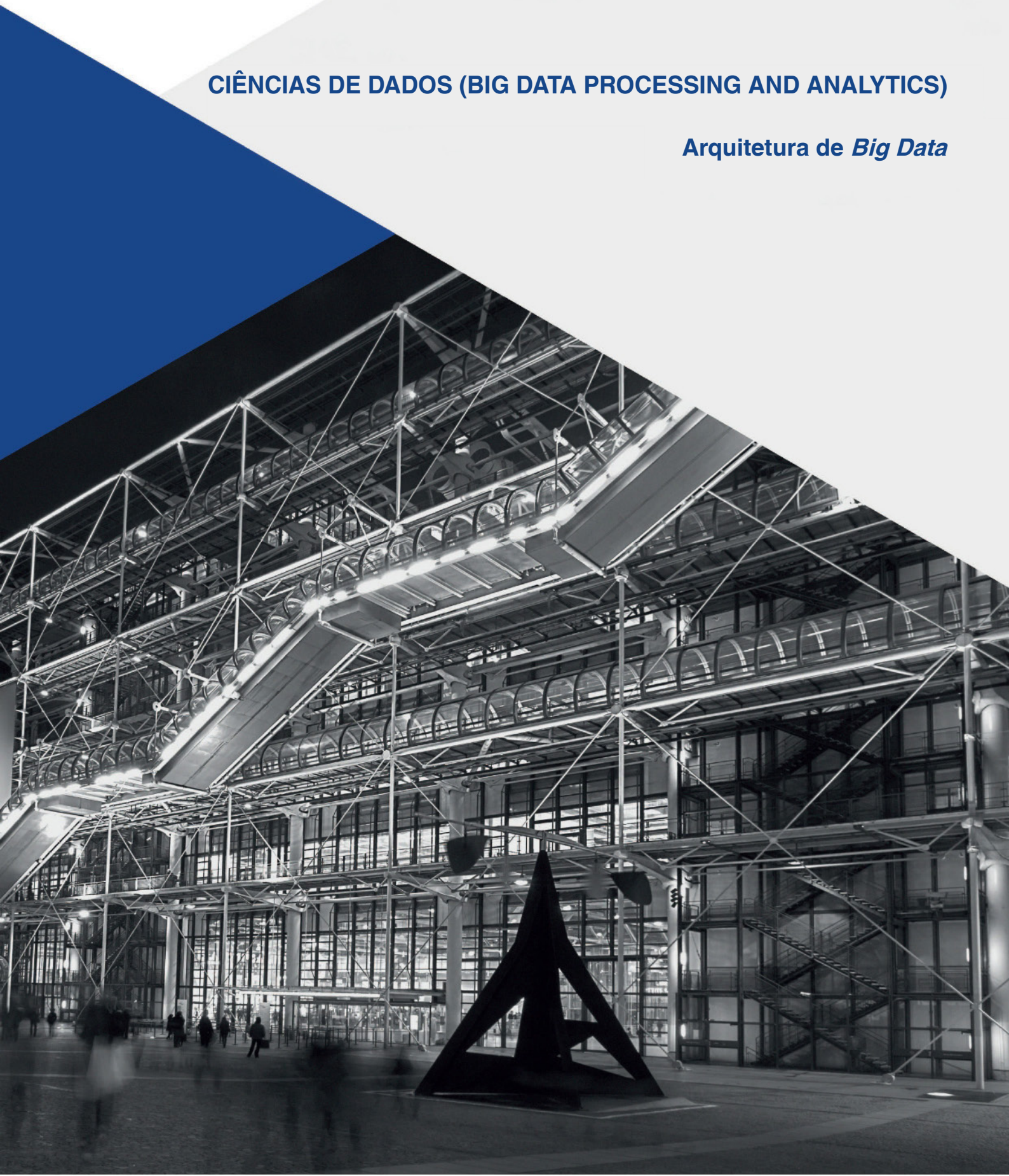
Universidade Presbiteriana Mackenzie

CIÊNCIAS DE DADOS (BIG DATA PROCESSING AND ANALYTICS)

Arquitetura de *Big Data*



Universidade Presbiteriana Mackenzie

Modalidade a distância

RECUPERAÇÃO DA INFORMAÇÃO NA WEB E EM REDES SOCIAIS

Trilha 5 – Processamento de Linguagem Natural: identificação de padrões de texto em documentos

Professor: Luciano Moreira Camilo e Silva

Sumário

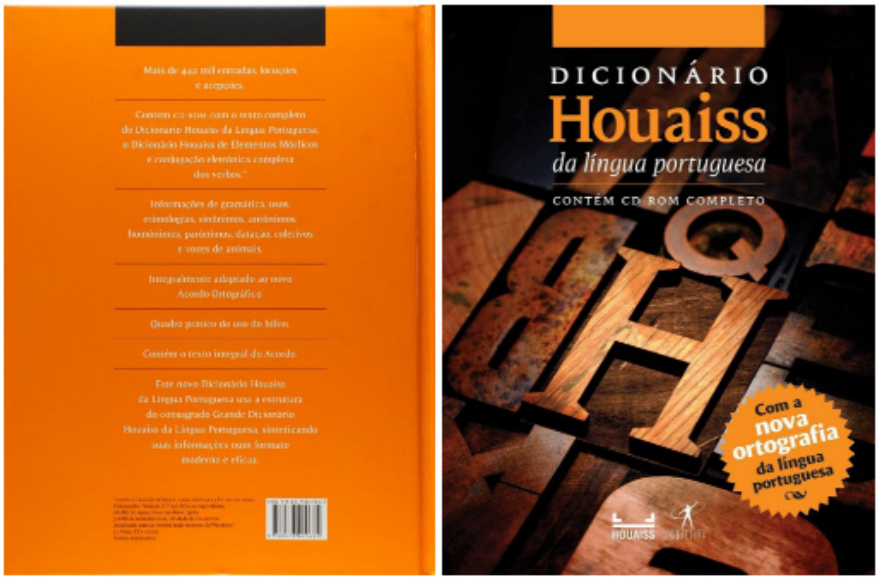
1. Introdução à Trilha

Uma das características que diferenciam os seres humanos dos demais animais é a capacidade de se comunicar por meio de um rico conjunto de palavras.

