Verena Saêta
Ricardo Tanaka

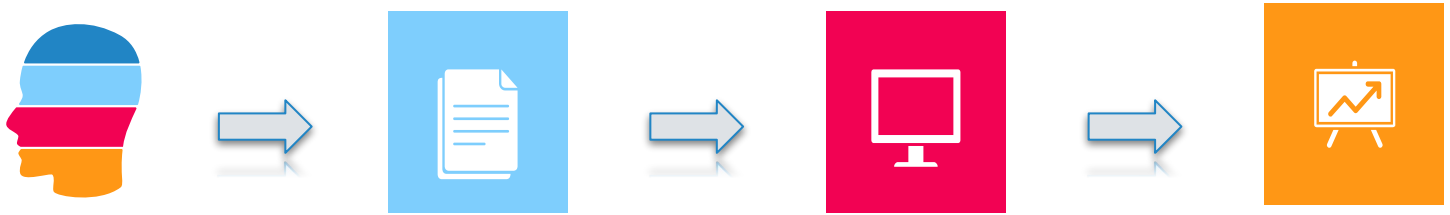
PROCESSAMENTO DE LINGUAGEM NATURAL

Extração de valor a partir de texto

O QUE É PROCESSAMENTO DE LINGUAGEM NATURAL?

O que é PLN?

- Conjunto de métodos para tornar a linguagem humana acessível a máquinas
- Subcampo da linguística e ciência de dados
- Inclui técnicas eficientes para representação de dados textuais
- Analisa e produz insights de dados de áudio e texto



O que é PLN?

Representação para busca

Modelos com base matemática

1990

- Classificação manual
- Palavras-chave
- Gramática

1990 - 2010

- Probabilidade e estatística
- Aprendizado de máquina
- Inferência de padrões e regras
- Internet

2010

- Redes neurais
- Vetorização de palavras
- Correlações

1.

Primeiros passos

Como iniciar o tratamento de dados textuais

Primeiros passos

Amplo conhecimento do contexto dos dados e possíveis interpretações

Limpeza eficiente dos dados:

- Remover palavras sem significado semântico (stopwords)
- Remover acentos e pontuações
- Reverter as palavras do vocabulário para o infinitivo (stemming/lemmatization)

Original	Lemma
somos, sou	ser
patos, pata	pato

Original	Stem
pedra, pedrada	pedr
abdicar, abdicou	abdic

Primeiros passos - exemplo

Texto:

“Eu gosto de conversar sobre processamento de linguagem natural com meus amigos!”

1. **Remover *stopwords***

   “gosto conversar processamento linguagem natural amigos!”

2. **Remover acentos e pontuação**

   “gosto conversar processamento linguagem natural amigos”

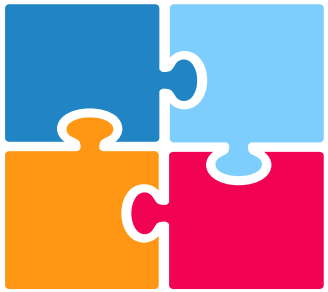
3. ***Lemmatizar* o texto**

   “gostar conversar processamento linguagem natural amigo”

Gostar conversar processamento linguagem natural amigo

0 0 1 0 0 0

Embedding



Desafio

Como representar textos numericamente sem perder o sentido e o contexto?

Soluções Propostas

Desenvolvimento de técnicas para representação de texto

Conceitos de álgebra linear (vetores/matrizes)

Exemplos:

tf-idf

word2vec

2.

tf-idf

term frequency - inverse document frequency

tf-idf

Frequência de termos

Hans Peter Luhn - 1957

Um documento

Aumenta pontuação para palavras repetidas no documento

$$wf_{t,d} = \begin{cases} 1 + log\ tf_{t,d}\ , \text{ if } tf_{t,d} > 0 \\ 0\ , \text{ otherwise} \end{cases}$$

