



Introdução ao Processamento de Linguagem Natural usando Python

QUEM SOMOS NÓS?

Uma **comunidade feminina** de tecnologia que visa o **protagonismo feminino nas áreas de inteligência artificial e ciência de dados**.

Através de meetups e workshops gratuitos qualificamos as mulheres para que elas possam competir com os homens no mercado de trabalho.



LinkedIn



Instagram



Facebook

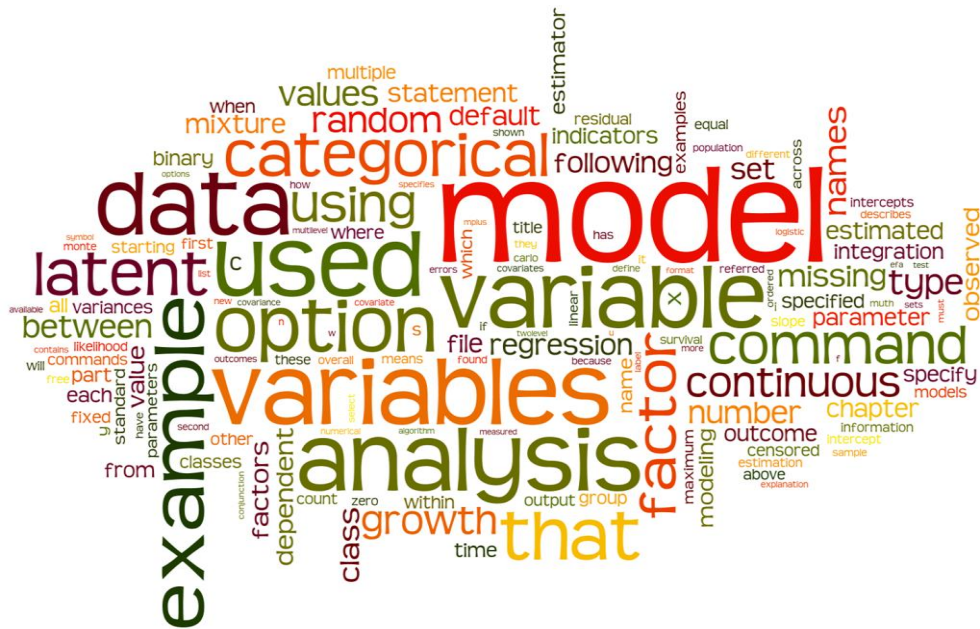


Dev.t



Roteiro de hoje

- Introdução
- Objetivo
- Ferramentas
- Relembrando...
- Fundamentos de Processamento de Linguagem Natural
- Pré Processamento dos dados
- Análise na Prática
- Bônus!
 - Um pouco de Deep Learning
- Quer praticar mais?
- Referências Bibliográficas



Introdução - O que é?

Processamento de língua natural (PLN)

É uma subárea da ciência da computação, **inteligência artificial** e da linguística que estuda os problemas da **geração e compreensão automática de línguas humanas naturais**. Sistemas de geração de língua natural convertem informação de **bancos de dados** de computadores em linguagem compreensível ao ser humano e sistemas de compreensão de língua natural convertem ocorrências de linguagem humana em representações mais formais, mais **facilmente manipuláveis** por programas de computador.

Processamento de linguagem natural é junção entre duas áreas: Ciência da computação e a Linguística.



Tipos de Linguagem

Linguagem Estruturada

Quando a linguagem é estruturada e independe de interpretação.

É fácil de ser processada pelo computador pois ela é definida por um conjunto restrito de regras ou gramaticais

Exemplo:

Matemática $\rightarrow y = 3x + 8$

Lógica $\rightarrow (A + B) \& (A + C)$

Programação $\rightarrow \text{Select nome from tabela;}$

Linguagem Não Estruturada

Qualquer texto que não siga uma estrutura pré-definida.

Possui regras gramaticais e algumas frases podem ter uma estrutura bem simples, mas na maior parte do tempo, a linguagem natural não é estruturada e ambígua.

Tipos de Linguagem

E como os computadores podem processar linguagem não estruturada?

Keywords, Parts of Speech, Names Entities, Datas, Quantidades, Contagem de Palavras, Bag of Words, Estatística....etc....

Ou seja, o computador primeiro precisa fazer a **extração da informação útil** daquele texto (Parse) de uma sentença antes de processá-la .



Linguagens

Línguas Naturais

Usadas no dia a dia e produzida por humanos.

Exemplo: Português, Inglês, Espanhol, Alemão.

Línguas Artificiais

Linguagens de programação e notações matemáticas.



PLN pode ser definido como uma forma de descobrir quem faz o quê, a quem, quando, onde, como e porquê.

Estágios da análise de NLP



Texto

Tokenização

Processo de dividir uma string em listas de pedaços (tokens).

Análise Léxica

Ou análise morfológica.
Estuda a construção da palavra, com seus radicais e afixos, que correspondem a partes estáticas e variantes das palavras, como as inflexões verbais

Análise Sintática

Envolve a análise das palavras em uma sentença de acordo com a gramática e arranjo de uma forma que mostra relação entre elas

Análise Semântica

Processos de mapeamento de sentenças de uma linguagem a fim de representar seu significado. É baseada nas construções obtidas na análise sintática

Análise Pragmática

Processamento da forma que a linguagem é utilizada para comunicar e como os significados da análise semântica agem sobre as pessoas e seu contexto.





