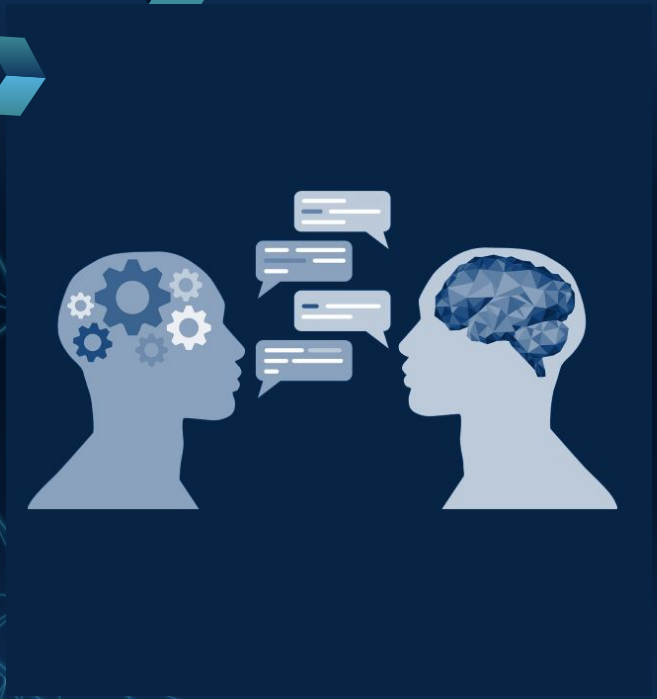
Natural Language processing

01

Conceitos Básicos



Afinal, o que é NPL?

- É uma área de inteligência artificial que envolve principalmente a implementação de sistemas e algoritmos que sejam capazes de interagir com a linguagem humana;

- Ex: Tradução, responder perguntas, sumarização, etc.

Principais aplicações

Sequence Classification

Aplicações de classificação, atribuir uma classe para cada sequência.

- Análise de sentimentos: Sequência (x) -> alegre;
- Categorização de documentos:: Documento -> tema: esportes;
- Seleção de frases para respostas: Pergunta -> selecionar melhor trecho de um texto para o questionamento -> resposta;

Pairwise seq. classification

Compara duas sequências de acordo com sua similaridade

- Em geral é binário, retornando 1 se a sequência tem o mesmo significado e -1 caso contrário;
- Caso prático - Quora Question Pairs (kaggle):
  Encontrar perguntas duplicadas;

https://www.kaggle.com/competitions/quora-question-pairs

Word Labeling

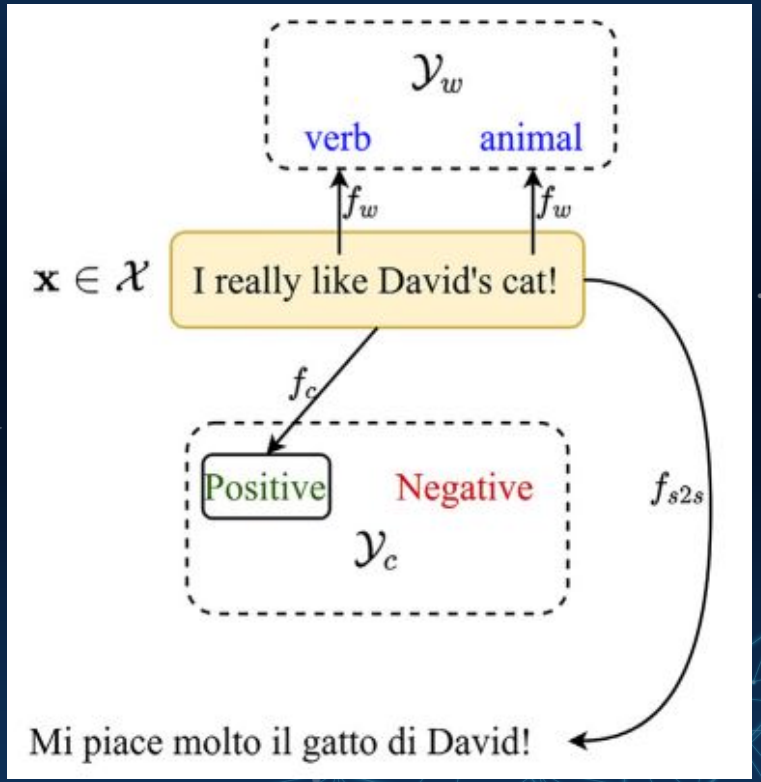
Atribui um rótulo a cada palavra (token) de uma sequência.

