

Processamento de Linguagem Natural e Processamento de Imagens

Aula 3

James Moraes de Almeida



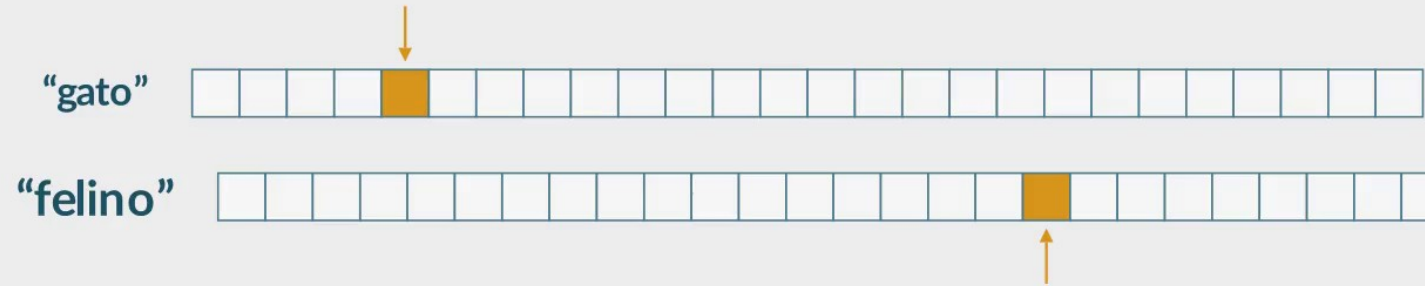
Conteúdo do curso

- Qual o plano?
 - Processamento de linguagem natural (PNL ou NLP do inglês).
 - ~~Mineração de dados.~~
 - ~~Histórico.~~
 - ~~Tokenização.~~
 - ~~Pré-processamento.~~
 - ~~Similaridade de sentenças.~~
 - ~~Bag-Of-Words.~~
 - ~~TF-IDF.~~
 - **Word embeddings.**
 - **Modelagem de tópicos.**
 - Grandes Modelos de Linguagem para extração de dados.
- Material principal: <https://www.youtube.com/@futuremojo>

Objetivos desta aula

- Ao final, você deve saber:
 - Explicar por que BoW e TF-IDF não capturam significado.
 - Interpretar a similaridade de cosseno entre dois vetores.
 - Descrever o treino do Word2Vec com negative sampling.
 - Usar vetores pré-treinados: similaridade, analogias, vizinhos.
 - Reconhecer as limitações dos embeddings estáticos.
 - Aplicar embeddings de sentença a um corpus real de abstracts.

Por que TF-IDF não enxerga significado



$|V| \approx 3\,000\,000$ colunas — uma por palavra do vocabulário

$$\cos(\text{gato}, \text{felino}) = 0,00$$

colunas diferentes → nenhuma coordenada em comum

Você conhece uma palavra pelas companhias que ela mantém

Ontem tomei um copo de ■■■■■ gelado.

O ■■■■■ na geladeira já venceu.

candidatos

suco

leite

vinho

~~café~~

café não vai na geladeira “vencido”

