Ciência de Dados:
Percurso Inicial para Tratamento do Dataset CORD-19

Jhonatan S. Gomes[1], **Eonay B. Gurjão**[1], **Klessis L. Dias**[1], **Klenilmar L. Dias**[1]

[1]GPTICAM – Grupo de Pesquisa em Tecnologias da Informação e Comunicação
na Amazônia – Instituto Federal do Amapá (IFAP)
Caixa Postal 68.900 – 398 – Macapá – AP – Brazil

jhonatan.dsgomes@gmail.com

{eonay.gurjao, klessis, klenilmar.dias}@ifap.edu.br

Abstract. *This paper focuses on presenting a preliminary path to the initial step of data processing the CORD-19 dataset, applying a few data science techniques based on Python science libraries.*

Resumo. *Este artigo se concentra em apresentar um percurso preliminar para a fase inicial de tratamento do dataset CORD-19, aplicando algumas técnicas de ciência de dados baseado em bibliotecas científicas do Python.*

1. Introdução

Uma pandemia de COVID-19 já provou ser um desafio global. Também mobilizou pesquisadores de diferentes ciências e de diversos países na busca de uma forma de combater essa doença potencialmente fatal. Como resposta à pandemia mundial, uma grande quantidade de literatura científica relacionada ao COVID-19 surgiu e continua a crescer rapidamente [Shuja et al. 2020]. Como resultado, o COVID-19 Open Research Dataset Challenge lançou o CORD-19[1], um *corpus* de artigos acadêmicos sobre COVID-19, SARS-CoV-2 e o grupo Coronavirus. O CORD-19, é um dataset com artigos de todo os países do mundo, composto de 500.000 artigos acadêmicos, incluindo mais de 200.000 com texto completo sobre o COVID-19 ou doenças relacionadas (última atualização do corpus em 25.05.2021).

Modelos de Aprendizado de Máquina (Machine Learning - ML) também desempenham um papel importante [Lalmuanawma et al. 2020]. A aplicação de modelos de ML nos datasets existentes, podem auxiliar no desenvolvimento de aplicações para combater não apenas a atual pandemia do COVID-19, mas também para combater a disseminação de doenças infecciosas no futuro. Usar ML permite que aplicações inteligentes possam fazer a categorização de artigos em vários subtópicos, tais como: precaução e evolução da doença [Afzal et al. 2020]. Contudo, datasets coletados de várias fontes, estão sujeitos a informações incompletas que irão gerar incertezas durante a análise dos dados e poderá afetar a precisão dos modelos de ML [Ridzuan and Zainon 2019]. A limpeza de dados oferece uma melhor qualidade nos dados dos datasets. É uma operação realizada nos dados existentes para garantir que seu dataset esteja pronto para as outras fases do processo de ML[2]. O processo de limpeza de dados é complexo e consiste em várias etapas.

[1]https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

[2]Extração e Seleção de Características, Treinamento do Modelo, Avaliação do Modelo e Implantação e Integração do Modelo

O objetivo deste artigo é apresentar os primeiros esforços no desenvolvimento de um projeto de Ciência de Dados aplicado a um Dataset do COVID-19. Especificamente, neste trabalho, será apresentado um percurso preliminar para a fase inicial de limpeza e ordenação de dados para remover erros ou valores ausentes no dataset CORD-19.

2. Metodologia

Este artigo é composto de um dataset com 68.204 amostras de artigos acadêmicos, retiradas de maneira randômica do datset CORD-19. Este fatiamento foi necessário para atender aos recursos computacionais disponíveis no momento da concepção do artigo. Estas amostras foram rotuladas com quatro características (features): paper_id, title, abstract, text.

Para esta fase inicial, fase de tratamento e análise do dataset, foram utilizados alguns pacotes do SciPy[3], que é um ecossistema de softwares baseado em Python, de código aberto para matemática, ciências e engenharia. A lista dos pacotes utilizados, assim como as descrições de cada um deles, estão descritos na Tabela 1.

Tabela 1. Pacotes do SciPy utilizados.

Bibliotecas Científicas	
Pacote	**Descrição**
NumPy	Biblioteca científica para manipular vetores e matrizes
zipfile	Biblioteca para acessar e descompactar arquivos no formato .zip
pandas	Biblioteca para acessar o dataset (manipular de forma rápida e expressiva, dados estruturados)
glob	Biblioteca para fazer leitura do disco
json	Os artigos do dataset estão no formato json
Seaborn	Biblioteca de visualização de dados baseada em matplotlib
spaCY	Biblioteca para Processamento de Linguagem Natural
scispaCY	Biblioteca para processamento de texto biomédico, científico ou clínico
NLTK	Biblioteca para Processamento de Linguagem Natural
Matplotlib	Biblioteca para criação de gráficos e visualização de dados em geral

Como ferramenta foi utilizado a distribuição 4.8.3 do Anaconda[4], que é uma plataforma de código aberto que reuni em um só arquivo diversas bibliotecas científicas de Python, incluindo todas da Tabela 1, focadas em Ciência de Dados. Além de instalar e configurar o Python no ambiente Linux, ela disponibiliza outras ferramentas. E uma delas é o Jupyter Notebook, que foi utilizado como ambiente interativo de desenvolvimento neste artigo.