- analisa cada palavra isoladamente no seu contexto.
- Ex: Named Entity Recognition (NER): identifica entidades importantes e classifica em categorias pré-definidas

Sequence2sequence

Gera uma nova sequência a partir de uma sequência de entrada.

EX: machine translation. Obs: A entrada e saída podem ou não estar alinhadas token a token



Principais dificuldades

Representação

Uma das principais dificuldades é encontrar uma boa representação de frases, tokens, documentos etc;

Subjetividade

Dificuldade de lidar com a ambiguidade da comunicação humana, como em contextos informais, sarcasmo, etc.

Custo computacional

O treinamento de um modelo robusto exige quantidades massivas de dados e processamento, um modelo robusto tem cerca de 110-340 milhões de parâmetros que podem ser aprendidos. Obs: também discute-se a pegada do carbono.

02

Introdução à NLP

Pré - Processamento

Tokenziação:

“Sou trainee do CIS!”
[“Sou”,”trainee”,”do”,”CIS”,”!”]

Remoção de Stop-Words

Os, As, O, A, e
The, And

Stemming

Raiz em comum:
“Correr, Correu, Correndo, Corra”
“Corr”

Lemmatização:

Palavra existente
Correr

Representação de Texto

One-Hot Encoding

Vetores binários com base na posição:
“Sou trainee do CIS!”
Trainee = [01000]

TF-IDF

Term Frequency - Inverse Document Frequency

Word-embedding

Representação Vetorial
Rei - Rainha
Homem - X

Modelos Básicos

N-Grams

“Sou trainee do CIS!”
“Sou Trainee”
“Trainee do”
“Do CIS”
“CiS !”

Bag of Words

Ordem das palavras não importam
Saco de palavras
Classificação de Texto

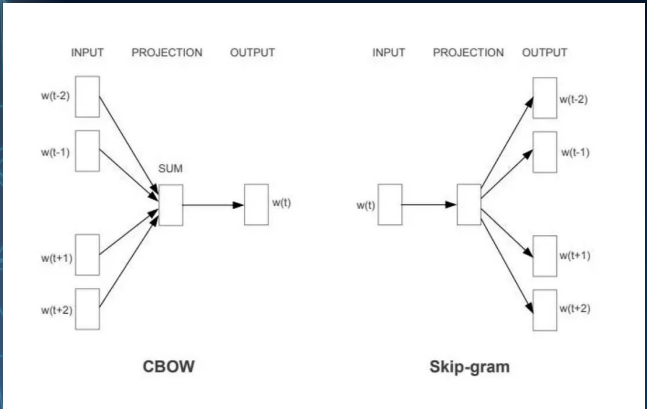
TD-IDF como modelo

Valores de TF-IDF como features dos modelo de Machine Learning
Classificador

Modelos Avançados

Word2Vec

Representação vetorial das palavras.
CBOW e SKIP-Gram



GloVe

Global Vector for Word Representation
Engloba todo o corpus
Contagem de palavras e predição baseada e contexto

BERT

Bidirectional Encoder Representation from Transformers
Modelo transformador pre-treinado bidirecionalmente
Considera as palavras anteriores e posteriores
Compreende Nunce e ambiguidades

03

Overview Histórico

Breve overview histórico

Abordagem estatística

- Anos 1990s - 2000s
- Baseada em relações estatísticas extraídas de grandes corpos de texto
- Aplicações: tradução automática, correção gramatical
- Limitações principalmente de contexto

Abordagem baseada em Deep learning

- 2010s - Hoje
- Uso de redes neurais profundas e representações vetoriais
- Aplicações:compreensão e geração de texto, classificação de sentimentos, etc
- Limitações: alto custo, necessidade de grande volume de dados

IBM Model 1

```
--------------------------------------------------
['aqui', 'estamos', 'mais', 'uma', 'vez']
['here', 'are', 'more', 'a', 'again']
--------------------------------------------------
['hoje', 'iremos', 'discutir', 'sobre', 'aumento', 'do', 'combustível']
['today', 'we', 'discuss', 'on', 'increase', 'of', 'contain']
--------------------------------------------------
```

TF-IDF

- Ponte entre estatística clássica e representações vetoriais
- Mesmo não usando mecanismos de atenção, é possível alcançar resultados satisfatórios

04

TF-IDF e Recomendação de Filmes

Term Frequency-Inverse Document Frequency

Frequência do Termo (TF – Term Frequency)

