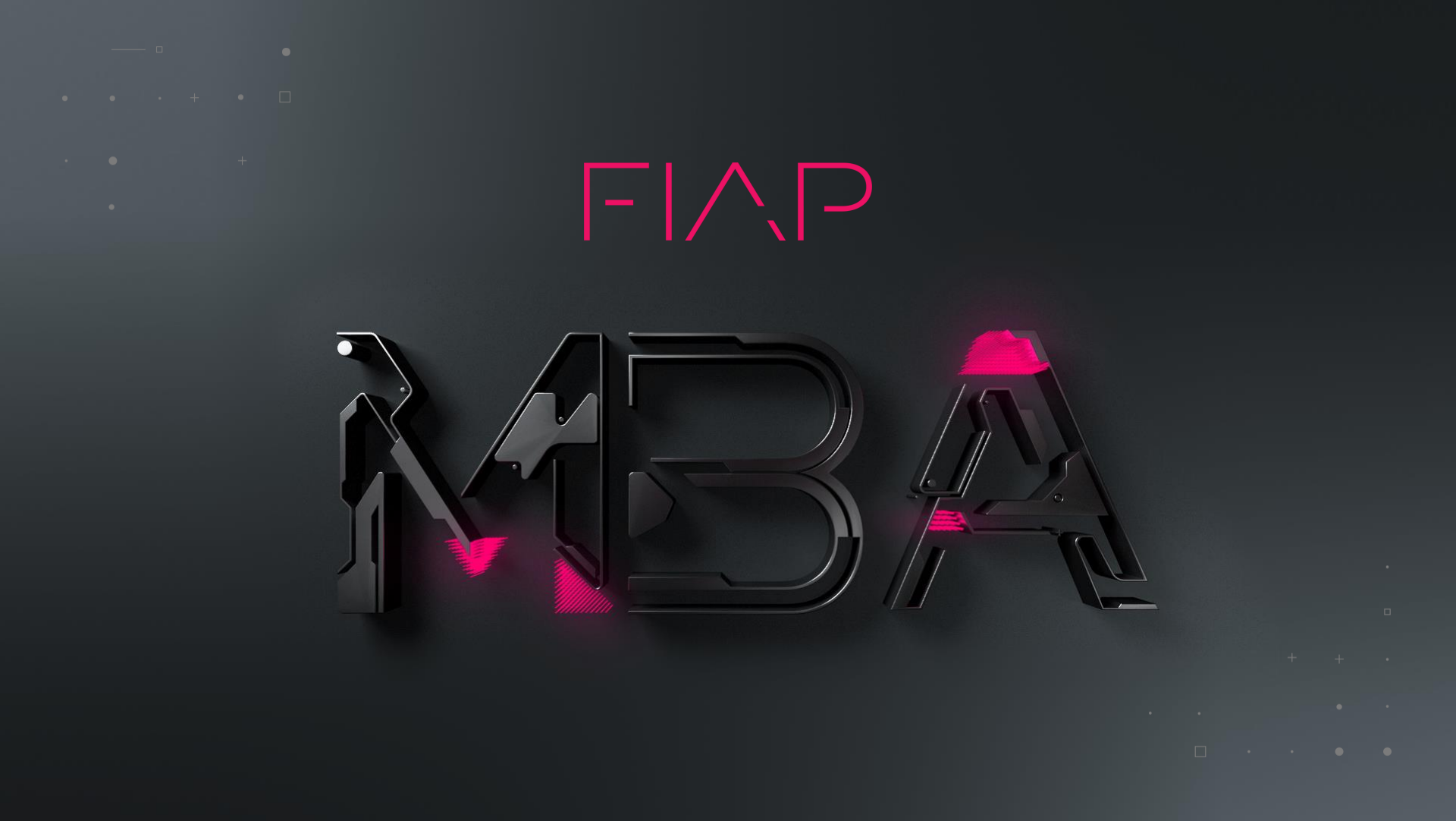

FIAP
MBA

Embeddings

Prof. Anderson Dourado

Agenda

1. **Embeddings**
2. **Word2Vec**
3. **Demo Word2Vec**
4. **Exercício**

Embeddings

Representações de textos

- Até o momento tratamos palavras como símbolos atômicos: *hotel, conferência, andar...*
- Ignoramos o contexto e a ordem das palavras.
- Consideremos a relevância das palavras pela frequência ou representação de 0´s e 1.