Identificar exatamente quantas palavras existem em cada idioma é uma tarefa muito complicada. Por exemplo, você contaria as palavras estudar, estudando, estudarão, estudado como quatro ou apenas uma palavra com diferentes flexões? E incluindo as palavras estudantes, estudioso, estudantil, que possuem o mesmo radical[1] de "estudar"? Contam-se como palavras novas, ou continua sendo a mesma palavra?

Buscando como referência o novo "Dicionário Houaiss da Língua Portuguesa", há 442 mil verbetes (HOUAISS, 2009). Outro dicionário extremamente popular no Brasil, o "Dicionário Aurelio da Língua Portuguesa", escrito pelo primo do compositor brasileiro Chico Buarque de Holanda e irmão do historiador Sergio Buarque de Holanda, Aurélio Buarque de Holanda Ferreira, identifica 435 mil termos em sua edição mais recente (FERREIRA, 2010).

Figura 1 – Capa e contracapa do dicionário Houaiss da língua portuguesa



1 "Radical é um morfema básico (que mostra o sentido básico da palavra), indivisível (porém existem palavras cujo radical se altera, como na conjugação de verbos anômalos) e comum a uma série de palavras. Também pode ser classificado como um morfema lexical." (Radical (linguística), Wikipédia).

Outros trabalhos tentam estimar a quantidade de palavras em outros idiomas, como este trabalho para o idioma inglês (BRYSBAERT et al., 2016), o qual não só buscou a quantidade de palavras (e lemas) conhecidos, como também estudou o aprendizado de novos vocábulos ao longo dos anos.

É importante notar que, embora idiomas como o português e inglês tenham centenas de milhares de palavras conhecidas, o uso no dia a dia exige bem menos conhecimento para estabelecer a comunicação.

Segundo Stuart Webb, professor de linguista aplicada da universidade de ***Western Ontario***, Canadá, em uma entrevista à rádio 4 da BBC (SAGAR-FENTON; MCNEILL, 2018), basta o conhecimento de 800 a 1.000 dos principais lemas[2] utilizados naquele idioma para você conseguir se comunicar com 75% de eficácia no dia a dia.

Como descrito na introdução do artigo de Chen et al. (2012),diferentes autores possuem diferentes estilos literários, e um meio comum de identificar esses estilos é pela frequência de uso das palavras mais comuns, como apresentado por John Burrows (2003).

A ideia para a Trilha 5 é continuar avançando na extração de palavras-chave, porém utilizar a vetorização de textos e combiná-la com análises estatísticas para tentar inferir o autor de um documento.

1.1. Caso prático

Hoje em dia é notório que a internet e a ***world wide web*** ganharam relevância no nosso cotidiano e revolucionaram a forma como lidamos com mídias.

Músicas, séries e filmes são consumidos por ***streaming***, sem a necessidade de comprar a mídia física ou mesmo baixar o arquivo digital. Notícias são reportadas quase em tempo real e levadas ao redor do mundo para que todos possam ler e comentar. E as redes sociais alavancaram o compartilhamento desses documentos em uma velocidade incrivelmente rápida, criando até uma nova palavra: viralizar.[3]

Uma das formas encontradas pelas pessoas ao compartilhar textos na internet é atribuir, muitas vezes falsamente, as declarações a pessoas que não o escreveram, porém possuem credibilidade, principalmente junto ao público-alvo almejado.

2 "O termo lema é usado para significar a forma básica de uma palavra, desconsiderando mudanças gramaticais como tempo verbal e pluralidade." (BIBER; CONRAD; REPPEN, 1998 apud LUCCA; NUNES, 2002).

3 Tornar viral, muito visto ou compartilhado por muitas pessoas, especialmente em redes sociais ou aplicativos de compartilhamento de mensagens.

O exercício pratico que acompanhará esta trilha avaliará textos erroneamente atribuídos ao cineasta e jornalista Arnaldo Jabor, já desmentidos pelo próprio articulista.

Será utilizado o conjunto de documentos (corpus) extraído para o exercício de ***web crawling***, seguido de extração de palavras-chave conforme exercício executado anteriormente para os anúncios classificados da OLX, procurando-se padrões nos textos.

Em específico, será identificada a distribuição de frequência das palavras mais usadas pelo autor, comparando-as com a distribuição de cada corpus associada ao autor.

Figura 2 – Arnaldo Jabor, março de 2008

Para o exercício, foram selecionados dois textos que circularam pelas redes sociais e que foram desmentidos como sendo do articulista, como pode ser validado em http://www.boatos.org, um site especializado em desvendar ***fake news***, ou mesmo ***hoaxes***.

Foram selecionados dois textos para se verificar a autenticidade, autointitulados:

1. Publicação de Arnaldo Jabor[4]

> Bolsonaro é um tipo de cara sem etiqueta, daqueles que encontramos coçando o saco no barzinho jogando bilhar. Apesar de ter mais estudo do que qualquer professor de humanas da geração Paulo Freire e mais inteligência emocional do que qualquer outro político brasileiro, não há polidez em suas palavras e tão pouco elegância em seu comportamento. Isso é o que mais incomoda artistas, jornalistas, feministas mal amadas e complexadas, homens frágeis, covardes oportunistas, religiosos falidos na luta contra a própria imoralidade, maconheiros,

4 Disponível em: <https://www.boatos.org/entretenimento/arnaldo-jabor-diz-bolsonaro-cara-sem-etiqueta-inteligente-voce-passa-gostar.html>.