Word Embeddings

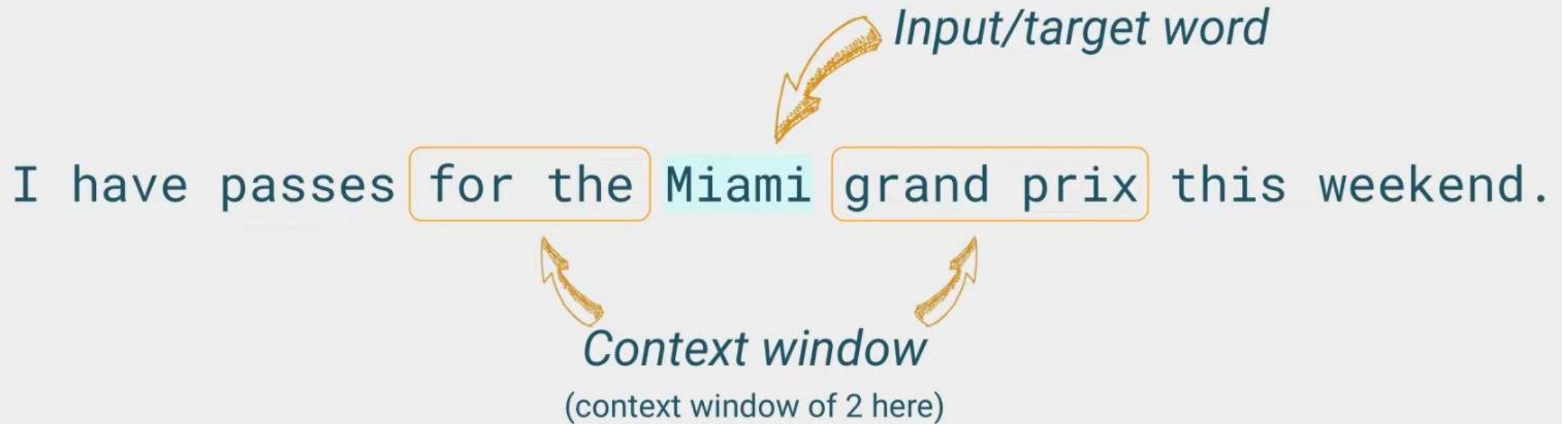
- Como representar as palavras dependendo do contexto?
- Bag of Words e TF-IDF são literais.
- Mas podemos vetorizar as palavras de maneira diferente.
- Vamos utilizar a sentença abaixo:

I have passes for the Miami grand prix this weekend.

Sentence from a corpus

Word Embeddings

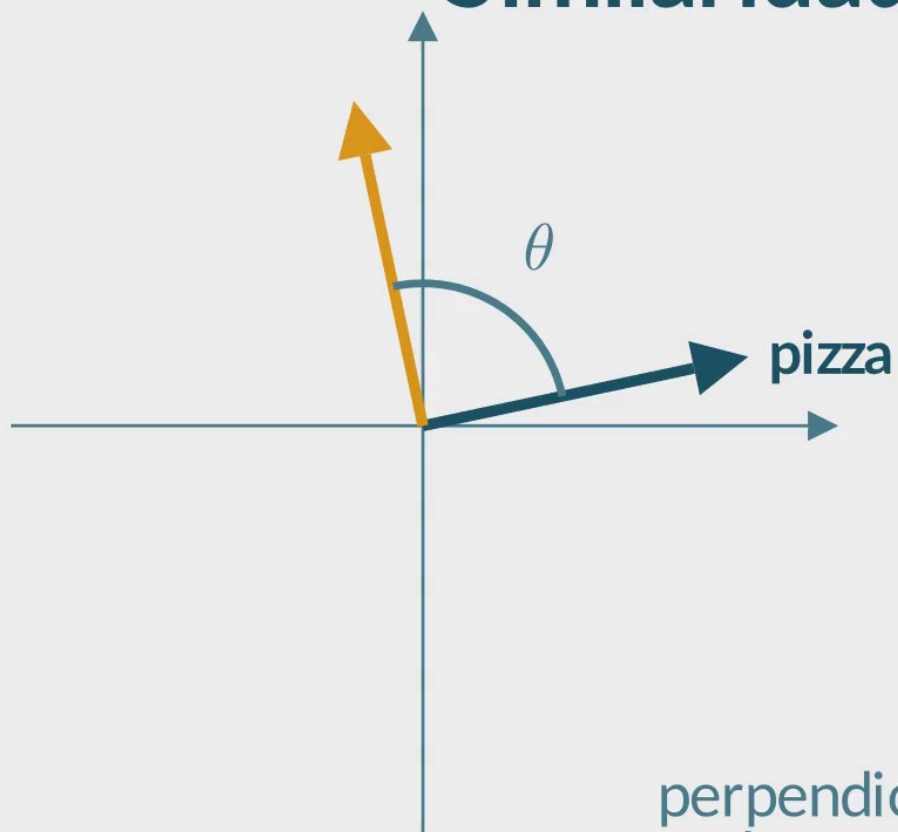
- Contexto!