- Chamamos essa representação de *one-hot vector*
- Exemplo:
  - Hotel: [0 0 0 0 0 0 1 0 0 0 0]
  - Motel: [0 0 0 0 1 0 0 0 0 0 0]

Principais problemas nas representações utilizadas até agora:

- Alta dimensionalidade dos dados
- Falta de representação semântica
- Representação esparsa
- Falta de contexto

As palavras não possuem qualquer tipo de correlação!

A maioria das representações de texto são discretas e possuem alguns problemas:

- Perda de nuances: sinônimos – *apto, bom, expert, proficiente.*
- Perde novas palavras (impossível de manter atualizado): *fodão*, *ninja...*
- Subjetivo (não leva em consideração contexto)
- Necessita trabalho humano para criar e adaptar
- Difícil calcular similaridade de palavras

O que gostaríamos?

- **Analise semântica**, onde o contexto importa.
- Representação que permita uma **comparação de textos**, a partir de um cálculo simples de distâncias por exemplo.
- **Representação compacta**, de forma a melhorar a performance dos métodos de ML.
- Representação que **aprenda os diferentes significados** que uma palavra pode ter:

"Sentado no **banco** da praça vi um assalto ao **banco**."

Para entendermos o conceito dessas representações, também chamadas de Representações Distribuídas, precisamos entender alguns pontos antes, premissas:

- **Similaridade Distribucional:** o significado de uma palavra pode ser entendido a partir do contexto em que aparece.
  Isto é conhecido também como conotação, ou seja, o significado é definido pelo contexto. Diferente de denotação, que é o significado literal de uma palavra.

- **Hipótese Distribucional:** Palavras que aparecem em **contextos similares possuem significados similares** então duas palavras que aparecem em contextos similares devem possuir vetores similares.

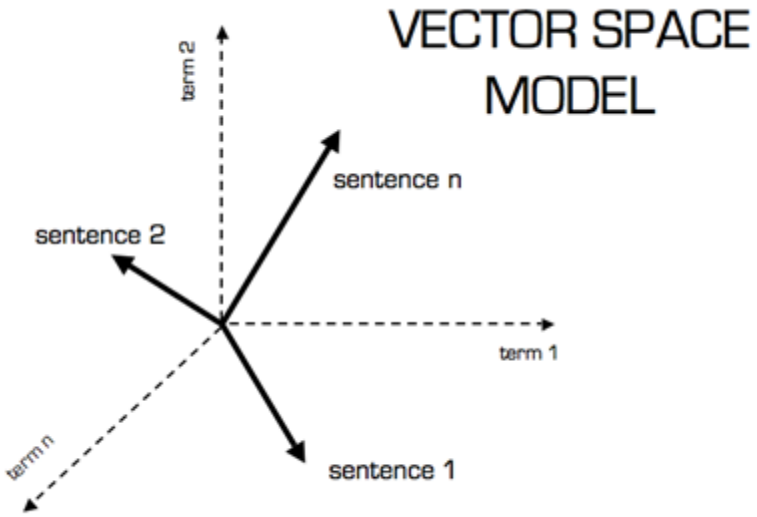
- **Representação Distribuída**: representação de texto através de vetores compactos (baixa dimensão) e densos (não-esparsos). Daqui, surgiu o conceito de *Word Embeddings*.

Ideia 1: Definir sentido pela distribuição linguística

Ideia 2: Sentido como um ponto **multidimensional no espaço**

Definir sentindo como um ponto do espaço baseado na distribuição

- Cada palavra é um vetor (não apenas “bom” ou “w_{45}”)
- Palavras similares são vizinhas no espaço semântico
- Construímos o espaço observando as palavras vizinhas no texto



Por que **vetores**?