> pedófilos, estupradores e toda patrulha do politicamente correto que suportou calada um circo de corrupção durante duas décadas, mas que agora é ferida com as palavras do presidente "não pudico". Bolsonaro é o milico com piadinhas sem graça, é o tiozão que pergunta se já temos pentelho, é um elefante em uma loja de cristais, mas o que me faz a cada dia gostar mais desse cara, é o tipo de gente que não gosta dele, que se ofende com tudo que o cara faz, que do óleo venezuelano em nossas praias à histeria mundial perante o coronavírus, buscam um meio de responsabilizá-lo. Bolsonaro realmente é o cara que você passa gostar, quando vê o lixo de gente que não gosta dele.

2. Crônica inteligente de Arnaldo Jabor[5]

> Brasileiro… Brasileiro é um povo solidário. Mentira. Brasileiro é babaca. Eleger para o cargo mais importante do Estado um sujeito que não tem escolaridade e preparo nem para ser gari, só porque tem uma história de vida sofrida; Pagar 40% de sua renda em tributos e ainda dar esmola para pobre na rua ao invés de cobrar do governo uma solução para pobreza;
>
> Aceitar que ONG's de direitos humanos fiquem dando pitaco na forma como tratamos nossa criminalidade… Não protestar cada vez que o governo compra colchões para presidiários que queimaram os deles de propósito, não é coisa de gente solidária. É coisa de gente otária.
>
> Brasileiro é um povo alegre. Mentira. Brasileiro é bobalhão. Fazer piadinha com as imundices que acompanhamos todo dia é o mesmo que tomar bofetada na cara e dar risada. Depois de um massacre que durou quatro dias em São Paulo, ouvir o José Simão fazer piadinha a respeito e achar graça, é o mesmo que contar piada no enterro do pai. Brasileiro tem um sério problema. Quando surge um escândalo, ao invés de protestar e tomar providências como cidadão, ri feito bobo.

Um terceiro texto, transcrito de um comentário que Arnaldo Jabor fez para a rádio CBN intitulado "Muita gente quer votar em Bolsonaro para se vingar" foi utilizado para validar a metodologia de autenticidade (JABOR, 2018)

> Amigos ouvintes, vocês assistiram aos debates dos jornalistas com os candidatos Jair Bolsonaro e Marina Silva. Não, pois perderam a chance de ver o absurdo que se avizinha no horizonte com nossos possíveis eleitos para presidência. É incrível. Primeiro Bolsonaro no programa Roda Viva foi um show de absurdos. Uma janelinha aberta para vermos a incompetência do homem. Creio também que os jornalistas, que são todos ótimos aliás, perderam a chance de fazer perguntas que decifrassem o enigma de sua utopia de caserna. O tempo todo ele se defendeu com a velha técnica dos reacionários sem razão que falam bem rápido, confusamente, de propósito para deixar a impressão de que algo foi respondido quando nada aconteceu. A maioria das perguntas tinha por fito provar que o Bolsonaro é

5 Disponível em: <https://www.boatos.org/brasil/arnaldo-jabor-brasileiro-babaca.html>.

> um despreparado, careta, preconceituoso, racista, etc. E perguntaram sobre aborto, mundo gay, estupro, relação com mulheres. Mas poderiam ter perguntado sobre o que planos ele teria pelo seu eventual governo. O povo que poderá votar nele está preocupado se ele gosta de gay ou não. Se aborto pode ou não pode. Ora francamente o povão bolsonarista quer denúncias violentas, ameaças machistas contra o crime que era uma espécie de trumpzinho de gericinó, dizendo que vai botar pra quebrar. E o maior perigo é que muita gente ignorante vai votar nele para se vingar do Brasil. Isso! Esse Brasil está em crise e o que muitos querem quebrar. De raiva. Isso é muito perigoso.
>
> Eu vi também a entrevista que a Marina Silva deu para outro grupo de jornalistas no dia seguinte. Meu Deus! Que coisa mais frágil! Que coisa mais pobre de ideias. Que coisa triste ver aquela boa mulher de ótimo caráter sem dúvida, mas sentadinha ali com sua vozinha falando em ética e ideias gerais sobre o país e visivelmente tentando encontrar um caminho entre um liberalismo maior e pequenas reverências aos petistas perdidos. Uma senhora sem força para ser presidente e um neonazista sem rumo. Não há um só programinha para o país dessa gente. Realmente é de gelar o sangue. O Brasil com candidatos risíveis, de chanchada, sem contar com os outros que vem aí como perigosíssimo Ciro Gomes. É triste pois como dizia um amigo meu o Brasil não tem pessoal.

2. Análise do corpus Arnaldo Jabor

Durante os estudos sobre ***web scraping*** na Trilha 3, foi construída uma aranha que fez a raspagem de todos os artigos produzidos por Arnaldo Jabor e publicados no jornal ***O Tempo*** de Minas Gerais.

Para carregar o corpus e, ao mesmo tempo, utilizar uma biblioteca bastante popular de manipulação de dados do Python, utilizamos o Pandas, carregando o arquivo com o comando abaixo (utiliza-se algo semelhante se o formato do arquivo for outro).