Quão frequentemente uma palavra aparece em *um documento específico.*

TF(t,d) = Número de Ocorrências de t em d/Número total de palavras em d

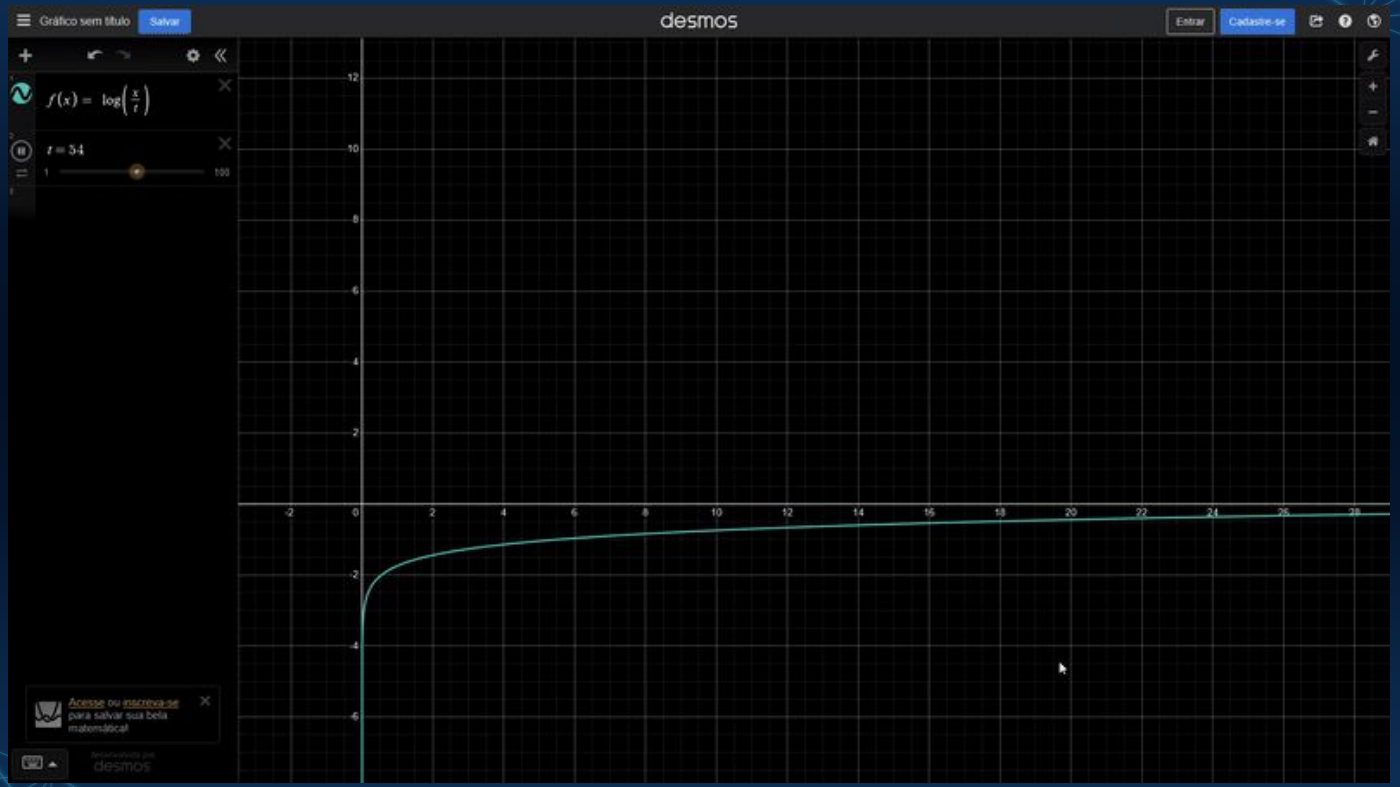
Frequência Inversa do Documento (IDF – Inverse Document Frequency)

Quão rara é uma palavra em *todos os documentos da coleção.*

IDF(t,D)= log(Número total de documentos em D/Número de documentos com t em D)

- O score TF-IDF de uma palavra é o **produto do seu TF pelo seu IDF**.
- TFIDF(t,d,D)=TF(t,d)*IDF(t,D)
- Uma palavra tem um **TF-IDF alto** se ela **aparece com frequência em um documento** (TF alto) **E é rara na coleção de documentos** (IDF alto). Isso a torna uma palavra "chave" para aquele documento.

Por que usar log no IDF?



