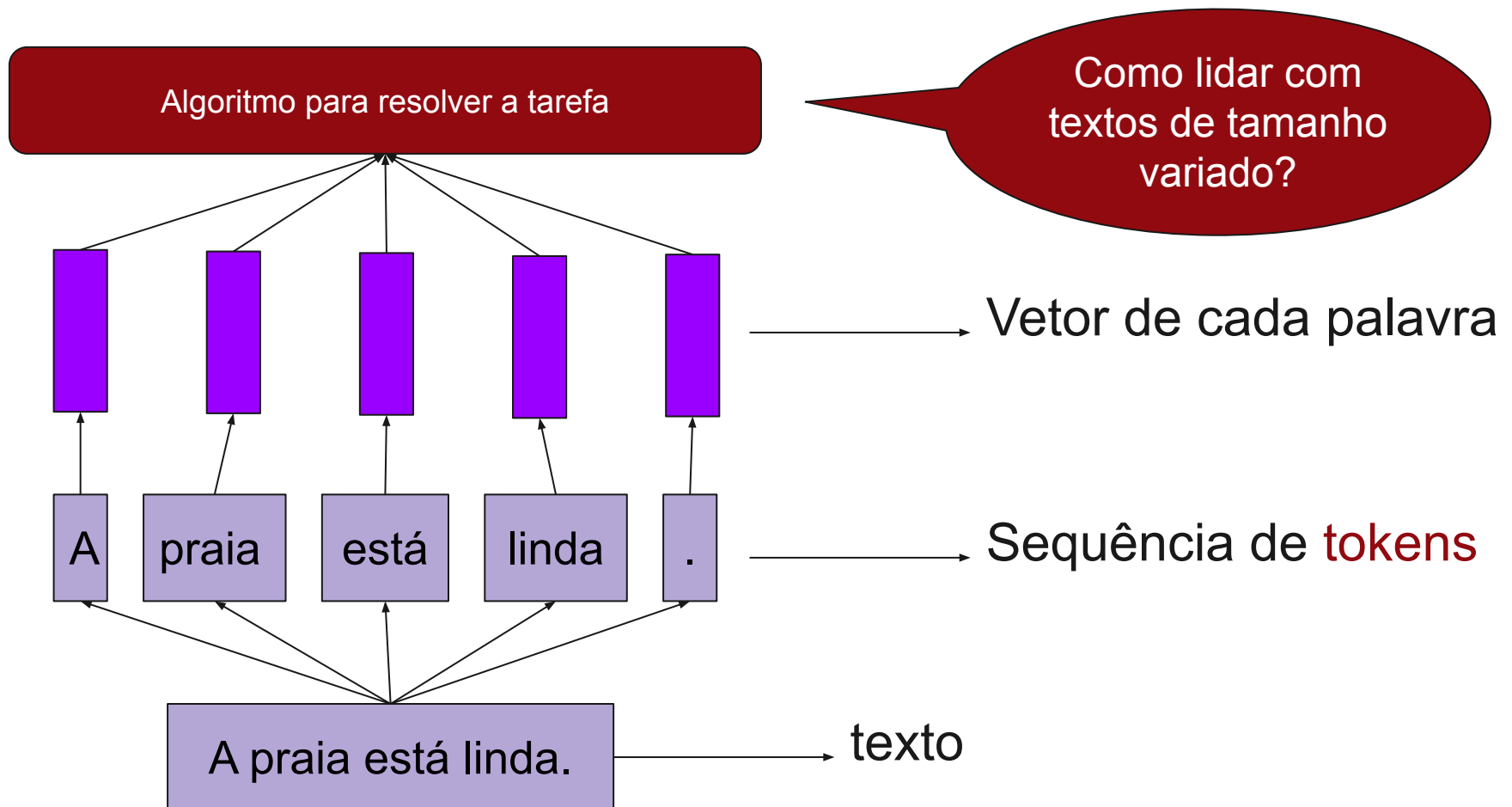
Como resolver uma tarefa?