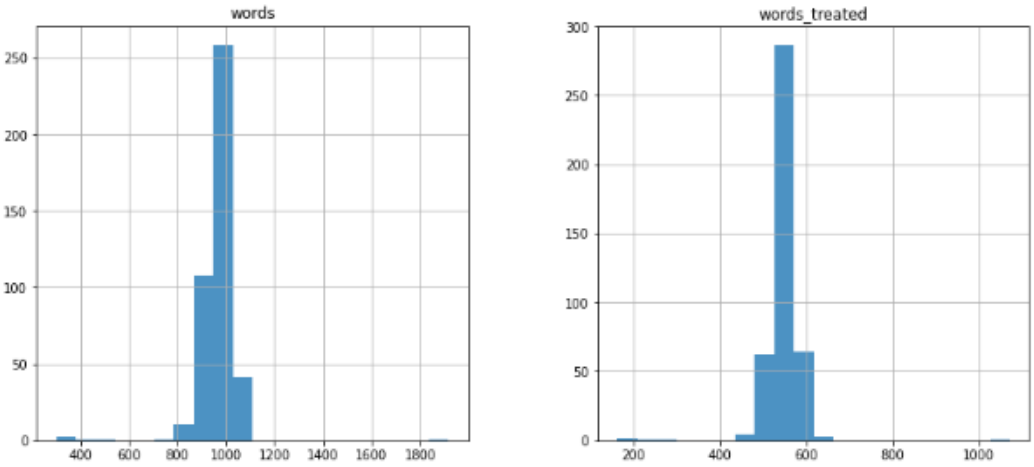
Caso esteja utilizando o ***Google Colab*** ou alguma outra biblioteca para edição de cadernos de ***Jupyter,*** lembre-se de fazer o carregamento do arquivo.

Uma primeira avaliação dos dados é feita contando as palavras e removendo documentos que podem ter sido carregados com problemas.

```
1.  #Carrega o texto em um DataFrame Pandas
2.  df_aj = pd.read_json("arnaldo_jabor.json")
3.
4.  #Faz a contagem de palavras por cada linha do artigo
5.  df_aj['words'] = df_aj['content'].str.split().str.len()
6.
7.  #Elimina qualquer artigo que tenha gerado menos de 100 letras
8.  df_aj = df_aj.loc[(df_aj['words'] > 100)]
9.
10. #Plota o DataFrame em histograma para uma avaliação
11. df_aj.plot.hist(bins=30, alpha=0.8)
```

Ao todo, estamos trabalhando com 424 artigos, com cerca de 1.000 palavras cada. São executados os mesmos tratamentos de limpeza de dados utilizados no exercício dos anúncios classificados da OLX, o que eliminou cerca de 50% das palavras.

Figura 3 – Distribuição de palavras por corpus



Fonte: Elaborada pelo autor.

2.1. Saco de Palavras

Uma das formas mais simples de representar documentos para o processamento de linguagem natural é por meio do método Saco de Palavras, ou Bag of Words, em inglês.

Assim como visto no exercício passado sobre codificações de textos, a ideia do saco de palavras é fazer uma codificação das frases. Cada palavra de um corpus é identificada, convertida em um número para associá-la mais facilmente, com menor consumo de memória nas estruturas de dados utilizados para seu armazenamento.

Para ilustrar, pegue como exemplo as próximas duas frases:

1. João adora as aulas de recuperação de informação. Maria adora também.
2. Maria também adora as aulas de machine learning.

Cada um dos dois documentos é representado pelo saco de palavras (já sem as palavras de parada):

1. {joao: 1, adora: 2, aulas:1, recuperacao: 1, informacao: 1, maria: 1, tambem: 1, machine: 0, learning: 0}
2. {joao: 0, adora: 1, aulas:0, recuperacao: 0, informacao: 0, maria: 1, tambem: 1, machine: 1, learning: 1}

Note que, na metodologia do saco de palavras, cada corpus é representado pela quantidade (contagem) de vezes que cada palavra aparece. A ordem das palavras, como evidenciada na frase 2, não é levada em consideração nessa metodologia, apenas a multiplicidade.

Uma das características do modelo de saco de palavras é que, ao concatenar dois documentos diferentes, basta somar os dois vetores que os representam. Assim, as frases "João adora as aulas de recuperação de informação. Maria adora também. Maria também adora as aulas machine learning" são representadas como:

> {joao: 1, adora: 3, aulas:1, recuperacao: 1, informacao: 1, maria: 2, tambem: 2, machine: 1, learning: 1}

Outra característica é a capacidade de representação matricial dos corpora em saco de palavras. Ainda para o exemplo acima, teríamos a tabela abaixo:

Tabela 1 – Matriz representando o Saco de Palavras do exemplo acima

	joão	adora	aulas	recuperação	informação	maria	tambem	machine	learning
Doc 1	1	2	1	1	1	1	1	0	0
Doc 2	0	1	0	0	0	1	1	1	1

Fonte: Elaborada pelo autor.

Em um caso real, a quantidade de palavras existentes na corpora é muito maior e seria impossível representá-la visualmente como na Tabela 1. O vocabulário dos artigos do Arnaldo Jabor é de 28.080 palavras.[1]

```
1.  from sklearn.feature_extraction.text import CountVectorizer
2.  vectorizer = CountVectorizer()
3.
4.  #Arnaldo Jabor
5.  XAJ = vectorizer.fit_transform(df_aj['content_treated'])
6.  vocabulary_aj = vectorizer.get_feature_names()
7.  pdXAJ = pd.DataFrame(data=XAJ.toarray(), columns=vocabulary_aj)
8.
9.  #Arnaldo Jabor - Verificar
10. XVER = vectorizer.fit_transform(df_ver['content_treated'])
11. vocabulary_ver = vectorizer.get_feature_names()
12. pdXVER = pd.DataFrame(data=XVER.toarray(), columns=vocabulary_ver)
```

1 Você pode consultar o tamanho por meio da variável vocabular_aj e do comando len(vocabular_aj).

2.2. Palavras mais utilizadas

Uma das formas de identificar se um autor, como mostrado em Burrows (2003), escreveu ou não um texto é avaliar as palavras mais utilizadas por ele. Os escritores possuem uma tendência natural de repetir as mesmas palavras em seus textos. Essa característica, como ressaltada por Chen et al. (2012)faz parte de seu estilo literário.