- Com palavras:
  - A feature é uma palavra
  - Necessita exatamente a mesma palavra

- Com embeddings:
  - A feature é um vetor
  - Podemos generalizar para uma palavra não vista

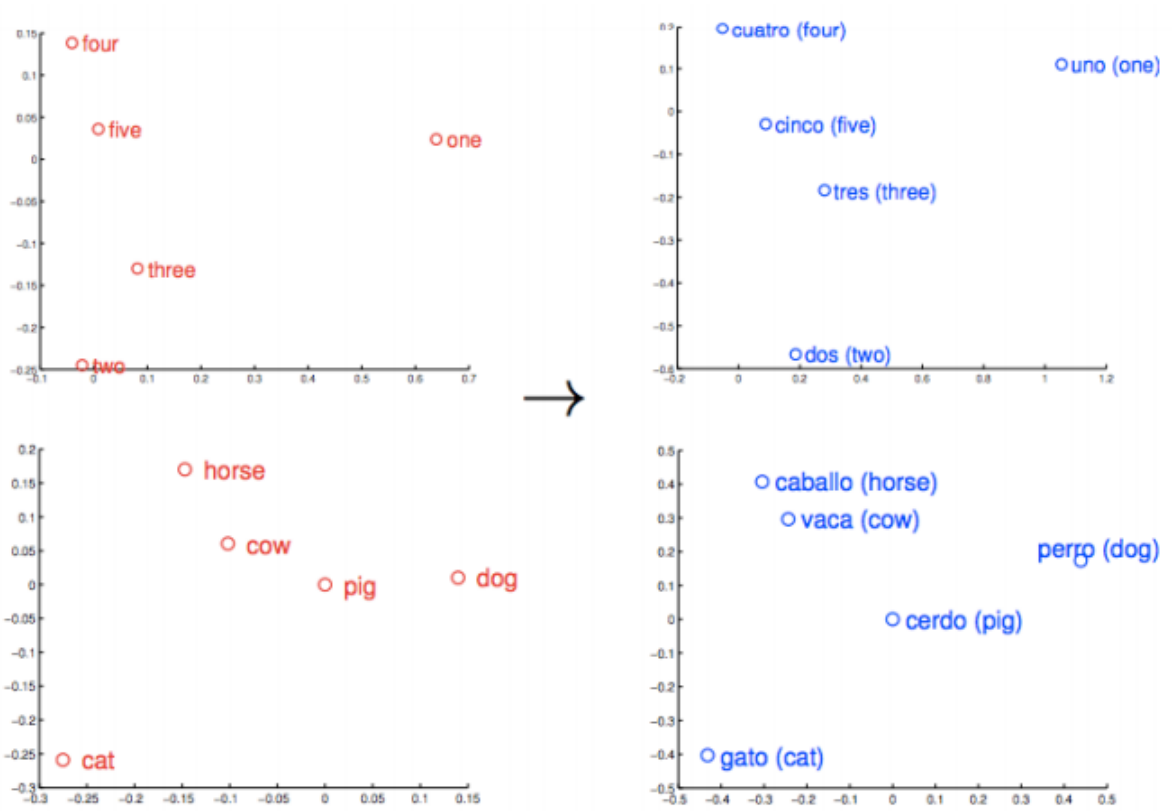
Word Embeddings - Conceitos e definições

Word Embedding é conjunto de modelos para mineração de textos, ou seja, é mais uma **técnica de pré-processamento** em NLP, onde os textos são transformados e as **palavras representadas por um vetor na forma numérica**, ou seja, em uma representação matemática de cada palavra.

Word Embeddings utilizam representações de **vetores densos de tamanho fixo** que são capazes de armazenar informações sobre o contexto e significado dos documentos.

Cada palavra é representada por um ponto em um espaço multidimensional (**embedding space**) e como falamos, cada palavra é representada de forma numérica no vetor, que na verdade são os pontos/dimensões de cada palavra.