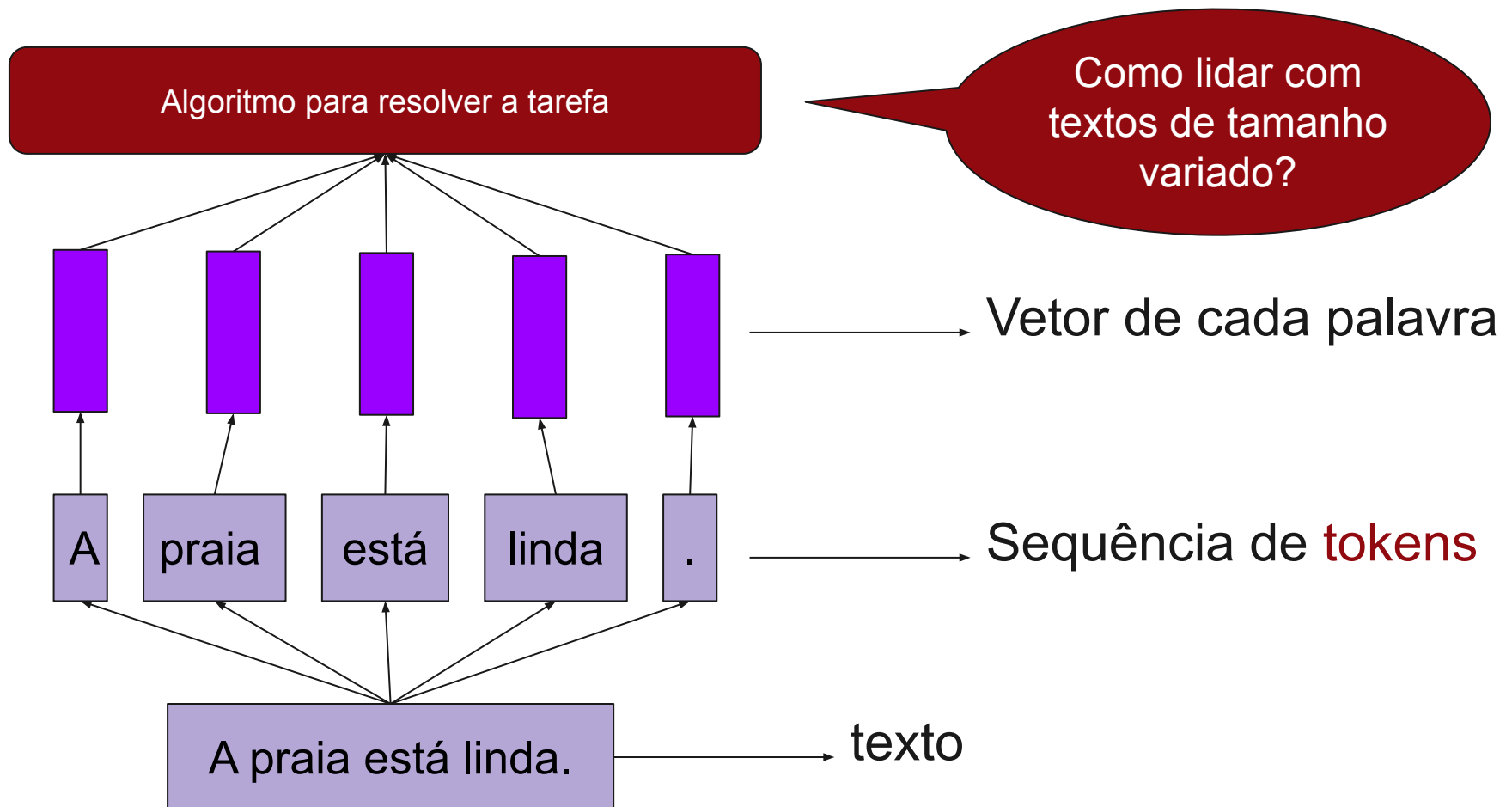
Como resolver uma tarefa?



Como resolver uma tarefa?