Evidentemente, existem outros fatores que podem e devem ser avaliados em um estudo forense. Para o intuito deste exercício, examinaremos apenas as palavras mais usadas.

A Tabela 2 mostra as 50 palavras mais utilizadas pelo articulista. Para quem acostumou-se a lê-lo e ouvi-lo, é possível até ouvir uma voz em sua cabeça falando algumas dessas palavras.

Tabela 2 – 50 palavras mais utilizadas por Arnaldo Jabor

Palavra	#	Palavra	#	Palavra	#	Palavra	#	Palavra	#
tudo	1141	bem	642	nunca	453	lula	382	diante	356
ser	1112	sempre	637	vai	452	amor	381	dentro	347
mundo	1022	pois	577	tempo	446	la	379	poder	342
hoje	972	porque	576	ai	443	futuro	378	fim	337
vida	806	contra	557	todos	426	sim	371	mulher	335
país	790	sobre	543	política	425	vez	369	novo	334
anos	735	onde	506	ainda	420	ninguém	366	apenas	330
nada	725	mal	483	outro	418	coisa	363	ver	328
grande	715	dia	471	pode	417	filme	361	ali	327
brasil	686	agora	470	assim	408	morte	357	cinema	318

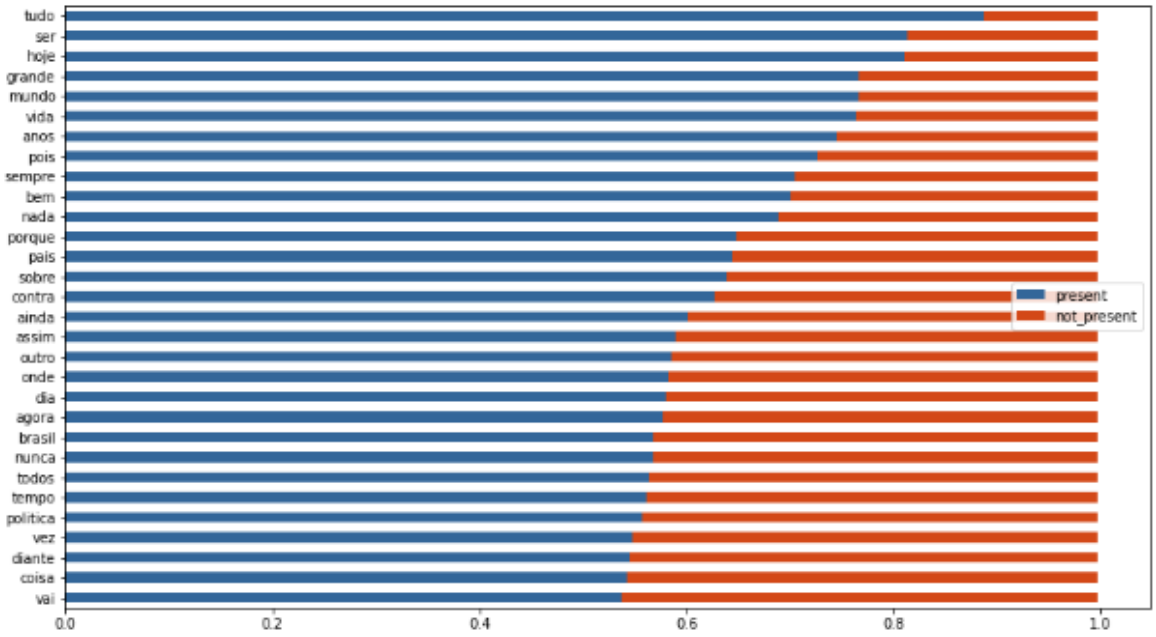
Fonte: Elaborada pelo autor.

Porém, analisar apenas as palavras mais utilizadas, mesmo sendo uma excelente forma de identificar preferências do autor, pode levar a falsos resultados quando elas não estão bem distribuídas entre todos os artigos.

Uma forma de analisar o uso de palavras mais frequentes e tratando um possível problema de repetição de palavras em um ou poucos documentos, é recalculando as palavras por percentual de artigos que as contêm.

```
# Transpõe a matriz para que as palavras fiquem como linhas e o documento, como colunas
pdXAJT = pdXAJ.T

#Calcula a quantidade de documentos em que aquela palavra entrou ou não entrou
pdXAJT['present'] = (pdXAJT.ne(0).sum(axis=1) -1) / len(df_aj['content'])
pdXAJT['not_present'] = pdXAJT.eq(0).sum(axis=1) / len(df_aj['content'])

#Remove as colunas individuais de cada documento para gerar o gráfico
pdXAJT.drop(pdXAJT.columns[0:len(df_aj['content'])], axis=1, inplace=True)

#Ordena o resultado final
pdXAJT = pdXAJT.sort_values(by='present', ascending=True)

#Gera o gráfico contendo as 30 palavras mais usadas
pdXAJT.tail(30).plot.barh(stacked=True, color=['#336699', '#D34817'], figsize=(14,8))
```