Inversa da frequência nos documentos

Karen Spärck Jones - 1972

Conjunto de documentos

Reduz pontuação para palavras que se repetem no conjunto

$$idf_t = \log \frac{N}{df_t}$$

tf-idf

$$w_{t,d} = wf_{t,d} * idf_t$$

Método estatístico

Pontuação de palavras

Busca na internet

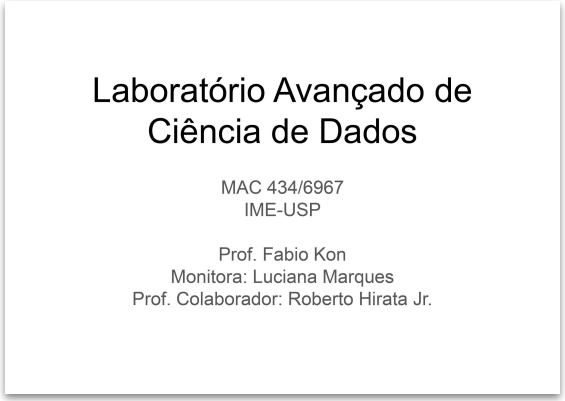
Mineração de dados

Vetorização de texto

Entrada para aprendizado de máquina

Tf-idf - exemplo

- Conjunto de dados:
  - Slides da aula de introdução a MAC0434
  - Cada slide representa 1 documento
- scikit-learn
  - sklearn.feature_extraction.text.TfidfVectorizer
- nltk - Natural Language Toolkit
  - nltk.corpus.stopwords.words("portuguese")
- https://github.com/raktanaka/mac0434-2020-tf-idf



Tf-idf - exemplo

Palavras	pontuação
grupo	1.06059
cliente	0.87529
trabalho	0.85097
presença	0.66884
dados	0.66821

Tf-idf - exemplo

documento	cliente	dados	grupo	presença	trabalho
0	0.0	0.14630	0.0	0.0	0.0
1	0.11937	0.16193	0.0	0.0	0.0
2	0.07328	0.0	0.08927	0.0	0.0
3	0.20341	0.27592	0.24780	0.0	0.41389
4	0.0	0.0	0.0	0.48247	0.0
5	0.10686	0.0	0.26036	0.18636	0.14496
6	0.0	0.0	0.0	0.0	0.12402
7	0.10765	0.0	0.0	0.0	0.0
8	0.14077	0.0	0.08574	0.0	0.0
9	0.12392	0.08404	0.37740	0.0	0.16809

3.
word2vec

Efficient Estimation of Word Representations in Vector Space

You shall know a word by the company it keeps.

Firth, J.R. 1957:11

Word2vec

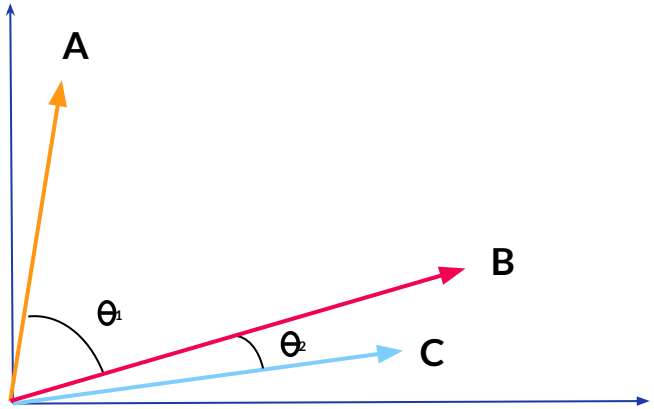
Desenvolvido em 2013 por Tomas Mikolov (Google)

Produz vetores semelhantes para palavras com contextos semelhantes

A similaridade (*S*) entre os vetores de palavras é medida através do produto escalar entre eles

Exemplo:

[“Cachorro”, “negócio”, “estratégia”]



$$S_{A,B} = \cos(\Theta_1) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

$$S_{B,C} = \cos(\Theta_2) = \frac{B \cdot C}{\|B\| \cdot \|C\|}$$

$$\downarrow \Theta \ \therefore \uparrow \cos(\Theta) \ \therefore \uparrow S$$

Word2vec

Composto por dois algoritmos de redes neurais:

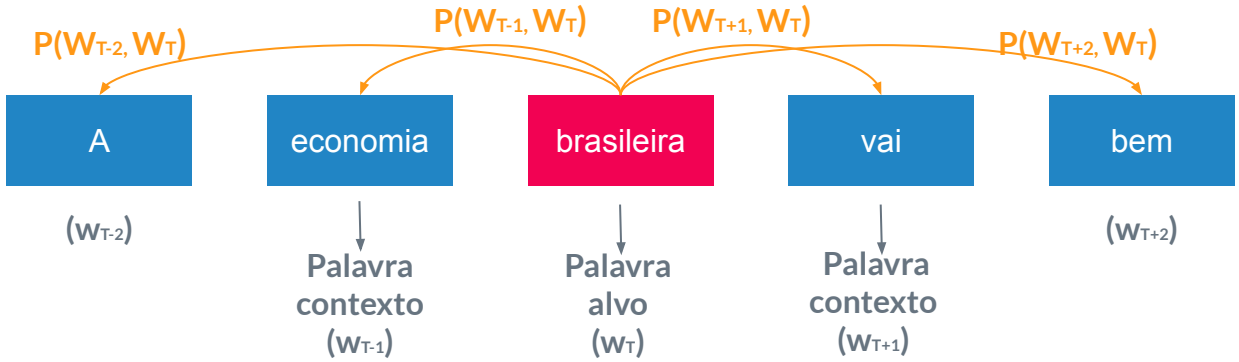
1. **Skip-gram (SG)**

   Prediz as palavras - contexto dada uma palavra - alvo

2. **Continuous bag of words (CBOW)**

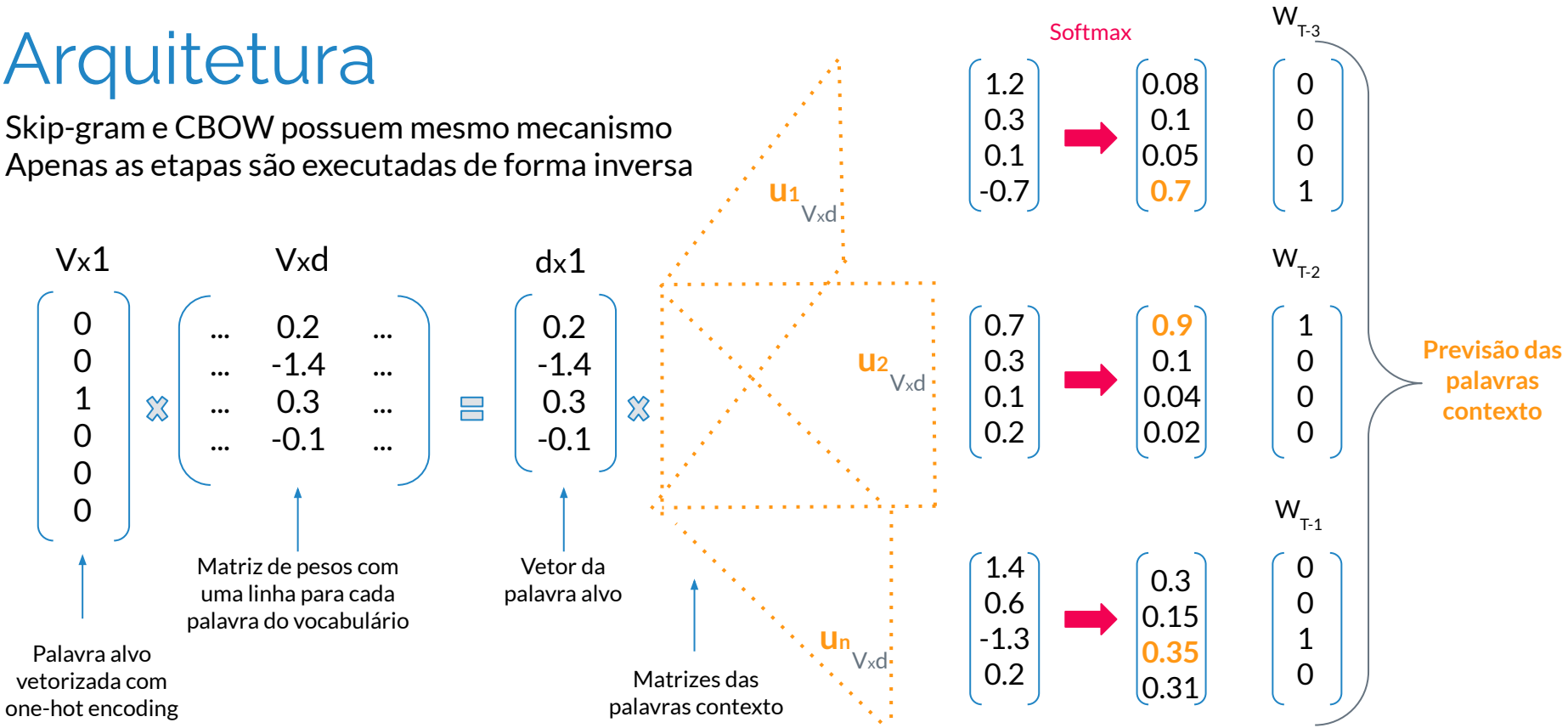
   Prediz a palavra - alvo dado um conjunto de palavras - contexto

Exemplo: "...A economia brasileira vai bem..."