Como resolver uma tarefa?



fastText

- Baseado na geração de **subpalavras**
 - Limites são parâmetros
 - Ex. explorando -> <explorando>
 - Suponha limites de 3 a 6

tamanho	Caracteres n-grams
3	<ex, exp, xpl, plo, lor, ora, ran, and, ndo, do>
4	<exp, expl, xplo, plor, lora, oran, rand, ando, ndo>
5	<expl, explo, xplor, plora, loran, orand, rando, ando>
6	<explo, explor, xplora, ploran, lorand, orando, rando>

- Nem todos os embeddings obtidos são aprendidos, define-se um bucket size de tamanho B e cada caracter n-gram cai em um inteiro de 1 a B (usando uma função hash)

GloVe

- Combina as estatísticas globais dos métodos baseados em contagem com a predição de vetores

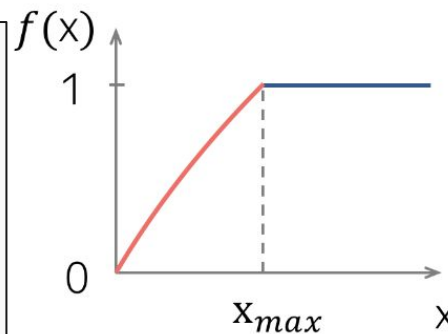
GloVe

Vetor de contexto Vetor da palavra Termos de bias (também aprendidos)

$$J(\theta) = \sum_{w,c \in V} f(N(w, c)) \cdot (u_c^T v_w + b_c + \bar{b}_w - \log N(w, c))^2$$

Função que:

Penaliza eventos raros
Não dá muito peso para
palavras frequentes



$$\begin{cases} (x/x_{max})^\alpha & \text{if } x < x_{max}, \\ 1 & \text{otherwise.} \end{cases}$$

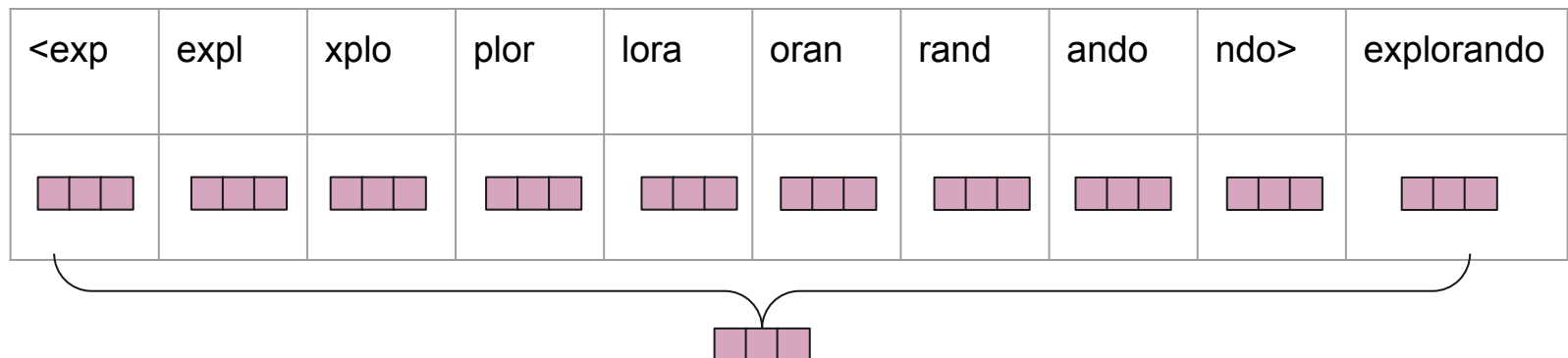
$$\alpha = 0.75, x_{max} = 100$$

Skipgram com negative sampling (fastText)

- Considera-se contexto e palavra alvo

Eu	estou	explorando	o	espaço
----	-------	------------	---	--------

- O embedding da palavra alvo é calculado a partir da soma dos vetores dos caracteres n-grams e a palavra inteira