Figura 4 – Percentual da presença de cada palavra nos textos do Arnaldo Jabor

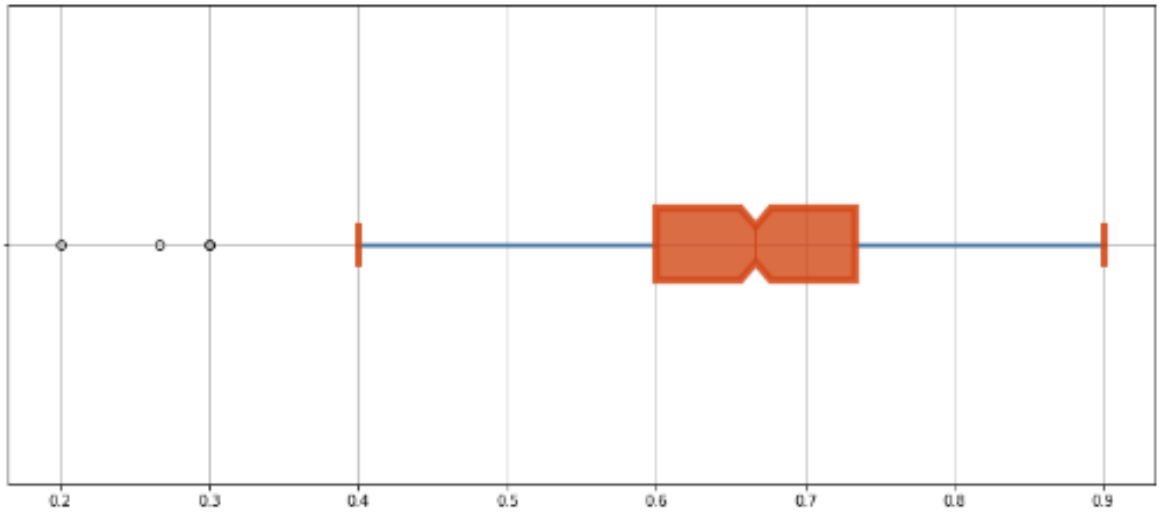


Fonte: Elaborada pelo autor.

A próxima etapa é avaliar o percentual de palavras pertencentes a esse conjunto de 30 palavras mais usadas presente nos 424 documentos do corpora. Da nossa amostra, apenas quatro documentos foram considerados ***outliers,***[2] totalizando menos de 1% da amostra. Observa-se que todos os textos do Arnaldo Jabor utilizam entre 37% e 87% das 30 palavras verificadas como mais frequentes.

2 "Um outlier é uma observação que se diferencia tanto das demais observações que levanta suspeitas de que aquela observação foi gerada por um mecanismo distinto, em outras palavras os outliers são dados que se distanciam radicalmente de todos os outros, valores que fogem da normalidade e que podem causar desequilíbrio nos resultados obtidos." (Outlier, Wikipédia).

Figura 5 – Percentual de uso das 30 palavras mais frequentes dos artigos de Arnaldo Jabor



Fonte: Elaborada pelo autor.

O código utilizado para gerar o ***boxplot***[3] pode ser visto abaixo:

```
#Reinicia a matriz transposta
pdXAJT = pdXAJ.T

#Recalcula tudo (para evitar problemas como ordem de execução das células)
pdXAJT['present'] = (pdXAJT.ne(0).sum(axis=1) -1) / len(df_aj['content'])
pdXAJT = pdXAJT.sort_values(by='present', ascending=False)
pdXAJT.drop('present', axis=1, inplace=True)

#Cria uma lista com as 30 palavras mais utilizadas
most_commons = list(pdXAJT.head(30).index.array)

#Filtra as 30 palavras e transpõe novamente
pdXAJT = pdXAJT.filter(items=most_commons, axis=0).T

# Cria novas colunas com o percentual das 30 palavras mais utilizadas
# presentes em cada corpus
pdXAJT['present'] = (pdXAJT.ne(0).sum(axis=1)) / 30

#Remove as colunas individuais de cada palavra para gerar o gráfico
pdXAJT.drop(pdXAJT.columns[0:30], axis=1, inplace=True)

#Gera o gráfico de boxplor
pdXAJT.boxplot(vert=False, figsize=(14,6), notch=True, patch_artist=True,
   boxprops=dict(linestyle='-', linewidth=5, color='#D34817DD', facecolor='#D34817CC'),
   medianprops=dict(linestyle='-', linewidth=2,  color='#D34817DD'),
   whiskerprops=dict(linestyle='-', linewidth=3, color='#336699DD'),
   capprops=dict(linestyle='-', linewidth=5,  color='#D34817DD')
)
```

3 "Em estatística descritiva, boxplot é uma ferramenta gráfica para representar a variação de dados observados de uma variável numérica por meio de quartis." (Diagrama de caixa, Wikipédia).

Agora que foi calculada a base de comparação, aplicaremos o mesmo cálculo nos textos que gostaríamos de verificar. Para isso, seguimos as mesmas etapas nos documentos a terem a autoria validada.

Tabela 3 – Presença das 30 palavras mais comuns nos textos analisados

	tudo	ser	hoje	grande	mundo	vida	anos	pois	sempre	bem	nada	porque	país	sobre	contra	ainda	assim	outro	onde	dia	agora	brasil	nunca	todos	tempo	política	vez	diante	coisa	vai	% uso
Texto 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	17%
Texto 2	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	2	0	23%
CBN	0	1	0	0	1	0	0	2	0	1	1	0	2	3	1	0	0	1	0	1	0	4	0	1	1	0	0	0	3	2	50%
Esperado	89%	81%	81%	77%	77%	76%	75%	73%	71%	70%	69%	65%	64%	64%	63%	60%	59%	58%	58%	58%	58%	57%	57%	56%	56%	56%	55%	54%	54%	54%	62%

Fonte: Elaborada pelo autor.

Analisando os textos frente ao histórico do articulista, podemos concluir que tanto o Texto 1 quanto o Texto 2 não possuem as características presentes em artigos. A transcrição da rádio CBN, sabidamente de autoria de Arnaldo Jabor, comportou-se como esperado e está na faixa da Figura 5.

3. Outras técnicas comuns em PLN

Apesar de não ter sido utilizado para a avaliação de autenticidade dos textos de Arnaldo Jabor, duas outras técnicas são bastante utilizadas em pré-processamento, ou mesmo na etapa de extração de características de textos.