O que é um word embedding

- Um vetor denso de dimensão fixa, aprendido a partir de texto.
 - Denso: quase toda coordenada é diferente de zero.
 - Dimensão fixa: 300, independente do tamanho do vocabulário.
 - Aprendido: ninguém escolheu o que cada coordenada significa.
- Palavras de significado próximo ficam próximas no espaço.
 - É essa proximidade que vamos medir e explorar.

Similaridade de cosseno



$$\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

só a direção importa –
o tamanho é normalizado

$$\cos(\theta) = +0.00$$

perpendiculares $\rightarrow \sim 0$
nada em comum

A janela de contexto gera os pares de treino

I have passes **for the Miami grand prix** this weekend

 palavra-alvo  contexto (± 2)

pares (alvo, contexto)

(Miami, for)	(Miami, the)
(Miami, grand)	(Miami, prix)

O conjunto de treino sai sozinho da frase

I have passes **for the** Miami **grand prix** this weekend

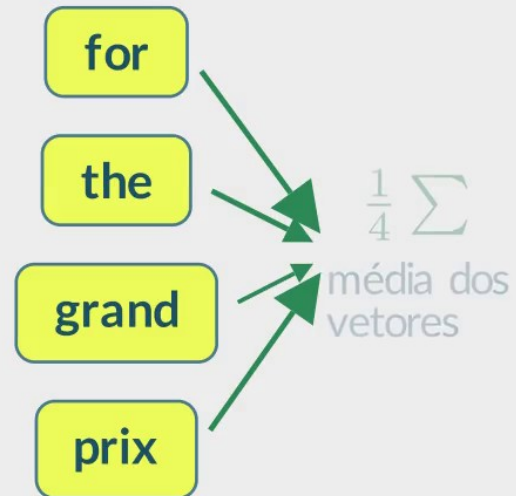
Exemplos positivos

(for, Miami)
(the, Miami)
(grand, Miami)
(prix, Miami)

Exemplos negativos

e os negativos? sorteados ao acaso no vocabulário inteiro — 3 000 000 de palavras

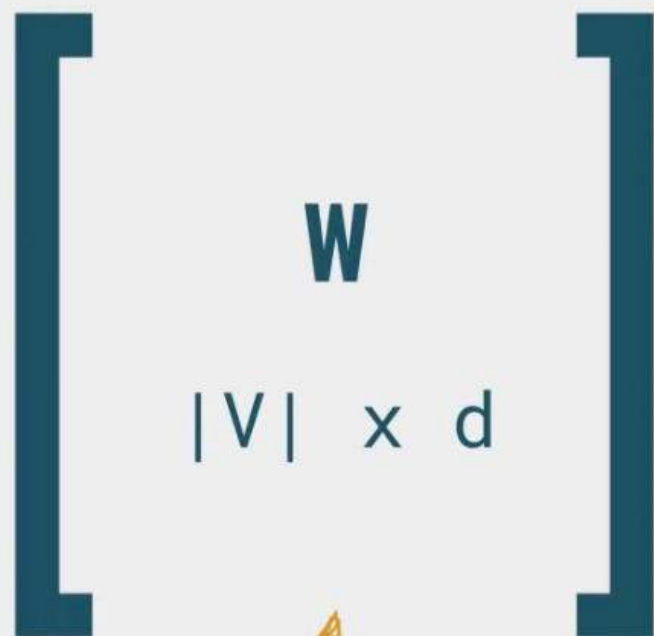
Skip-Gram e CBOW: a mesma janela, tarefas invertidas