Word Embeddings



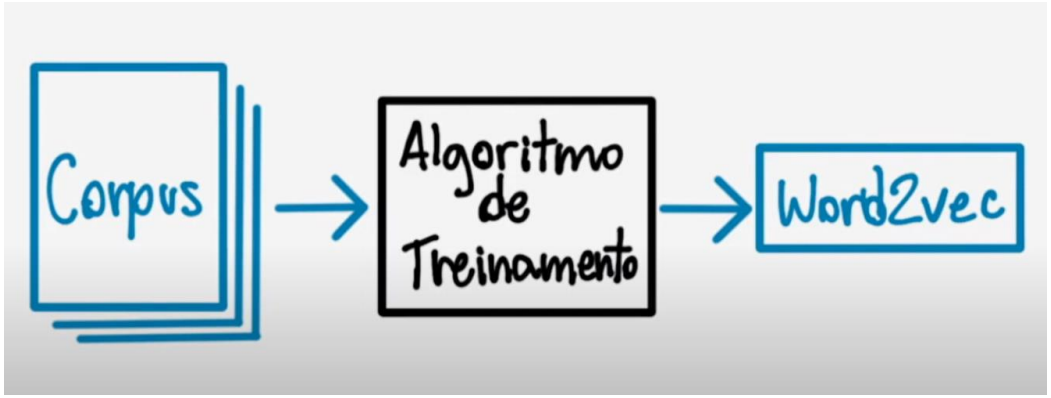
Exemplo de visualização de um Embedding: https://projector.tensorflow.org/

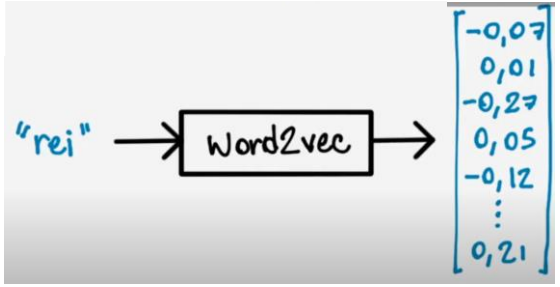
Word2Vec

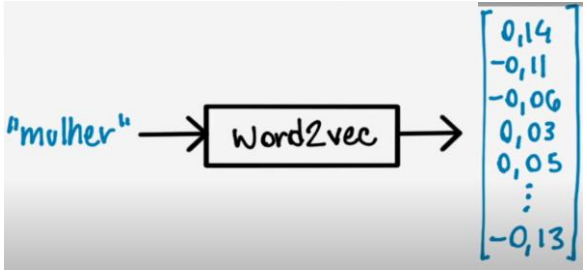
Word2Vec

- Como vimos, qualquer objeto pode ser representado através de vetores. Por hora, vamos olhar para vetores de palavras já treinados (depois vamos entender como chegar nesses vetores e o que ele representa) para entender suas principais propriedades.

- São técnicas que usam **redes neurais** para produzir **embeddings** que preservam algumas propriedades semânticas das palavras:







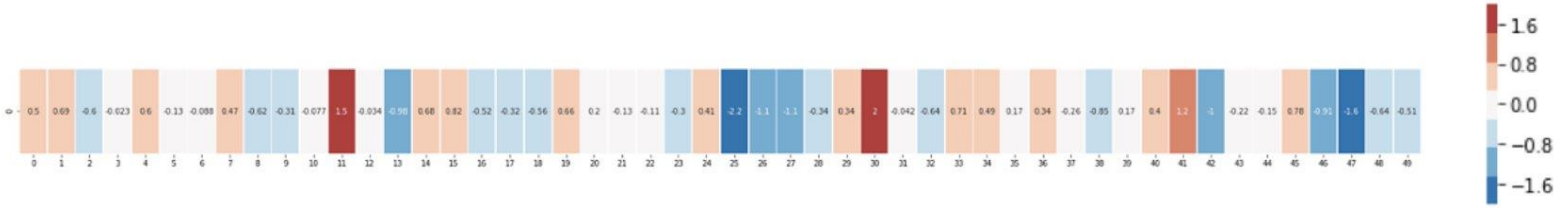
Word2Vec

- Este vetor representa a palavra “King”:

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 , -0.31012 ,
-0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961
, -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 ,
-0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 ,
-1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