WorldCloud

Análise de Sentimentos



Identificação de entidades

⟨Python⟩₁ não é somente ⟨uma⟩₂₃ (linguagem)₂ mais comumente conhecida para ⟨Ciencia de Dados⟩₃ ! ⟨Python⟩₄ também é utilizada em ⟨projetos⟩₅ grandes como por ⟨exemplo⟩₁₅ a ⟨NASA⟩₂₂ ! A ⟨NASA⟩₂₂ utiliza Python em diversos de seus ⟨projeto⟩₁₉, como por ⟨exemplo⟩₁₈ o ⟨SingleDop⟩₂₀, ⟨um⟩₂₄ ⟨toolkit⟩₁₁ que recuperara ⟨ventos⟩₁₀ bidimensionais de baixo ⟨nível⟩₈ a partir de ⟨dados⟩₆ de ⟨radar⟩₉ ⟨Doppler⟩₂₁ reais ou ⟨simulados⟩₁₆, e mais recentemente ⟨Python⟩₇ foi utilizado na ⟨criação⟩₁₇ da primeira ⟨imagem⟩₁₂ de ⟨um⟩₂₅ ⟨buraco negro⟩₁₃ da ⟨historia⟩₁₄ !



Google Cloud Platform

Natural Language API

1. Python Saliency: 0.13	OTHER	2. linguagem Saliency: 0.12	OTHER
3. Ciencia de Dados Wikipedia Article Saliency: 0.08	OTHER	4. Python Saliency: 0.05	OTHER
5. projetos Saliency: 0.05	OTHER	6. dados Saliency: 0.04	OTHER
7. Python Wikipedia Article Saliency: 0.04	ORGANIZATION	8. nível Saliency: 0.03	OTHER

17. criação Saliency: 0.03	EVENT	18. exemplo Saliency: 0.03	OTHER
19. projeto Saliency: 0.03	OTHER	20. Doppler Saliency: 0.03	PERSON
21. NASA Wikipedia Article Saliency: 0.03	ORGANIZATION	22. SingleDop Saliency: 0.02	ORGANIZATION
23. uma Saliency: 0.03	NUMBER	24. um Saliency: 0.03	NUMBER
25. um Saliency: 0.03	NUMBER		

Análise de dados do Reclame Aqui



ReclameAQUI



Disclaimer: Os dados que iremos analisar foram coletados diretamente do site Reclame Aqui manualmente. Não foram utilizados nenhum método de Web Scraping e o objetivo é puramente educacional.

Ferramentas - O que vamos usar?

Bibliotecas e Ferramentas



NLTK

spaCy

matplotlib



Relembrando...

- **Variáveis - Int e Strings**

- String representa um conjunto de caracteres disposto numa determinada ordem. Sempre que falarmos o termo String, estaremos nos referindo a um conjunto de caracteres (textos).
- Int são os dados compostos por caracteres numéricos.

- **Listas**

- Uma lista no Python armazena valores separados por vírgulas.

- **Expressão Regular/Regex**

- São usadas para identificar se um padrão existe em uma determinada sequência de caracteres (string) ou não.

- **Função**

- É uma sequência de comandos que executa alguma tarefa e que tem um nome definido por nós.

- **List comprehensions**

- É uma estrutura importantíssima para se trabalhar com grandes conjuntos de dados, tendo uma performance superior a outras estruturas em python e simplifica a escrita do código.

- **DataFrames**

- Dados tabulares bidimensionais, contém eixos rotulados (linhas e colunas).
- Podemos pensar como uma planilha excel



List Comprehension



```
lst = []
```

```
for x in 'PyLadies':
```

```
    lst.append(x)
```

```
lst = [x for x in 'PyLadies']
```



Notebook 1 - Relembrando Python ... com exercícios



Fundamentos de Processamento de Linguagem Natural





Notebook 2 - Intro NLP ... com exercícios também

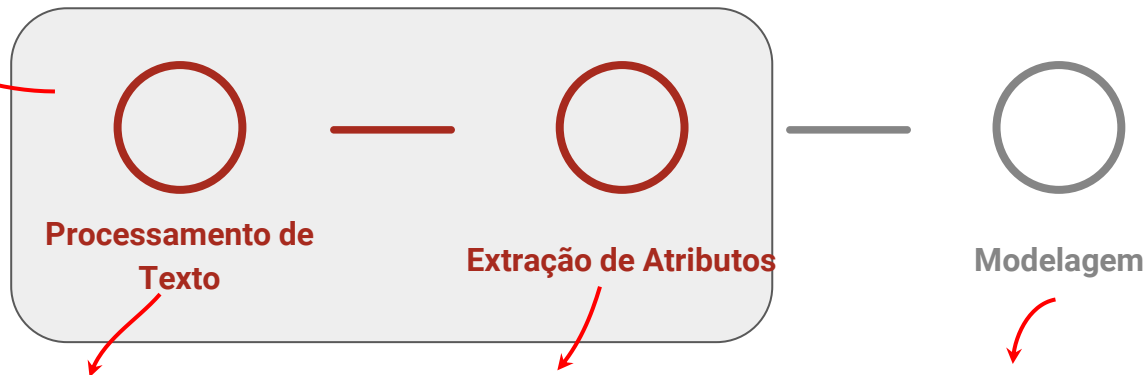


Pipeline de NLP



Pipeline de NLP

O que vamos ver hoje!