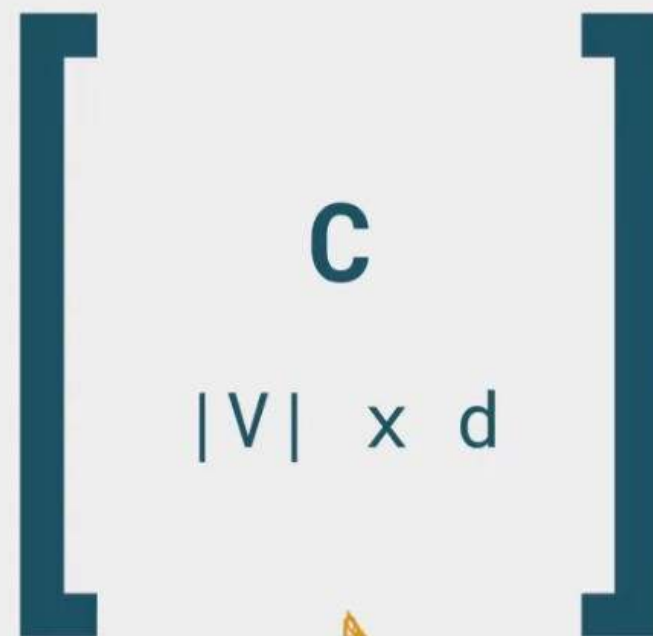
CBOW: do contexto para a palavra

Word Embeddings

Create two matrices (initialized randomly)



Target word embeddings



Context word embeddings



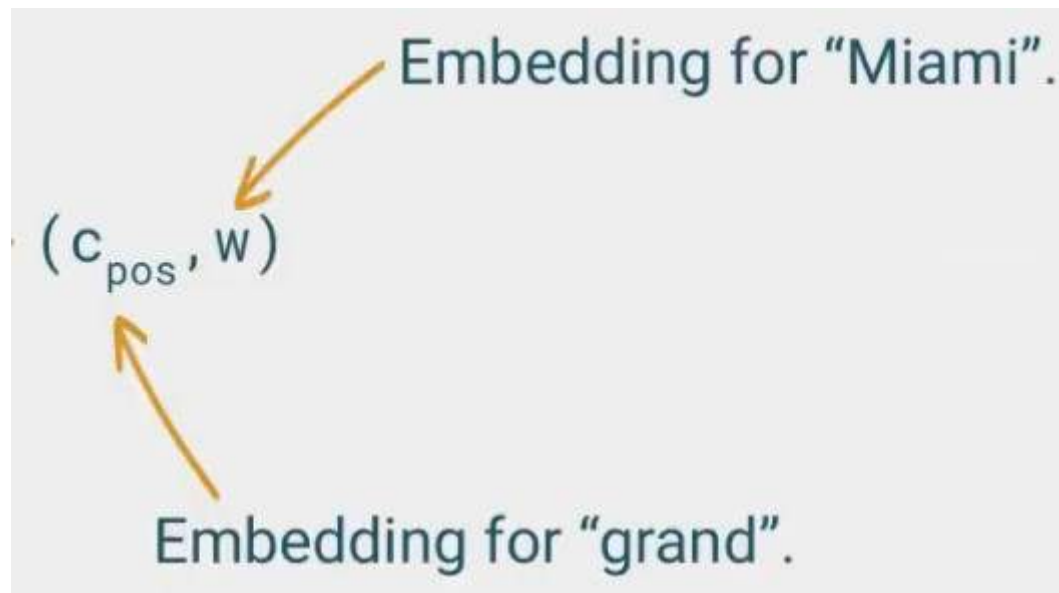
A 10,000-word vocabulary and a goal to generate 300-dimension embeddings would mean **W** and **C** will each be $10,000 \times 300$.

Word Embeddings

- Agora o exemplo positivo virou um par de vetores.

Positive Samples

(for, Miami)
 (the, Miami)
 (grand, Miami)
 (prix, Miami)
 ...



Word Embeddings

- A seguir, escolhemos k exemplos negativos.
- k é um hiperparâmetro.

Negative Samples

(mountain, Miami)

(elephant, Miami)

(rice, Miami)

(prix, Miami)

...

← Sample k negative samples

Word Embeddings

- A seguir, escolhemos k exemplos negativos.
- k é um hiperparâmetro.