Como sempre, a aplicabilidade dessas técnicas depende principalmente do caso de uso, mas também de outros fatores, como tamanho e qualidade do corpora.

As duas principais técnicas de PLN não abordadas até o momento estão descritas nos próximos itens: são o pré-processamento de texto utilizando ***steming*** de palavras e a vetorização TF-IDF em alternativa ao saco de palavras.

3.1. Steming

Recorde o que foi dito na Introdução à trilha acerca das palavras estudantes, estudantil, estudados entre tantas outras variações. São palavras com a mesma origem: ***Studiare,*** do latim medieval, que, por sua vez, tem origem na palavra latina ***studium*** e significa “esforço”, “diligência” que todos os estudantes têm ou deveriam ter.

Em alguns casos, diferenciar essas palavras não agrega ao resultado ou mesmo atrapalha. No entanto, caso você esteja buscando um conjunto de textos que versa sobre estudantes em geral, você excluiria aqueles que contêm a palavra “estudantil”? Se você quer ler sobre democracia, excluiria documentos que contivessem as palavras “democrático” e “democratismo”?

Encontrar o lema de uma palavra é um processo muito complicado e, em 1980, o professor Martin Porter publicou seu algoritmo de ***steming***. O algoritmo fazia o tratamento das palavras, removendo caracteres à direita, tentando chegar em um radical comum. Por exemplo, as inflexões do verbo estudar, estudado, estudando etc., utilizadas neste texto como exemplo, são reduzidas a estud.

A vantagem de utilizar o algoritmo de ***steming*** no lugar de um algoritmo mais robusto de lematização está associada à facilidade e velocidade do primeiro. Além disso, resultados muito promissores foram obtidos com o uso do ***steming*** de Porter e as variações criadas ao longo dos anos (WILLETT, 2006).

Além disso, outra vantagem é reduzir o tamanho das matrizes envolvidas no saco de palavras. Veja o caso dos 424 textos de Arnaldo Jabor analisado na seção 2 Análise do

corpus Arnaldo Jabor. Havia 28.080 palavras, já desconsideradas as palavras de paradas, pontuação e números. Com ***steming*** reduz-se a 12.637, uma baixa de mais de 50%.

Steming, por ser um algoritmo simplificado, está sujeito a erros que, dependendo do caso, podem impactar o resultado. Sem dúvidas, há vários casos de uso que se beneficiam dessa técnica e, como sempre, precisam ser avaliados em cada caso de uso.

Quando se vai utilizar algum algoritmo em que é necessário repetir o cálculo diversas vezes em cima da mesma matriz, essa economia pode impactar significativamente o desempenho, como nos casos do ***machine learning***.

Tenha em mente, porém, que se, por um lado, o ***steming*** diminui a dispersão de inflexões de uma mesma palavra aumentando a sensibilidade dos seus algoritmos, por outro, pode-se impactar bastante na precisão dos seus cálculos.[1]

Para fins ilustrativos, se o processo de validação de autenticidade de Arnaldo Jabor tivesse utilizado de ***steming***, todas essas palavras com significados diferentes (parou, paro, pariu, pariram, parir, parindo, paridos, pares, parem, parei, pare, paravam, parava, pararia, pararam, parar, param, parados, parado, paradas, parada, para, par) seriam transformadas no radical par. Com certeza, impactando na precisão da análise.

3.2. TF-IDF

Imagine a situação em que se quer classificar matérias de um periódico futebolístico (ou, ainda, um agregador de notícias de futebol) e separar quais matérias são mais relevantes para cada torcedor.

Para treinar o modelo, provavelmente, serão utilizados centenas ou milhares de artigos previamente classificado por time. Possivelmente, entre as palavras mais frequentemente utilizadas estarão jogo, tempo, gol, pênalti, primeiro e segundo [tempo] e minutos. Todas essas palavras são comuns no jornalismo esportivo que trata de futebol. Mas elas discriminam muito pouco o que um palmeirense ou um torcedor do tricolor de aço gostaria de acompanhar.

Uma das formas de tratar a relevância de uma palavra para um documento é calculando o coeficiente TF-IDF.

1 "Precisão é a fração de instâncias recuperadas que são relevantes, enquanto sensibilidade é a fração de instâncias relevantes que são recuperadas." (Precisão e revocação, Wikipédia).

Equação 1 – Fórmula TF-IDF

$$tfidf(t, d, D) = tf(t, d) * idf(t, D)$$

Onde,

t é o termo avaliado, *d* é o documento avaliado e *D* é o conjunto de todos os documentos.

3.2.1 Frequência de termos (Term-Frequency)

É presumível que um palmeirense se interesse por matérias esportivas que contenham as palavras palmeiras, palestra, verdão etc. Essas palavras, contudo, podem aparecer, inclusive, em matérias de rivais, como em uma reportagem sobre a preparação do adversário para enfrentar o "campeão do século [XX]".

Possivelmente, tal matéria terá uma ou duas ocorrências da palavra palmeiras. Por outro lado, a mesma partida comentada para agradar o torcedor alviverde reunirá diversas citações ao time.

Apenas contar as palavras pode levar a outra má interpretação. Uma matéria sobre a história do "Sport Club Corinthians Paulista" deverá ter uma boa contagem do termo palmeiras, porém diluída em centenas de frases de uma longa e bem escrita matéria jornalística.

De forma a melhorar a relevância dos termos na avaliação do processamento da linguagem natural, usa-se um cálculo de frequência de termos, primeiramente citado pelo renomado pesquisador de ***hashs*** Hans Peter Luhn (1975).

A fórmula pode ser vista na Equação 2. O cálculo é bastante simples, apesar de poderoso, para discernir palavras mais relevantes no documento. Apenas divide-se a quantidade de vezes em que uma palavra apareceu em um documento pela quantidade de palavras existentes do documento. Isso após o pré-tratamento.