Aplicar ao nosso dataset (corpus) algum desses tipos de análise e realiza transformações no dataset:

- Limpeza
- Normalização
- Tokenização
- Stemming
- Lemmatization
- outros

Só terá resultados úteis se realizar o processamento de texto de forma bem-feita

- Aplicar técnicas estatística
- Coletar informações resumidas
- Outros tipos de análise

Aplicação de técnica de Machine Learning ou Deep Learning para automatizar o processo e entregar o resultado final

Corpus

O que é um Corpus?

Corpus é o conjunto de textos escritos e registros orais em uma determinada língua e que serve como base de análise.

Corpora é o plural de Corpus.

O termo dataset também é usado quando falamos de Corpus



- Memórias Póstumas de Brás Cubas (1881)
- Dom Casmurro (1899)
- Gênesis
- Folha de São Paulo (1994)

```
In [13]: from nltk.corpus import machado
```

Cada arquivo corresponde a uma das obras de Machado de Assis.

```
[ 'contos/macn001.txt', 'contos/cn002.txt', 'contos/macn003.txt', 'contos/macn004.txt', 'contos/macn005.txt', 'contos/macn006.txt', 'contos/macn007.txt', 'contos/macn008.txt', 'contos/macn009.txt', 'contos/macn010.txt', 'contos/macn011.txt', 'contos/macn012.txt', 'contos/macn013.txt', 'contos/macn014.txt', 'contos/macn015.txt', 'contos/macn016.txt', 'contos/macn017.txt', 'contos/macn018.txt', 'contos/macn019.txt', 'contos/macn020.txt', 'contos/macn021.txt', 'contos/macn022.txt', 'contos/macn023.txt', 'contos/macn024.txt', 'contos/macn025.txt', 'contos/macn026.txt', 'contos/macn027.txt', 'contos/macn028.txt', 'contos/macn029.txt', 'contos/macn030.txt', 'contos/macn031.txt', 'contos/macn032.txt', 'contos/macn033.txt', 'contos/macn034.txt', 'contos/macn035.txt', 'contos/macn036.txt', 'contos/macn037.txt', 'contos/macn038.txt', 'contos/macn039.txt', 'contos/macn040.txt', 'contos/macn041.txt', 'contos/macn042.txt', 'contos/macn043.txt', 'contos/macn044.txt', 'contos/macn045.txt', 'contos/macn046.txt', 'contos/macn047.txt', 'contos/macn048.txt', 'contos/macn049.txt', 'contos/macn050.txt', 'contos/macn051.txt', 'contos/macn052.txt', 'contos/macn053.txt', 'contos/macn054.txt', 'contos/macn055.txt', 'contos/macn056.txt', 'contos/macn057.txt', 'contos/macn058.txt', 'contos/macn059.txt', 'contos/macn060.txt', 'contos/macn061.txt', 'contos/macn062.txt', 'contos/macn063.txt', 'contos/macn064.txt', 'contos/macn065.txt', 'contos/macn066.txt', 'contos/macn067.txt', 'contos/macn068.txt', 'contos/macn069.txt', 'contos/macn070.txt', 'contos/macn071.txt', 'contos/macn072.txt', 'contos/macn073.txt', 'contos/macn074.txt', 'contos/macn075.txt', 'contos/macn076.txt', 'contos/macn077.txt', 'contos/macn078.txt', 'contos/macn079.txt', 'contos/macn080.txt', 'contos/macn081.txt', 'contos/macn082.txt', 'contos/macn083.txt', 'contos/macn084.txt', 'contos/macn085.txt', 'contos/macn086.txt', 'contos/macn087.txt', 'contos/macn088.txt', 'contos/macn089.txt', 'contos/macn090.txt', 'contos/macn091.txt', 'contos/macn092.txt', 'contos/macn093.txt', 'contos/macn094.txt', 'contos/macn095.txt', 'contos/macn096.txt', 'contos/macn097.txt', 'contos/macn098.txt', 'contos/macn099.txt', 'contos/macn100.txt', 'contos/macn101.txt', 'contos/macn102.txt', 'contos/macn103.txt', 'contos/macn104.txt', 'contos/'
```

Corpus

Entre no notebook e execute as células do “1. Exemplo - Corpus”

O que mais podemos fazer com esse corpus?



NLTK



**AGORA É
COM VOCÊ**

Dataset

Conjunto de Reclamações do Reclame Aqui

Nosso conjunto de dados é uma pequena amostra de reclamações retiradas do famoso site Reclame Aqui.

Vamos usar pandas para importar o csv

```
# Importando Pandas
import pandas as pd

# Vamos agora importar os dados que vamos trabalhar!
reclamacoes = pd.read_csv('reclamacoes.csv', sep=';')

print(reclamacoes.shape)

reclamacoes.head()
```



ReclameAQUI

Atenção!

Os nomes das lojas foram ocultados para mantermos a segurança da loja.
A empresa foi substituído por códigos para representar cada uma individualmente.

Dentro do corpo da reclamação, o nome da loja foi substituído por "LOJA"

Nomes e emails também foram removidos.



Dataset Reclame Aqui

No notebook e execute as células do

“Dados Reclame Aqui - 1. Importando os Dados”

Analise os dados, olhe o que temos em cada coluna, e como o dado está



**AGORA É
COM VOCÊ**

Pré - Processamento dos dados



Antes de qualquer coisa, precisamos limpar nossos dados!