Negative Samples

$(\text{mountain, Miami}) \Rightarrow (c_{\text{neg-1}}, w)$
 $(\text{elephant, Miami}) \Rightarrow (c_{\text{neg-2}}, w)$
 $(\text{rice, Miami}) \Rightarrow (c_{\text{neg-3}}, w)$
 (prix, Miami)
 $\dots$

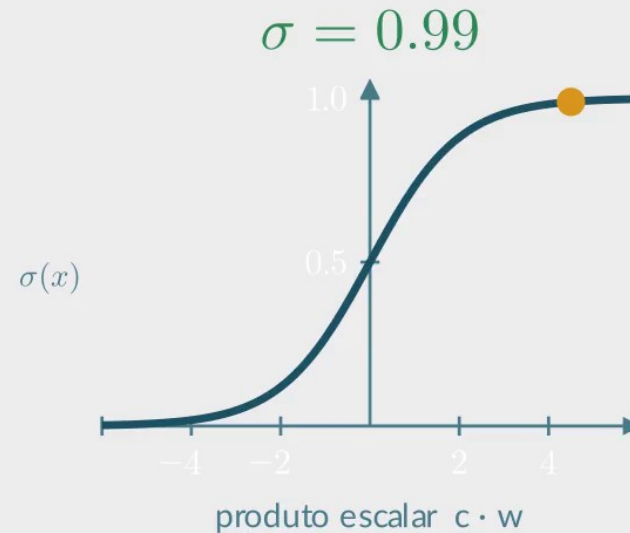
Word Embeddings

- Dadas duas palavras (w, c), é provável delas ocorrerem próximas no texto?
 - Probabilidade de ocorrer:
 - Probabilidade de não ocorrer:

Word Embeddings

- Dadas duas palavras (w, c) , é provável delas ocorrerem próximas no texto?
- Função de perda:
- Para atualizar os embeddings:

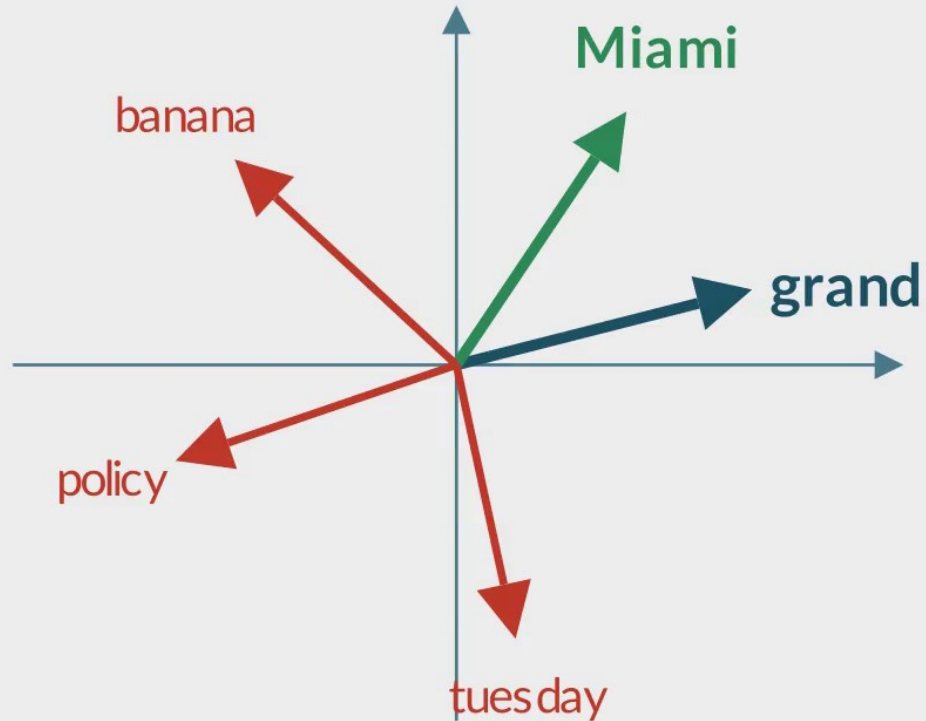
Do produto escalar à probabilidade



$$P(+ \mid c_{pos}, w) = \sigma(c_{pr}$$

escalar bem positivo $\rightarrow$ probabilidade perto de 1

O que o gradiente faz com os vetores



$$\frac{\partial L}{\partial c_{pos}} = [\sigma(c_{pos} \cdot w) - 1] w$$

negativo → puxa para junto

$$\frac{\partial L}{\partial c_{neg}} = [\sigma(c_{neg} \cdot w)] w$$

positivo → empurra para longe

uma atualização

Word Embeddings

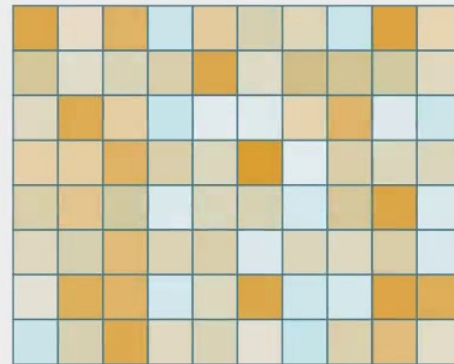
- Treino completo.
- Podemos utilizar somente a **W**, ou somar/média com a **C**.

After training is complete...

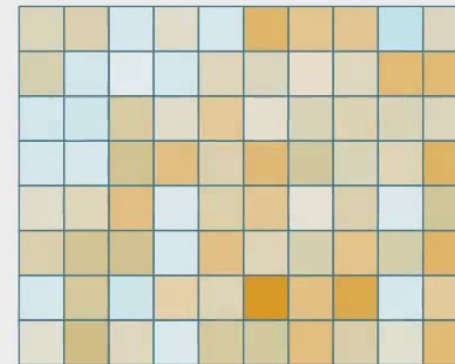
$$\begin{bmatrix} \mathbf{W} \\ |V| \times d \end{bmatrix} + \begin{bmatrix} \mathbf{C} \\ |V| \times d \end{bmatrix}$$

Cada palavra tem dois vetores, não um

W
papel de alvo



C
papel de contexto



milhões de pares depois...

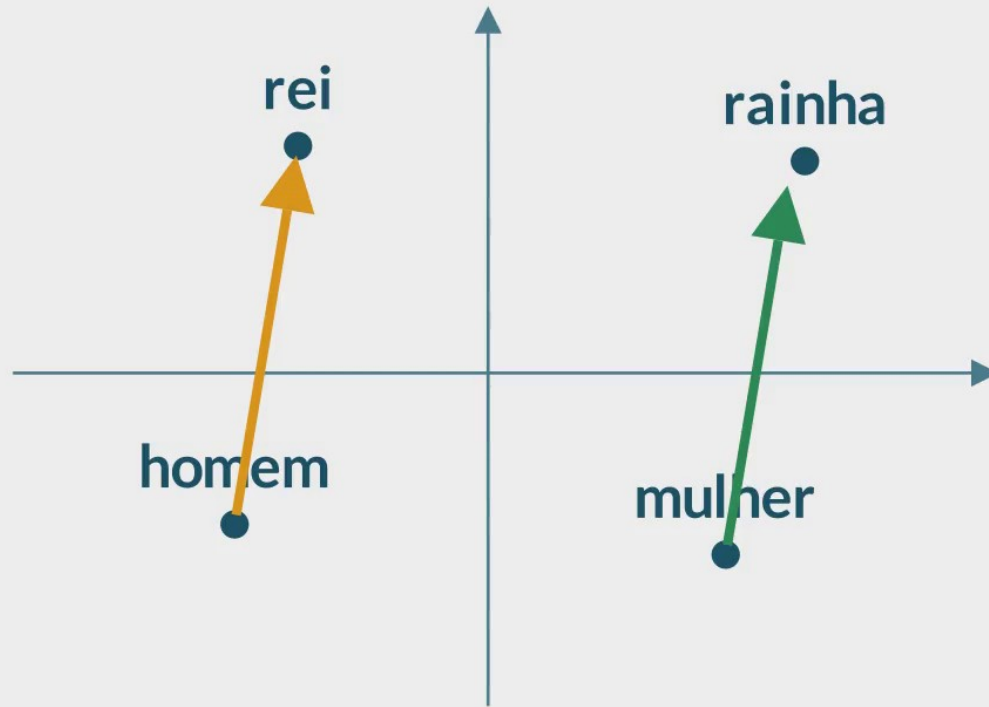
Os vetores do Google News

- Word2Vec treinado pelo Google sobre ~100 bilhões de palavras.
 - 3 milhões de palavras e expressões.
 - 300 dimensões por vetor.
 - Arquitetura Skip-Gram com negative sampling.
- No notebook carregamos só as 200 000 mais frequentes.
 - O arquivo completo tem ~1,6 GB — limit=200000 economiza memória.
 - As palavras mais frequentes são as mais bem treinadas.