Equação 2 – Fórmula TF

$$tf(t_n, d) = \frac{contar(t_n)}{\sum contar(t_i \in d)} = \textbf{Onde,}$$

Onde,

t_n é o termo avaliado, t_i são todos os termos existentes no documento d.

3.2.2 Frequência inversa do Documento (Inverse Document Frequency)

Frequência inversa do documento foi um conceito criado pela pioneira da computação britânica Karen Spärck Jones, em um artigo que pode ser traduzido para "Uma interpretação estatística da especificidade do termo e sua aplicação na recuperação da informação" (JONES, 1972).

A ideia da equação é endereçar o problema citado anteriormente das palavras futebol, tempo e gol em artigos esportivos de futebol. Essas palavras, por estarem presentes em grande parte dos textos disponíveis e classificados para todos os torcedores, acabam por não discernir o que cada um quer ler.

Calcula-se a quantidade de documentos distintos que possuem aquela palavra, ou seja, $tf\ (t_n,d) \neq 0$. Se todos os artigos possuem a palavra gol, ela não é relevante para o nosso contexto. A palavra não agrega semanticamente em nossa recuperação de informação.

O índice é calculado dividindo-se o total de documentos do corpora pela quantidade de documentos que possuem aquela palavra, muito semelhante ao que fizemos na Tabela 3 do exercício do Arnaldo Jabor para a linha esperada. Porém, aqui, faremos o cálculo inverso. Ao invés de dividir as 376 vezes que a palavra tudo apareceu nos 424 documentos, dividimos os 424 documentos pelas 376 aparições.

Por fim, há mais duas considerações na Equação 3 que aparece no final dessa seção. Para evitar divisão por zero, quando se tenta calcular o IDF de uma palavra que não apareceu em nenhum documento, é comum adicionar uma unidade no denominador da equação.

E, por fim, ajusta-se a escala do computo aplicando logaritmo. Como o objetivo é calcular relevância, a ideia é não valorizar excessivamente documentos relevantes. Afinal, 200 ou 100 aparições de uma palavra já é relevante, e não precisamos ponderar com o dobro de peso o segundo termo.

Equação 3 – Fórmula IDF

$$idf\,(t_n,\ D) = log\frac{|\{d \in D\}|}{|\{d \in D\ :t \in d\}|+1}$$

Onde,

t é o termo avaliado, *d* é o documento avaliado e *D* é o conjunto de todos os documentos.

4. Síntese

Muita coisa foi discutida nesse material e é fundamental que se busque exercitar o que foi estudado, não apenas para o caso apresentado dos textos do Arnaldo Jabor, mas também para outras situações.

Trabalhar com processamento de linguagem natural, como em várias outras áreas das ciências, exige prática. E diversos assuntos foram tratados.

Iniciou-se com uma avaliação das palavras extraídas de diversos artigos, onde se aprendeu que, a partir da análise dos termos usados, é possível identificar a autenticidade ou, no mínimo, suspeitar da mesma em textos. Já pensou em analisar outros autores? Textos históricos? Analisar seus próprios documentos?

Ainda que não tenha sido o ponto central, utilizamos várias bibliotecas do Python, como o Pandas, NLTK e SKLearn. Gaste um tempo para se familiarizar com essas bibliotecas.

Por fim, foram apresentados dois conceitos que podem ser utilizados em projetos distintos dentro da disciplina de recuperação de informação: steming e tf-idf. Qual projeto você quer fazer para aprender e utilizar na prática esses conceitos? Quais são suas ideias?

Lembre-se sempre de soltar sua imaginação. Como disse Albert Einstein, ela é o verdadeiro sinal da inteligência.

5. Referências

BIRD, Steven; KLEIN, Ewan; LOPER, Edward. ***Natural Language Processing with Python***. Sebastopol, CA: O'Reilly Media., 2009.

BRYSBAERT, M. et al. How many words do we know? Practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. ***Frontiers in Psychology***, 29 jul. 2016. Disponível em: <https://www.frontiersin.org/articles/10.3389/fpsyg.2016.01116/full>. Acesso em: 26 out. 2021.

BURROWS, J. Questions of authorship: attribution and beyond: a lecture delivered on the occasion of the Roberto Busa Award ACH-ALLC 2001. ***Computers and the Humanities***, p. 5-32, 2003.

CHEN, Z. et al. More than word frequencies: authorship attribution via natural frequency zoned word distribution analysis. ***ArXiv***, 2012.

FERREIRA, A. B. H. ***Dicionário Aurélio da Língua Portuguesa***. 5. ed. São Paulo: Positivo, 2010.

HOUAISS, A. ***Dicionário Houaiss da Língua Portuguesa***. Rio de Janeiro: Objetiva, 2009.

JABOR, A. Muita gente quer votar em Bolsonaro para se vingar do Brasil. ***O Comentário de Arnaldo Jabor***, 2 ago. 2018.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. ***Journal of Documentation***, p. 11-21, 1972.

LUCCA, J. L.; NUNES, M. G. V. Lematização versus Stemming. ***Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional***, NILC, ICMC-USP, 2002.

LUHN, H. P. A statistical approach to mechanized encoding. ***IBM Journal***, p. 309-317, out 1957.

PORTER, M. F. An algorithm for suffix stripping. ***Program: electronic library and information systems***, p. 130-137, 1980.

SAGAR-FENTON, B.; MCNEILL, L. How many words do you need to speak a language? ***More or Less, BBC Radio 4***, Londres, 24 jun. 2018.

SILVA, L. M. C. Recuperação de informação por raspagem: robôs. *Recuperação de informação na web e em redes sociais*, v. 3, 2021.

WILLETT, P. The Porter stemming algorithm: then and now. *Program*, 2006.