Pré - Processamento dos dados

2.1 Separando os dados da coluna data_reclamacao

Repare bem na coluna data_reclamacao

Ela tem duas informações em um único dado... vamos separar!



Dado

```
# Vamos separar as informações

print('Como era antes:')
print(reclamacoes['data_reclamacao'][:3])

# Separando...
reclamacoes['data'] = reclamacoes['data_reclamacao'].str.split('às', expand=True)[0]

print('\n')
print('Como ficou agora:')
print(reclamacoes['data'][:3])
```

Como era antes:

0	23/03/19 às 18h33
1	23/03/19 às 18h20
2	23/03/19 às 11h07

Name: data_reclamacao, dtype: object

Como ficou agora:

0	23/03/19
1	23/03/19
2	23/03/19

Name: data, dtype: object



Informação

Pré - Processamento dos dados

2.1 Separando os dados da coluna data_reclamacao

Sua vez!

Faça a mesma coisa para a informação hora.

Você consegue identificar algo que está faltando ao utilizar essa nossa técnica?

Dica: utilize `reclamacoes.data[0]` para identificar

Tente resolver!



**AGORA É
COM VOCÊ**

Pré - Processamento dos dados

2.2 Quebrando a coluna local em Cidade e Estado

Sua vez...de novo!

Da mesma forma que você separou os dados de Data/Hora, separe agora os dados de Cidade/Estado da variável local

Não se esqueça de resolver aquele problema que identificamos no slide anterior!



**AGORA É
COM VOCÊ**



Pré - Processamento dos dados

Como ficou nossos dados

Depois de todas essas alterações como estão nossos dados?

```
# Visualizando as alterações que fizemos  
reclamacoes[['data_reclamacao', 'data', 'hora', 'local', 'cidade', 'estado']].head()
```

	data_reclamacao	data	hora	local	cidade	estado
0	23/03/19 às 18h33	23/03/19	18h33	Guarulhos - SP	Guarulhos	SP
1	23/03/19 às 18h20	23/03/19	18h20	Taubaté - SP	Taubaté	SP
2	23/03/19 às 11h07	23/03/19	11h07	Franco da Rocha - SP	Franco da Rocha	SP
3	23/03/19 às 10h57	23/03/19	10h57	Teresina - PI	Teresina	PI
4	22/03/19 às 19h49	22/03/19	19h49	São Gonçalo do Pará - MG	São Gonçalo do Pará	MG



Pré - Processamento dos dados

2.3 Alteração dos tipos das variáveis

Você percebeu algo estranho quando importou o seu dataset ou quando separou a data da coluna data_reclamacao?

```
# Verificando o tipo de dados  
reclamacoes.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 28 entries, 0 to 27  
Data columns (total 9 columns):  
empresa                28 non-null object  
data_reclamacao        28 non-null object  
local                  28 non-null object  
titulo_reclamacao      28 non-null object  
corpo_reclamacao       28 non-null object  
tags                   27 non-null object  
teve resposta          28 non-null int64  
data                   28 non-null object  
hora                   28 non-null object  
dtypes: int64(1), object(8)  
memory usage: 2.0+ KB
```



Vamos transformar para datetime!

Pré - Processamento dos dados

Alteração dos tipos das variáveis

Resultado:

```
# Colunas que são categoricas
reclamacoes['teve_resposta'] = reclamacoes['teve_resposta'].astype('category')

# Colunas que são datetime
reclamacoes['data'] = pd.to_datetime(reclamacoes['data'])

reclamacoes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28 entries, 0 to 27
Data columns (total 9 columns):
empresa                28 non-null object
data_reclamacao        28 non-null object
local                  28 non-null object
titulo_reclamacao      28 non-null object
corpo_reclamacao       28 non-null object
tags                    27 non-null object
teve_resposta          28 non-null category
data                   28 non-null datetime64[ns]
hora                   28 non-null object
dtypes: category(1), datetime64[ns](1), object(7)
memory usage: 1.9+ KB
```

Por que isso é importante?

Pré - Processamento dos dados

2.4 Lower Case

Antes de aplicar técnicas de NLP, precisamos fazer tratamento no texto também!

Uma delas é deixar todo o texto em caixa alta ou caixa baixa.

```
print('Antes:')
print(reclamacoes['corpo_reclamacao'].head())
```

```
Antes:
0  Nunca mais compro nessa loja pelo fato de que ...
1  Eu a LOJA comprar uma luva de musculação, na q...
2  Estive na LOJA da Marginal Tiete no dia 15 de ...
3  Comprei 3 produtos no dia 13.03 e recebi email...
4  comprei um tenis esportivo e ao receber o avis...
Name: corpo_reclamacao, dtype: object
```

```
# Aplicando Lower Case
reclamacoes['corpo_reclamacao'] = [str(token).lower() for token in reclamacoes['corpo_reclamacao']]

print('Depois:')
reclamacoes.corpo_reclamacao.head()
```

```
Depois:
0  nunca mais compro nessa loja pelo fato de que ...
1  eu a loja comprar uma luva de musculação, na q...
2  estive na loja da marginal tiete no dia 15 de ...
3  comprei 3 produtos no dia 13.03 e recebi email...
4  comprei um tenis esportivo e ao receber o avis...
Name: corpo_reclamacao, dtype: object
```