D1: "Arroz e feijão faz bem para a saúde"
D2: "Feijão é muito bom"
D3: "Dieta com arroz e feijão"
D4: "O homem é condenado a ser livre"

DOCUMENTO	FRASE ORIGINAL	FRASE TOKENIZADA
D1	"Arroz e feijão faz bem para a saúde"	["arroz", "e", "feijão", "faz", "bem", "para", "a", "saúde"]
D2	"Feijão é muito bom"	["feijão", "é", "muito", "bom"]
D3	"Dieta com arroz e feijão"	["dieta", "com", "arroz", "e", "feijão"]
D4	"O homem é condenado a ser livre"	["o", "homem", "é", "condenado", "a", "ser", "livre"]

Calculando o TF(t,d)

Documento	a	arroz	bem	bom	com	condenado	dieta	e	é	faz	feijão	homem	livre	muito	o	para	saúde	ser	Total Termos
D1	1.250	1.250	1.250	0	0	0	0	1.250	0	1.250	1.250	0	0	0	0	1.250	1.250	0	8
D2	0	0	0	2.500	0	0	0	0	2.500	0	2.500	0	0	2.500	0	0	0	0	4
D3	0	2.000	0	0	2.000	0	2.000	2.000	0	0	2.000	0	0	0	0	0	0	0	5
D4	1.429	0	0	0	0	1.429	0	0	1.429	0	0	1.429	1.429	0	1.429	0	0	1.429	7

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

Calculando o IDF(t)

Termo	Doc. Frequência (df)	IDF(t, D) - Log(N/df)
a	2	log10(4/2)=0.3010
arroz	2	log10(4/2)=0.3010
bem	1	log10(4/1)=0.6021
bom	1	log10(4/1)=0.6021
com	1	log10(4/1)=0.6021
condenado	1	log10(4/1)=0.6021
dieta	1	log10(4/1)=0.6021
e	2	log10(4/2)=0.3010
é	2	log10(4/2)=0.3010
faz	1	log10(4/1)=0.6021
feijão	3	log10(4/3)=0.1249
homem	1	log10(4/1)=0.6021
livre	1	log10(4/1)=0.6021
muito	1	log10(4/1)=0.6021
o	1	log10(4/1)=0.6021
para	1	log10(4/1)=0.6021
saúde	1	log10(4/1)=0.6021
ser	1	log10(4/1)=0.6021

$$IDF(t) = log \frac{N}{1 + df}$$

Calculando o TF-IDF(t,d)

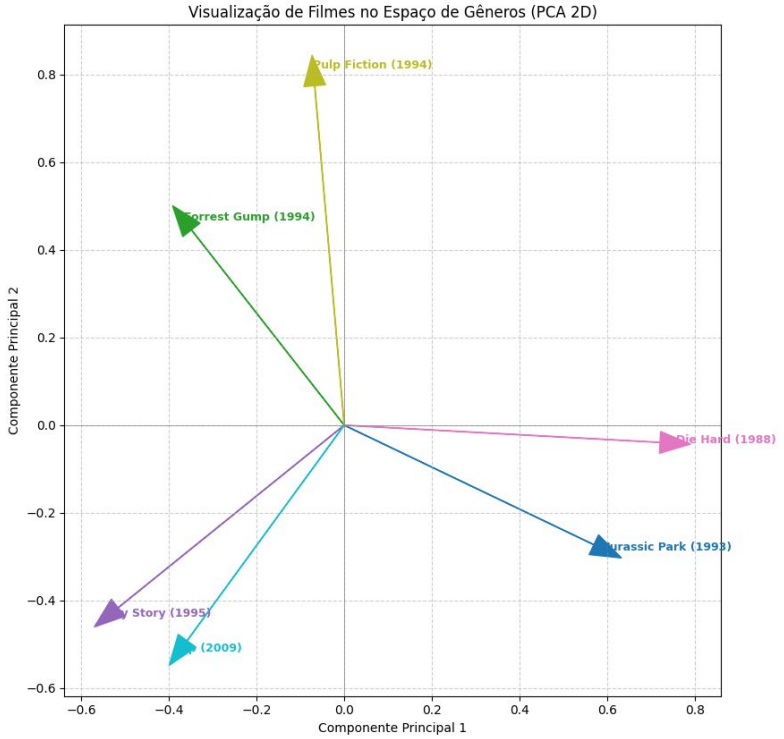
Documento	a	arroz	bem	bom	com	condenado	dieta	e	é	faz	feijão	homem	livre	muito	o	para	saúde	ser
D1	376	376	753	0	0	0	0	376	0	753	156	0	0	0	0	753	753	0
D2	0	0	0	1.505	0	0	0	0	753	0	312	0	0	1.505	0	0	0	0
D3	0	602	0	0	1.204	0	1.204	602	0	0	249	0	0	0	0	0	0	0
D4	430	0	0	0	0	860	0	0	430	0	0	860	860	0	860	0	0	860