O que dá para fazer com os vetores

- `wv.similarity(a, b)` — cosseno entre duas palavras.
- `wv.most_similar(p)` — os vizinhos mais próximos.
 - Aceita `positive=` e `negative=` : é assim que se fazem analogias.
- `wv.doesnt_match([...])` — qual palavra destoia do grupo.
- `wv.n_similarity(l1, l2)` — similaridade entre dois conjuntos.
 - Útil para comparar documentos curtos pela média dos vetores.

Analogias são aritmética no espaço dos vetores



$$\vec{rei} - \vec{homem}$$

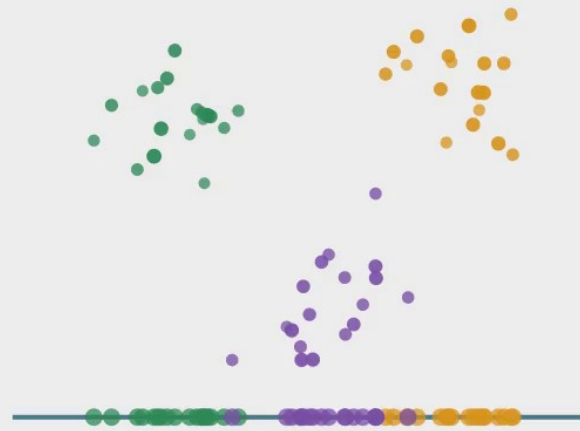
o deslocamento que
“dá realeza” a uma palavra

$$+ \vec{mulher}$$

o mesmo deslocamento,
aplicado em outro ponto

Todo gráfico de embeddings já é uma projeção

300 dimensões



projetar = achatar contra um plano

Limitação 2 – a ordem das palavras se perde

dog bites man

man bites dog

média dos vetores das palavras



$$\cos = 1,00$$

As mesmas palavras, em ordem invertida, produzem o mesmo vetor — e os significados são opostos.

Exercícios — primeira metade

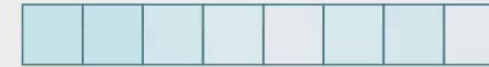
- Encontre três pares de palavras com cosseno $> 0,7$.
- Monte uma analogia que funcione e uma que falhe.
 - Por que a segunda falhou? O que faltava no corpus de treino?
- Use `doesn't_match` com um grupo da sua área.
- Procure uma palavra OOV e trate o `KeyError`.
- Encontre um viés no modelo (profissão, gênero, nacionalidade).
 - De onde ele veio? O modelo inventou ou aprendeu?

Um vetor por palavra – ou um vetor por ocorrência?

Sentei no **banco** da praça.

Abri uma conta no **banco**.

Word2Vec



um vetor a meio caminho entre
“assento” e “instituição financeira”
— e portanto bom para nenhum dos dois

as duas ocorrências caem no mesmo vetor

Do vetor de palavra ao vetor de sentença

- Um documento inteiro vira um único vetor.
 - Ingênuo: média dos vetores das palavras — perde a ordem.
 - Sentence Transformers: um modelo treinado para isso.
- all-MiniLM-L6-v2: 384 dimensões, rápido e leve (~90 MB).
 - `model.encode(lista_de_textos)` devolve uma matriz $N \times 384$.
- Dois abstracts próximos no espaço tratam de temas próximos.

Estudo de caso: ligas de alta entropia

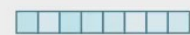
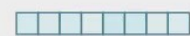
- Corpus: abstracts exportados da Web of Science.
- O que vamos fazer com eles:
 - Codificar cada abstract num vetor de 384 dimensões.
 - Buscar por sentido, não por palavra-chave.
 - Agrupar por tema com K-Means, sem rótulo nenhum.
 - Visualizar em 2D com t-SNE.