2.5 Tokenização

Nada mais que uma segmentação de Palavras ou quebra a sequência de caracteres

Existem 2 formas de tokenizar um texto:

- Por Palavra/Tokens
- Por Sentença

Pré - Processamento dos dados

2.5 Tokenização

13 Tokens

- Por Palavra/Tokens

A história do NLP começou na década de 1950, com Alan Turing

- Por Sentença

O Processamento de Linguagem Natural (PLN) é a subárea da Inteligência Artificial (IA) que estuda a capacidade e as limitações de uma máquina em entender a linguagem dos seres humanos. O objetivo do PLN é fornecer aos computadores a capacidade de entender e compor textos. "Entender" um texto significa reconhecer o contexto, fazer análise sintática, semântica, lexical e morfológica, criar resumos, extrair informação, interpretar os sentidos, analisar sentimentos e até aprender conceitos com os textos processados.

3 Sentenças



Pré - Processamento dos dados

2.5 Tokenização

Sua vez!

Separe os nossos textos em tokens e coloque em uma nova coluna chamada `corpo_reclamacao_tokens`

**AGORA É
COM VOCÊ**

OBS: Não esqueça de colocar o resultado dentro de uma nova coluna do dataset para não comprometer os nossos dados originais ok?

Dica: Use list Comprehension ou For



Pré - Processamento dos dados

2.6 Stopwords

Um detalhe **muito importante** no processamento de linguagem natural é identificar as chamadas **stopwords do idioma**.

Stopword nada mais é que uma palavra que possui **apenas significado sintático** dentro da sentença, porém **não traz informações relevantes sobre o seu sentido**.



Caso contrário, os algoritmos de Machine Learning podem dar **importância para palavras como: “e”, “ou”, “para”**e isso certamente atrapalha a análise.

OBS: O processo de “tokenização” do **NLTK** considera as pontuações do texto como tokens, por isso não podemos deixar de retirá-los também.

Pré - Processamento dos dados

2.6 Stopwords

Vamos para o Notebook!

```
# Removendo StopWords de todas as reclamações

# Percorre a lista de reclamações e cria uma coluna nova com o texto sem stopwords
for idx,text in enumerate(reclamacoes.corpo_reclamacao):
    print('Removendo StopWords do index {}'.format(idx))
    reclamacoes.at[idx, 'corpo_reclamacao_semStopWords'] = remove_stopwords(text, portuguese_stopwords)
    print('---'*20)
```

Removendo StopWords do index 0
Tamanho do texto original 467
Tamanho do texto sem stopwords 223
Foram removidas 244 stopwords

Removendo StopWords do index 1
Tamanho do texto original 159
Tamanho do texto sem stopwords 78
Foram removidas 81 stopwords

Removendo StopWords do index 2
Tamanho do texto original 160
Tamanho do texto sem stopwords 85
Foram removidas 75 stopwords

Removendo StopWords do index 3
Tamanho do texto original 288
Tamanho do texto sem stopwords 163
Foram removidas 125 stopwords



Pré - Processamento dos dados

2.6 Stopwords

Tokens como “de”, “estive na”,
“na”, “da” não trazem valor a
análise

```
reclamacoes.head()
```

titulo_reclamacao	corpo_reclamacao	tags	teve_resposta	data	corpo_reclamacao_tokens	corpo_reclamacao_semStopWords	Tags_semStopWords
Mau Atendimento	nunca mais compro nessa loja pelo fato de que ...	Calçados Esportivos	0	2019-03-23	[nunca, mais, compro, nessa, loja, pelo, fato,...	nunca compro nessa loja fato pior atendimento ...	calçados esportivos
Produtos misturado na gondola e divergência na...	eu a loja comprar uma luva de musculação, na q...	Divergência de valores, Acessórios para Muscul...	0	2019-03-23	[eu, a, loja, comprar, uma, luva, de, musculaçã...	loja comprar luva musculação gondola descrito ...	divergência valores acessórios musculação arti...
Informação errada do vendedor.	estive na loja da marginal tiete no dia 15 de ...	Produto errado, Raquetes e Tacos, Artigos Espo...	0	2019-03-23	[estive, na, loja, da, marginal, tiete, no, di...	loja marginal tiete dia 15 março 2019 procura ...	produto errado raquetes tacos artigos esportivos
Paguei por um produto fora do estoque	comprei 3 produtos no dia 13.03 e recebi email...	Acessórios de Vestuário	1	2019-03-23	[comprei, 3, produtos, no, dia, 13.03, e, rece...	comprei 3 produtos dia 13.03 recebi email conf...	acessórios vestuário
entregaram meu pedido a outra pessoa falando q...	comprei um tenis esportivo e ao receber o avis...	Problemas na finalização da compra Tênis Calça...	0	2019-03-22	[comprei, um, tenis, esportivo, e, ao, receber...	comprei tenis esportivo receber aviso pedido c...	problemas finalização compra tênis calçados es...



Repare que
foram
removidas os
tokens
desnecessários!