Arquitetura

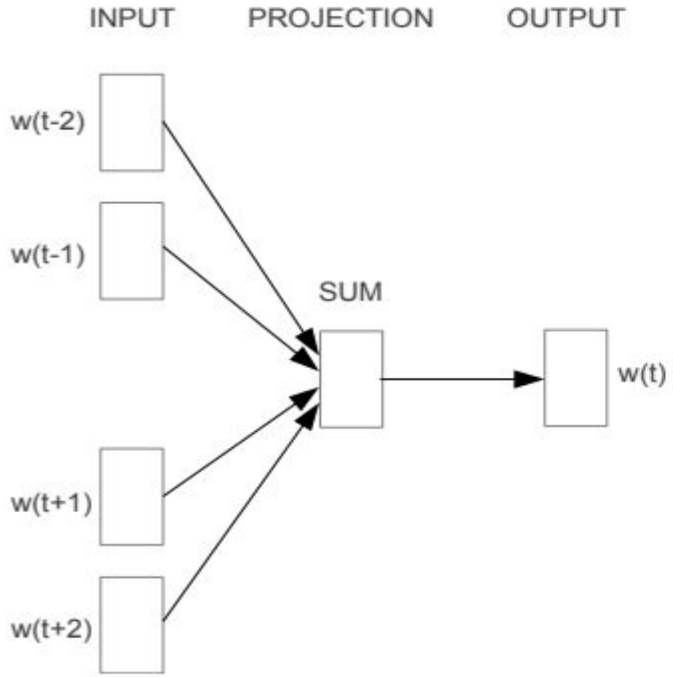
Skip-gram e CBOW possuem mesmo mecanismo
Apenas as etapas são executadas de forma inversa



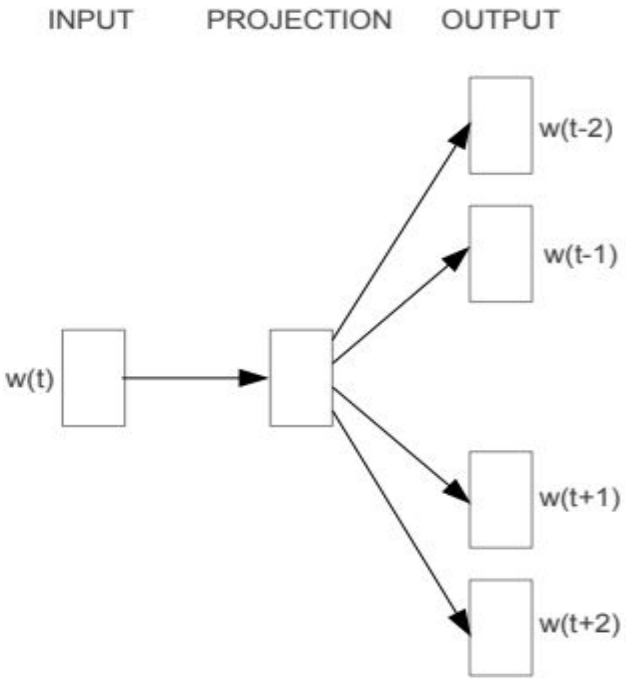
- Tamanho do vocabulário (V)
- Hiper-parâmetro (d)
- Window size (T)

CBOW ← SG →

Arquitetura



CBOW



Skip-gram

Detalhes

- Função de custo (J): vetores de palavras são continuamente reajustados para minimizar a perda
- Função softmax: viabiliza a distribuição de probabilidades para estimação dos vetores
- Matriz inicial de pesos: inicialização randômica para posterior otimização
- Backpropagation: otimização dos pesos via minimização da função de custo

$$J(\Theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-m<=j<=m} log\left(p\left(W_{T+j}, W_T\right)\right)$$

Função de custo

$$p(o|c) = \frac{exp(u_c^T \cdot v_a)}{\sum_{w=1}^{V} exp(u_w^T \cdot v_a)}$$

Função Softmax

Novas Soluções

Doc2Vec

Extensão de Word2Vec

Vetor parágrafo

BERT

Bidirectional Encoder Representations from Transformers

Representação não-supervisionada de linguagem, profunda e bi-direcional

Obrigada!

Alguma dúvida?

Alunos:

Verena Saêta

Ricardo Tanaka