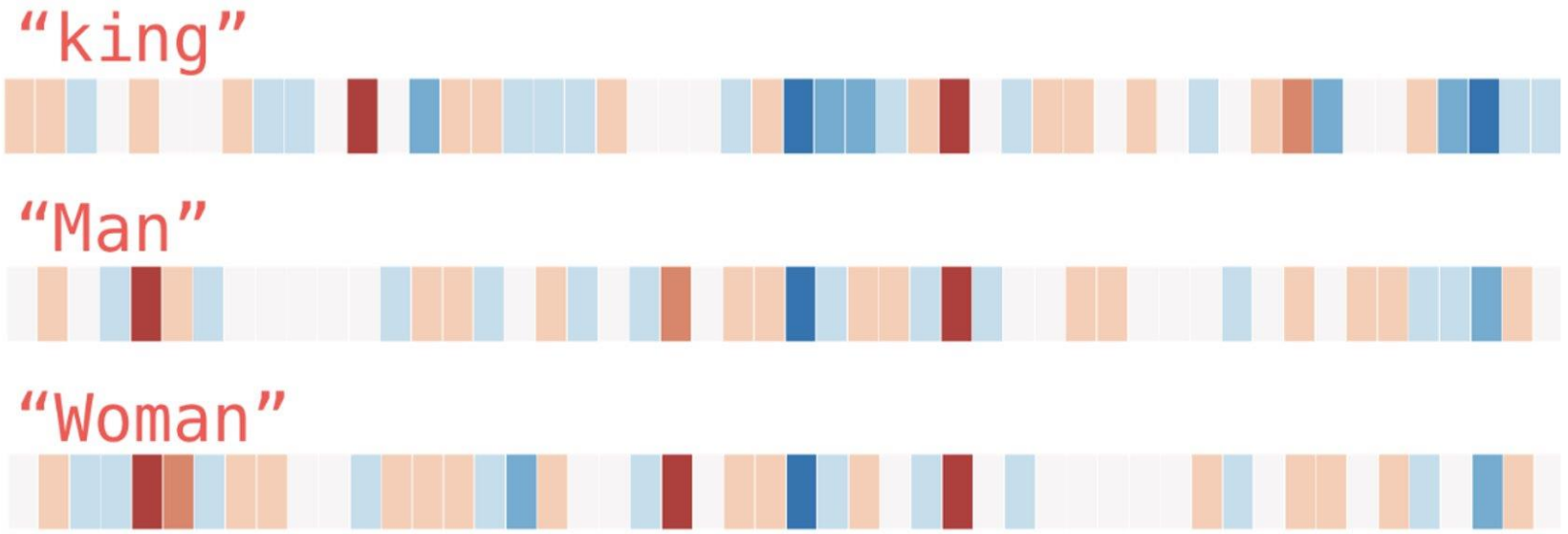
- É uma lista de 50 números. Vamos coloca-los numa única linha para poder compará-la com vetores de outras palavras:

- Podemos colorir cada uma das células para melhor visualização e comparação:

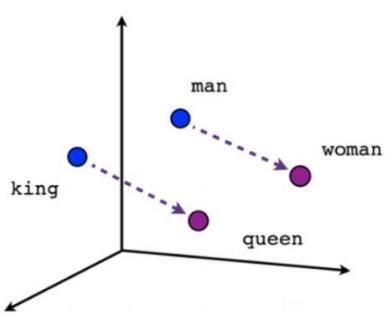


- Vamos deixar os números de lado e focar somente nas cores.

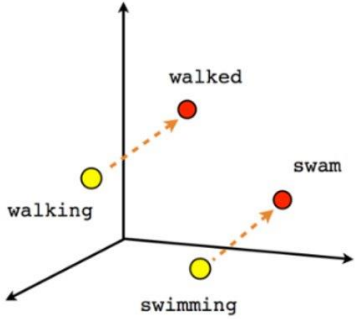
- Comparando com outros vetores de palavras:



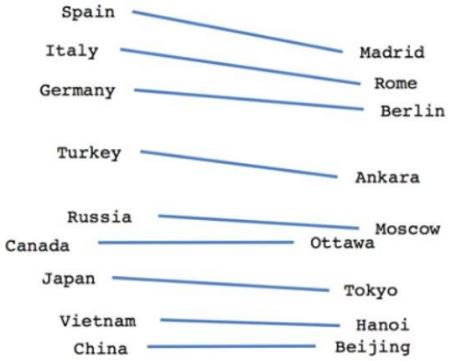
- Esse tipo de representação vetorial é que nos permite estabelecer relações como as seguintes:



Male-Female

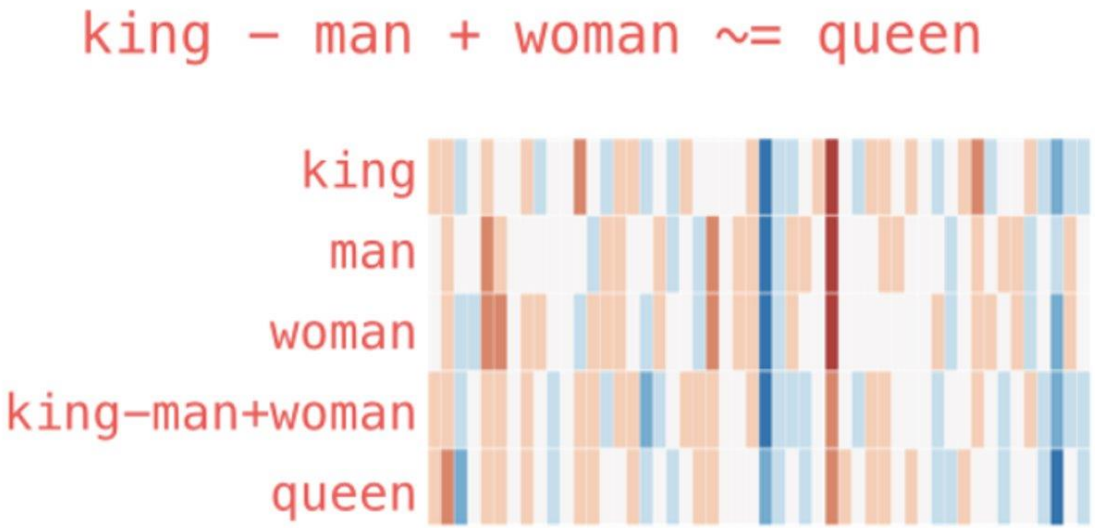


Verb tense



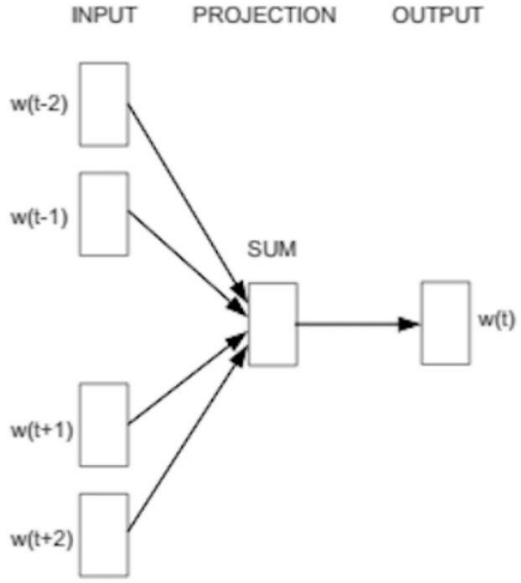
Country-Capital

- E mais, eu posso realizar operações algébricas do tipo: v[‘king’]-v[‘man’]+v[‘woman’] = v[‘queen’]

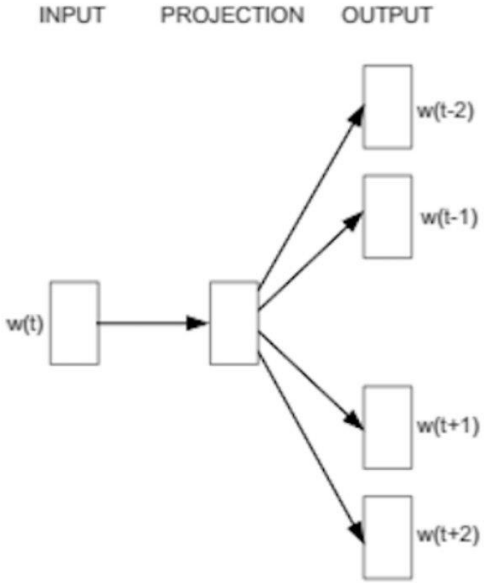


Qual a ideia? Encontrar uma representação a partir do contexto das palavras vizinhas.