2.7 Normalização das palavras - Stemming

Stemming é uma técnica de remover prefixos e sufixos de uma palavra, chamada *stem*. Por exemplo, o *stem* da palavra reclamação é reclam. Essa técnica é muito usada em mecanismos de buscas para indexação de palavras. Pois, ao invés de armazenar todas as formas de uma palavra, o mecanismo de busca armazena apenas o *stem* da palavra, reduzindo o tamanho do índice e aumentando a performance do processo de busca.

```
nltk.download('rslp')
stemmer = nltk.stem.RSLPStemmer()

palavras = ['reclamação', 'reclamei', 'reclamando']

for w in palavras:
    print(stemmer.stem(w))
```

```
[nltk_data] Downloading package rslp to /home/nbuser/nltk_data...
[nltk_data]   Package rslp is already up-to-date!
reclam
reclam
reclam
```

Pré - Processamento dos dados

2.7 Normalização das palavras - Lemmatization

Lemmatization consiste em aplicar técnicas para deflexionar as palavras, retirando a conjugação verbal, caso seja um verbo, e altera os substantivos e os adjetivos para o singular masculino, de maneira a reduzir a palavra até sua forma de dicionário.

```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()
```

Exemplo de lematização, porém não existe uma biblioteca em português apenas em inglês no momento.

```
palavras = ['jumps', 'ladies', 'oranges']

for w in palavras:
    print(lemmatizer.lemmatize(w))
```

Análise na prática

3. Análise dos Dados

Sobre o que é o nosso principal dado?

RECLAMAÇÕES !!

Sobre o que?

Sobre quem?

Houve
resolução do
problema?



Vamos para o Notebook!

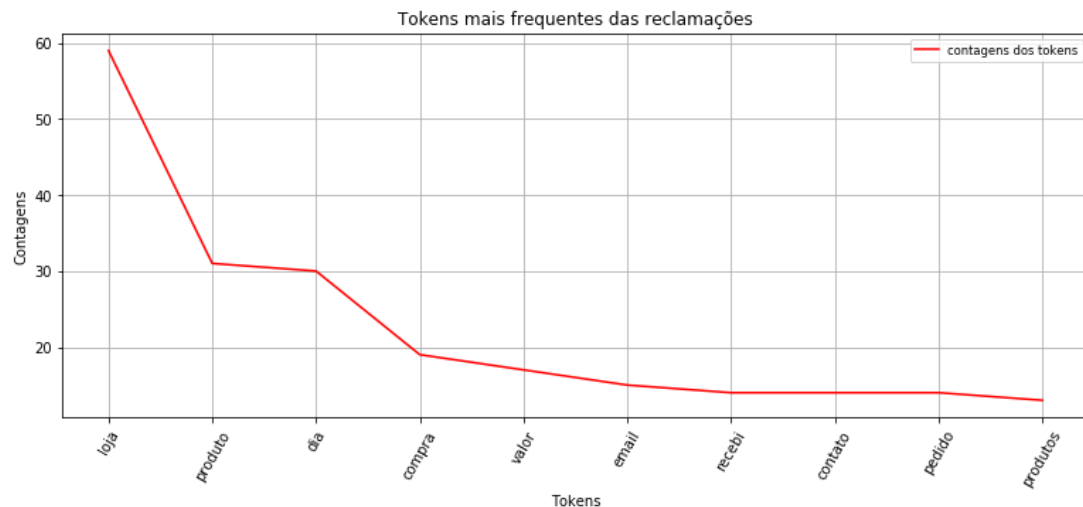
Análise na prática

FreqDist - é usada para codificar “distribuições de frequência”, que conta o número de vezes que cada resultado, no nosso caso palavras ocorre no nosso

corpus

```
#Juntando todas as mensagens em um unico texto
lista_reclamacao = ' '.join(reclamacoes.corpo_reclamacao_semStopWords.tolist())
frequencia = freq_Words(lista_reclamacao,10,"reclamações")
```

juntando todas as reclamações em um grande texto



usando nossa função `freq_words()` para verificar a frequências das palavras e plotando as 10 mais frequentes

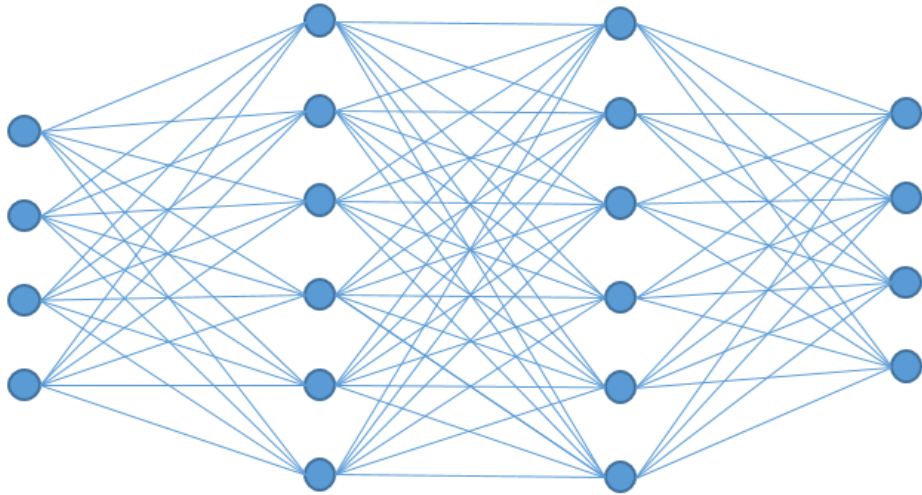
Exercício!

Como você faria para criar um WorldCloud de uma única empresa?