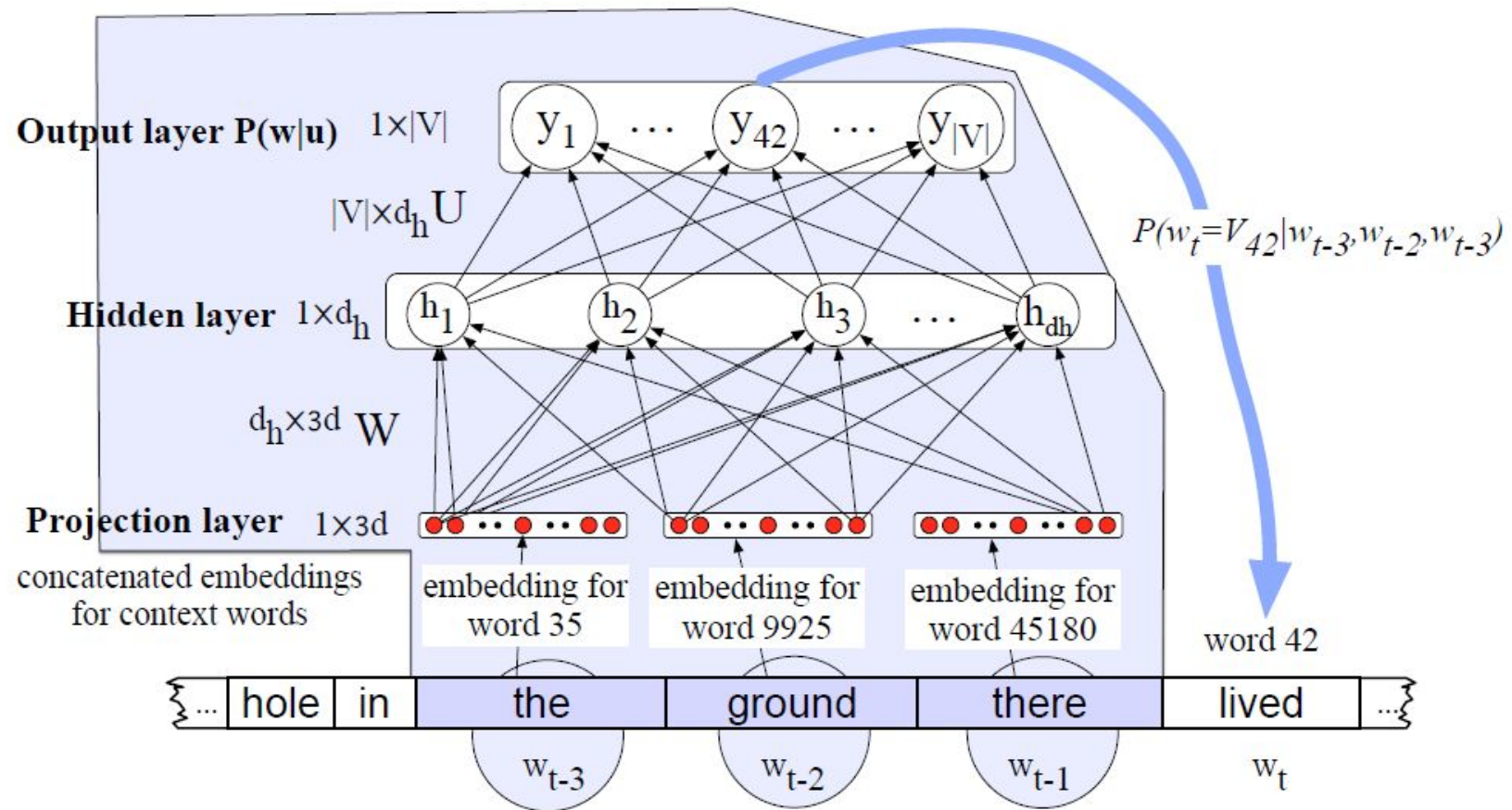
Skipgram com negative sampling (fastText)

- Para as palavras de contexto, consideramos o **embedding da palavra completa** apenas
- Para os exemplos negativos, é o mesmo processo do Skipgram com **negative sampling**
- Calcula-se o produto interno dos embeddings da palavra de contexto e alvo, e aplica-se sigmóide
- Atualiza os embeddings

Modelos de linguagem neurais

- O **contexto** é representado pelos **embeddings** das palavras
 - Palavras não vistas, com embeddings similares podem ser a próxima palavra
 - Modelo de linguagem probabilístico torna isso mais difícil
 - *Por favor, certifique-se de alimentar o XXX*
 - Suponha que 'gato' não tenha aparecido no treinamento, apenas 'cão'
 - PLM vai sugerir apenas 'gato'
 - NLM pode sugerir ambos (embeddings similares)

Neural Language Model (embeddings pré-treinados)



Neural Language Model (embeddings treinados)