- Word2Vec pode criar vetores densos a partir de duas abordagens: ***C**ontinuous **B**ag-**o**f-**W**ord (CBOW)* e *Skip-gram:*



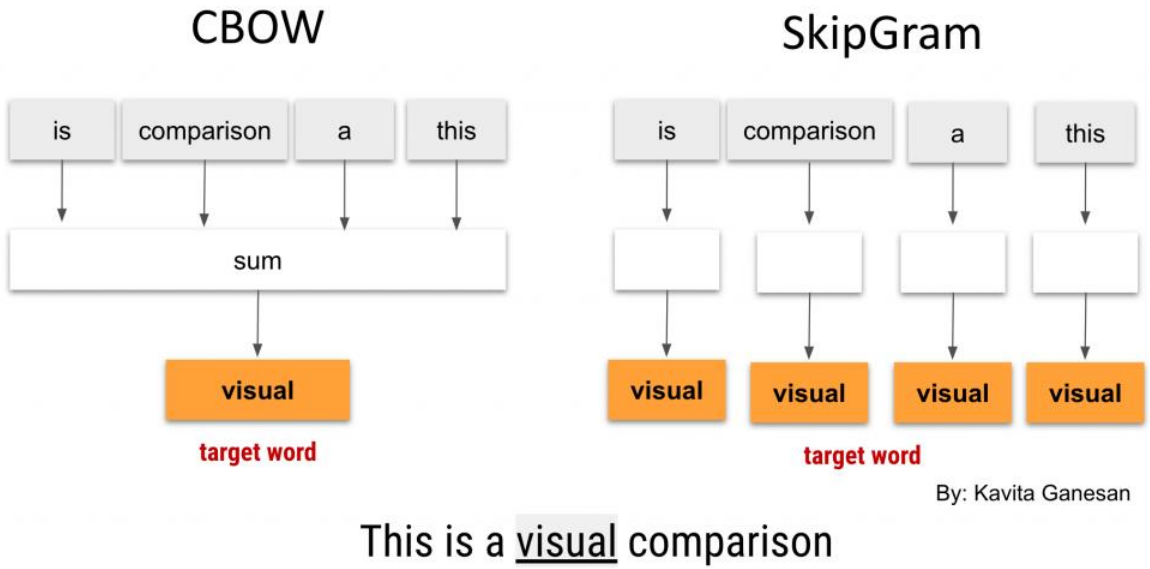
CBOW



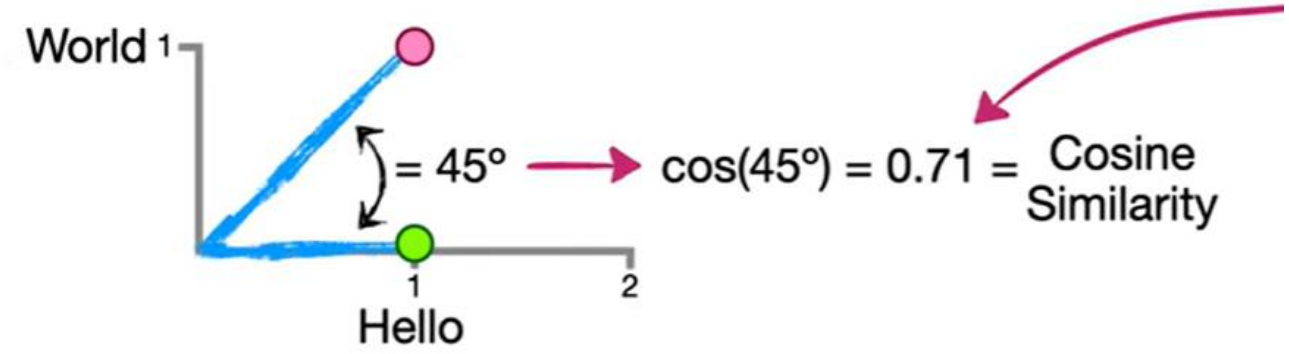
Skip-gram

CBOW
predizer uma palavra a partir de um contexto (outras palavras).

Skip-gram
predizer o contexto (outras palavras) a partir de uma palavra.



A similaridade cosseno é uma medida de semelhança entre dois vetores não nulos.



Demo e Exercício

Obrigado!

profanderson.dourado@fiap.com.br

/anderson-dourado


FIAP