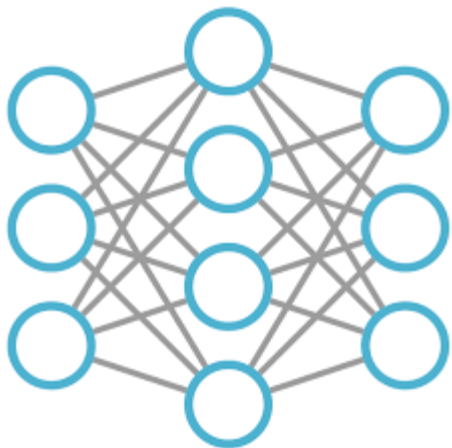
**AGORA É
COM VOCÊ**

Bônus!

Um pouco de Deep Learning!



Um pouco de Deep Learning!



Usando **spaCy** para coletar mais informações

Biblioteca de software de código aberto para processamento avançado de linguagem natural

Ao contrário do *NLTK* , que é amplamente utilizado para ensino e pesquisa, o spaCy se concentra no fornecimento de software para uso em produção

Vantagem do Spacy: Possui modelos de Deep Learning pré-treinados em Português!

The logo for spaCy, featuring the word "spaCy" in a blue, sans-serif font.

Vamos para o Notebook!

spaCy - NER (Named Entity Recognition)

É a extração de informações que procura localizar e classificar menções de entidades nomeadas em texto não estruturado em categorias predefinidas, como nomes de pessoas e organizações, locais, códigos médicos, expressões de tempo, quantidades, valores monetários, percentagens, etc.

Exemplo:

[Jim] Person comprou 300 ações da [Acme Corp.] Organization em [2006] Time .

```
# Visualizando de uma forma mais bonita!  
  
from spacy import displacy  
  
displacy.render(doc, style="ent")
```

Apesar da Maria PER morar em São Paulo LOC , ela me disse que seu sonho era morar em Nova York LOC



spaCy - NER (Named Entity Recognition)

Sua vez!

Se você estiver com o Spacy instalado na sua máquina tente aplicar o NER nas reclamações de uma empresa

**AGORA É
COM VOCÊ**

crianças brincam e toda parte e nunca vi esse tipo de atitude, pois acredito que isso de as crianças poderem experimentar as coisas é uma coisa legal, e meu filho ser chamado a atenção eu de verdade não entendo, e em segundo lugar e não menos importante uma pessoa se passar por gerente de loja, por que sei que toda loja tem um gerente geral e querer justificar que tinha placa se esse nem era o problema.

NUNCA MISC MAIS COMPRO EM NENHUMA LOJA MISC , E QUEM EU PUDEIR DIZER E CONVENCER ORG DE NÃO MISC COMPRAR FAREI ORG . Eu a LOJA MISC comprar uma luva de musculação, na qual a gondula está descrito melhor custo beneficio, os produtos estão todos misturados na gondula sem qualquer identificação para qual seria o produto de 24.99, quando fui ao caixa com o vale troca e um produto de mesma gondula , o caixa informou que haveria uma.diferenca a pagar ,mas o produto bfoi retirado desta gondula com.varios tipos de luva de musculação, sendo está preta com polegar. Tirei Loc uma foto e chamei um atendente no qual disse que a loja não tem culpa dos produtos estarem misturados pois a loja estava muito cheia, e não aceitou trocar o produto no qual havia comprado no dia anterior. A diferença era se 5.00 ,mas o que



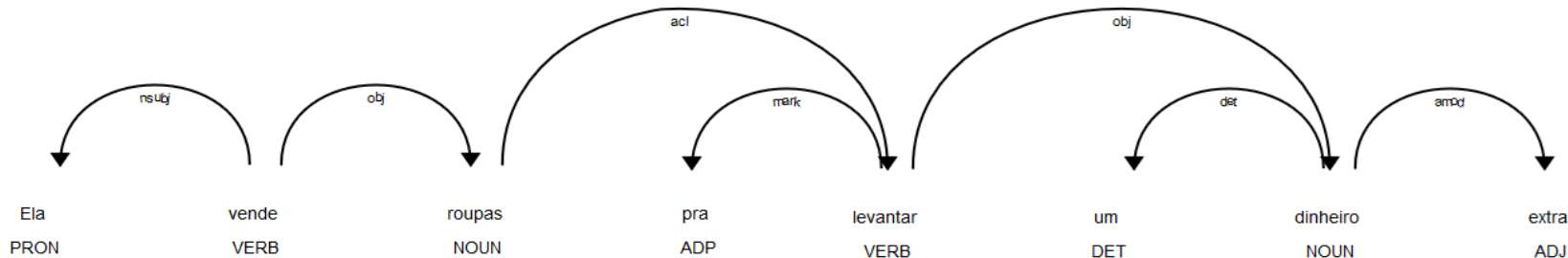
spaCy - POS Tagging (Part of Speech Tagging)

É a análise das classes gramaticais de um texto/frase.

Com ela é possível identificar os verbos, substantivos, adjetivos de uma frase.

Muito utilizada quando queremos gerar tradução automática de textos ou prever a próxima palavra (já que precisamos saber quais foram as últimas palavras antes da próxima e quais o contextos que elas estavam).

Exemplo:



Quer praticar mais?