Busca semântica: procurar por sentido, não por palavra

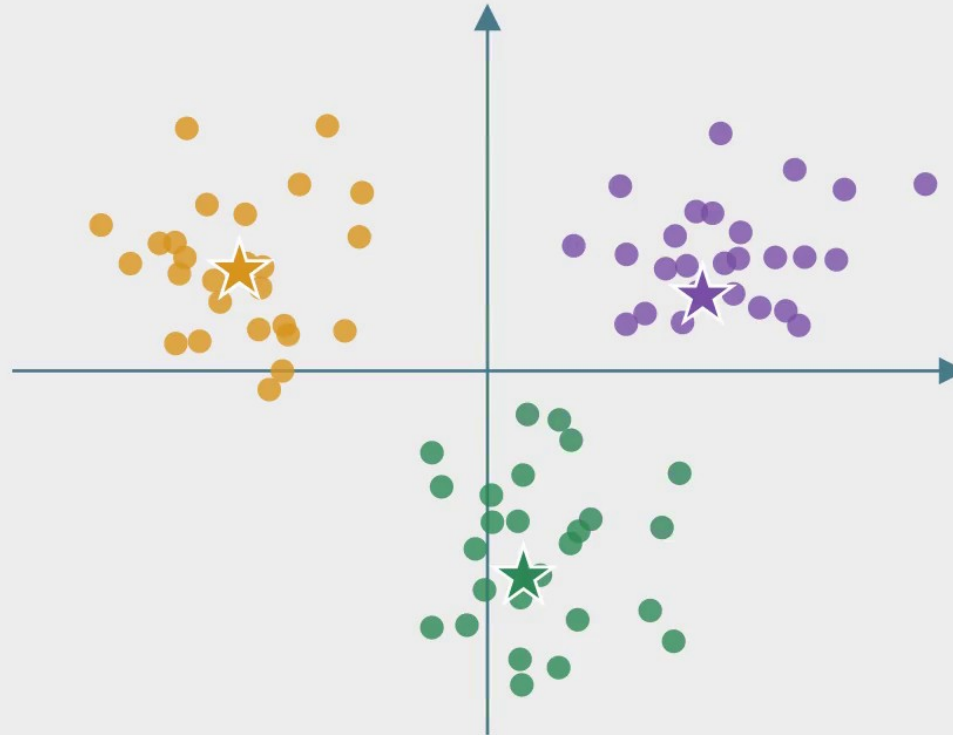
consulta

“resistência mecânica em altas temperaturas”



-  Creep behaviour of refractory HEAs at 1000 °C
-  Corrosion resistance in saline environments
-  Yield strength retention above 900 °C
-  Magnetic properties of CoCrFeNi
-  Thermal stability of nanostructured alloys
-  X-ray diffraction of as-cast samples

K-Means: atribuir, recalcular, repetir



iteração 3 • cada ponto vai para o centroide mais próximo

Word Embeddings

- Como são utilizados os word embeddings?
 - Usualmente, utilizamos um pré-treinado.
 - É possível fazer um ajuste fino de um pré-treinado.
 - Também é possível treinar do zero.
- Explicação mais detalhada:
 - <https://www.youtube.com/watch?v=lebL0RQF5lg>
 - Também faz o tutorial que passarei o Notebook para vocês.
 - Porém modifiquei consideravelmente.

Exercícios — segunda metade e referências

- Faça três buscas semânticas e compare com busca por palavra.
 - Ache um caso em que a busca semântica ganha claramente.
- Rode o K-Means com k diferente do cotovelo. O que muda?
- Leia os 10 abstracts mais próximos de um centroide.
 - Consegue dar um nome ao grupo?
- Referências: NLP Demystified (futuremojo), módulo Word Vectors.
 - Jurafsky & Martin, Speech and Language Processing, cap. 6.

OBRIGADO

James Moraes de Almeida
james.almeida@ilum.cnpem.br

ilum.cnpem.br