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

Como funciona no sistema de Recomendação

- Cada filme é um DOCUMENTO, e os gêneros desses filmes como as palavras que compõem esse documento
- Para um filme específico, quão frequentemente um gênero aparece nele. Como um gênero aparece apenas uma vez por filme (ou não aparece), o TF será simplesmente 1 (se presente) ou 0 (se ausente), ou uma proporção se você normalizar pela contagem de gêneros no filme.
- IDF calcula o quão raro é um gênero em *todos os filmes da sua coleção.*
- Gêneros muito comuns (como "Drama" ou "Comédia") terão um IDF baixo, pois não ajudam muito a distinguir um filme do outro.
- Gêneros mais específicos (como "Film-Noir" ou "Musical") terão um IDF alto, indicando que são mais distintivos.

Como funciona no sistema de Recomendação




```
--- Ângulos de Similaridade de Cosseno entre Filmes (em Graus) ---
'Jurassic Park (1993)' e 'Die Hard (1988)':
  Similaridade de Cosseno: 0.2843, Ângulo: 73.48°
'Forrest Gump (1994)' e 'Pulp Fiction (1994)':
  Similaridade de Cosseno: 0.3545, Ângulo: 69.24°
'Toy Story (1995)' e 'Up (2009)':
  Similaridade de Cosseno: 0.5813, Ângulo: 54.46°
```

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

05

Demonstração do Sistema

O Dataset

movies.csv

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action|Crime|Thriller
7	Sabrina (1995)	Comedy|Romance
8	Tom and Huck (1995)	Adventure|Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action|Adventure|Thriller
11	American President, The (1995)	Comedy|Drama|Romance
12	Dracula: Dead and Loving It (1995)	Comedy|Horror
13	Balto (1995)	Adventure|Animation|Children
14	Nixon (1995)	Drama
15	Cutthroat Island (1995)	Action|Adventure|Romance

ratings.csv

userId	movieId	rating	timestamp
1	1	4	964982703
1	3	4	964981247
1	6	4	964982224
1	47	5	964983815
1	50	5	964982931
1	70	3	964982400
1	101	5	964980868
1	110	4	964982176
1	151	5	964984041
1	157	5	964984100
1	163	5	964983650
1	216	5	964981208
1	223	3	964980985
1	231	5	964981179
1	235	4	964980908

06

Conclusão

Este modelo de recomendação é

- Simples de implementar
- Computacionalmente leve
- Altamente interpretável

Resultados relevantes

- Técnica escalável para catálogos grandes
- Abordagem serve como base para futuras melhorias com modelos mais avançados

Possíveis Melhorias

- Lematização e stemização
  - Reduz palavras à sua forma base

    (ex: “investigação”, “investigador” → “investigar”)
  - Pode ser feita com bibliotecas como spaCy ou NLTK
  - Agrupa variações da mesma palavra, aumentando a precisão da comparação

- Uso de embeddings em vez de TF-IDF
  - Substitui a vetorização tradicional por Word2Vec, Doc2Vec ou BERT
  - Captura semântica, contexto e sinônimos
  - Permite recomendação mesmo sem termos exatos em comum
  - Resulta em recomendações mais precisas e “inteligentes”

Outras Possíveis Aplicações

01

Recomendação de livros

Observando gêneros, palavras-chave ou resenhas textuais

02

E-commerce

Sugestão de produtos similares

03

Classificação e triagem de currículos

Representam experiências como vetores de termos

04

Filtragem de notícias

Agrupam e recomendam artigos com base em conteúdo textual

05

Análise de perfis em redes sociais

Identificam interesses ou comunidades a partir de vocabulário frequente

Referências

GIORGI, Pierpaolo Di; CATARSI, Federico; MANCO, Giuseppe. A survey on textual sequence classification tasks. *Neurocomputing*, v. 470, p. 250–271, 2022. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0925231221010997. Acesso em: 21 jun. 2025.

RUDER Sebastian, "A Review of the Neural History of Natural Language Processing", 2018.. Disponível em: http://ruder.io/a-review-of-the-recent-history-of-nlp/. Acesso em: 20 jun. 2025

“A Brief History of NLP”. Disponível em: https://www.wwt.com/blog/a-brief-history-of-nlp. Acesso em: 21 jun. 2025

Thanks!

- Yasmin Dayrell
- Samuel Arthur
- Halycia
- Paulo
- Felipe Gibin

Please keep this slide for attribution