List Comprehensions - <http://twixar.me/XM6K>

Sumarização de Textos - <https://bit.ly/2SMmVi4>

Spacy - <https://spacy.io/usage/linguistic-features>

POS - <http://twixar.me/qPCK>



Quem fez esse curso acontecer

Organizadoras

Deborah Froni

Linkedin - [in/deborah_froni/](https://www.linkedin.com/in/deborah_froni/)

Jessica Cabral

Linkedin - [in/jessica-cabral-carvalho/](https://www.linkedin.com/in/jessica-cabral-carvalho/)

Juliana Neves

Linkedin - [in/juliana-neves/](https://www.linkedin.com/in/juliana-neves/)

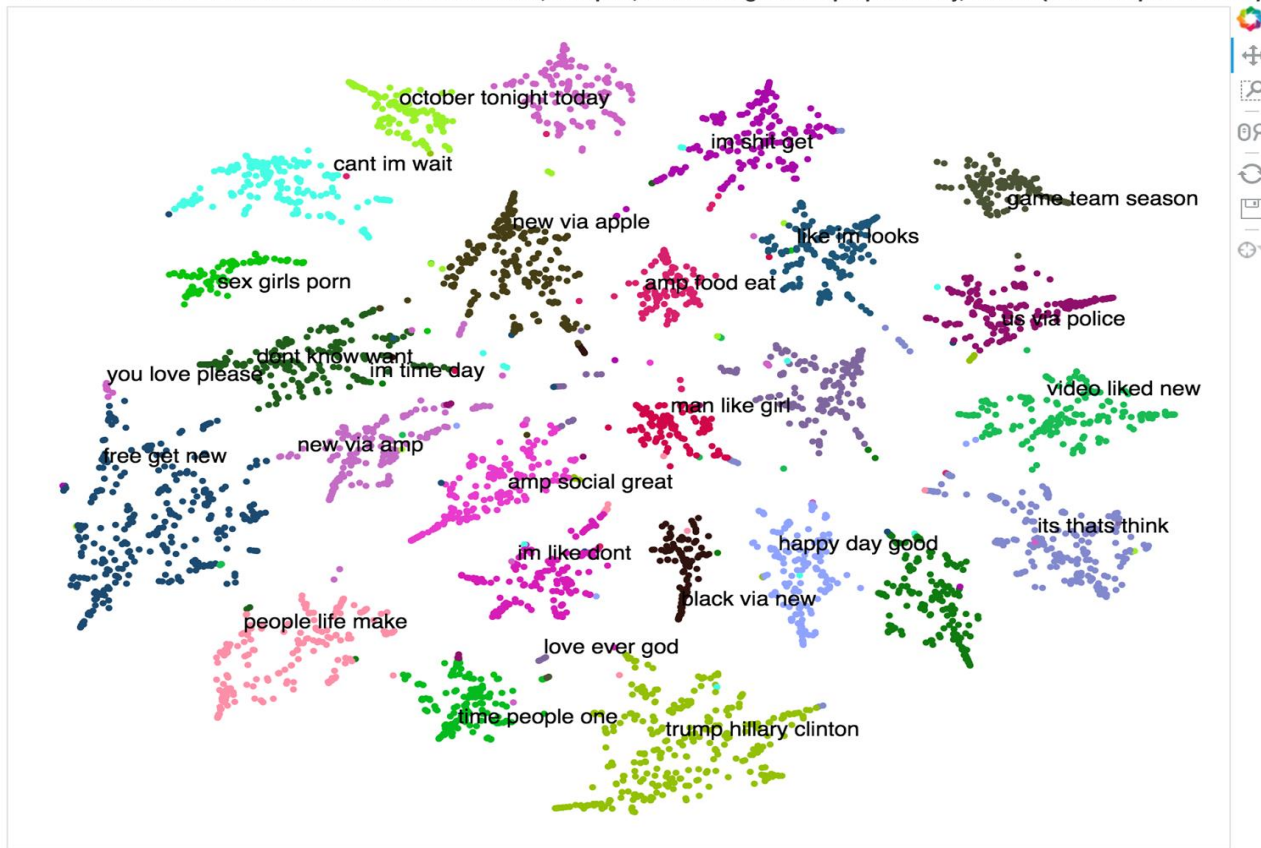


Referências bibliográficas

1. <http://www.eripi.com.br/2017/images/anais/minicursos/5.pdf>
2. <https://www.datacamp.com/community/tutorials/wordcloud-python>
3. <https://github.com/bsacash/Introduction-to-NLP/tree/master/1.%20Quick%20Python%20Refresher>
4. https://en.wikipedia.org/wiki/Named-entity_recognition
5. https://pt.wikipedia.org/wiki/Processamento_de_linguagem_natural
6. <https://code.nasa.gov/?q=python>
7. <https://www.datacamp.com/community/tutorials/stemming-lemmatization-python>
8. <https://towardsdatascience.com/python-list-comprehensions-in-5-minutes-40a68cbe4561>

Material de Apoio - Outras aplicações

t-SNE visualization of LDA model trained on 2000000 tweets, 25 topics, thresholding at 0.3 topic probability, 100 iter (5000 data points and top



**Segmentação de
Palavras/Documentos**

Material de Apoio - Outras aplicações

Entendimento da sintaxe do texto

✓ Dependency ✓ Parse Label ✓ Part of Speech ✓ Lemma ✓ Morphology