3. Resultados

Seguindo a metodologia descrita foi gerado um Notebook, possibilitando dividir cada etapa de código em blocos independentes chamados células, para aplicar os passos iniciais. As principais células de código e com suas respectivas descrições são apresentadas a seguir.

[3]https://www.scipy.org/

[4]https://www.anaconda.com/products/individual#Downloads

1. Instalação e Importação dos Pacotes

```
import numpy as np # Biblioteca cientifica para trabalhar com vetores e matrizes
import zipfile # Biblioteca para acessar e descompactar arquivos no formato .zip
import pandas as pd # Biblioteca para acessar dataset
import glob # Biblioteca para fazer leitura do disco
import json # Os artigos da base de dados estão no formato json
import seaborn as sns # Biblioteca que gera gráficos
import spacy # Biblioteca para Processamento de Linguagem Natural
import scispacy # Biblioteca para processamento de texto biomédico
import nltk # Biblioteca para Processamento de Linguagem Natural
from IPython.core.display import HTML # Biblioteca para gerar arquivos HTML dentro do console
from matplotlib import pyplot as plt # Biblioteca para gerar gráficos
```

2. Criação do Dataframe com os Textos do dataset

```
# Criando um Dicionário e definindo quais colunas ("chave") serão acessadas nos arquivos json do dataset, que são:
# "paper_id"; "titulo"; "abstract"; "texto"
# Para isso cria-se um dicionário para essas colunas (que são as features)
# {chave:valor} - os valores estarão vazio nesse momento da criação do dicionário

corona_features = {'paper_id': [], 'title': [],
                   'abstract': [], 'text': []}
```

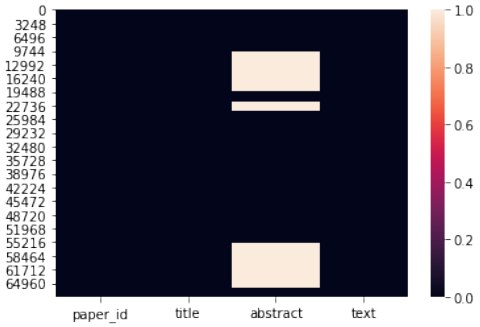
```
# Transformando o dicionario "corona_features" para o formato de um dataframe
# Cria-se o dataframe (chamado de corona_df) no formato do Pandas
corona_df = pd.DataFrame.from_dict(corona_features)
```

```
# Agora vamos percorrer os arquivos json para popular esse dataframe com cada um dos registros
# Para isso é criado uma lista (json_filenames) para receber inicialmente os registros do dataset

json_filenames = glob.glob(f'{"/Path do Local onde se encontra o dataset CORD-19"}//**/*.json', recursive = True)
```

3. Pré-Processamento dos Textos

```
# Verificando valores faltantes (NaN) usando recursos do pacote "Seaborn"
# Aplicando a função "heatmap" que gera um gráfico em formato de mapa de calor
sns.heatmap(corona_df.isnull()); # A função "isnull" retorna os registros que não possuem valores
```




```
# Verificando valores com espaço vazio (vazio é diferente de Nulo (NaN))
# Resposta = 0 indica que todos os artigos possuem valores para "paper_id", "title", "abstract", "text"
# Resposta <> 0 indica que existem artigos que não possuem valores para "paper_id", "title", "abstract", "text"
len(corona_df[corona_df['paper_id'] == '']) # (resposta = 0)
len(corona_df[corona_df['title'] == ''])    # (resposta = 4997)
len(corona_df[corona_df['abstract'] == '']) # (resposta = 12949)
len(corona_df[corona_df['text'] == ''])     # (resposta = 2375)
```

```
# Removendo registro do tipo vazio('')
# Fazendo o filtro para os artigos onde o "title", "abstract", "text" são diferentes de vazio
corona_df = corona_df[corona_df['title'] != '']
corona_df = corona_df[corona_df['abstract'] != '']
corona_df = corona_df[corona_df['text'] != '']
```

```
# Verificando o shape após a remoção de registros vazio
corona_df.shape
```

```
(51636, 4)
```

```
# Removendo registro do tipo Nulo(NaN)
corona_df.isnull().sum() # Identificando em qual campo tem dados Nulo (NaN) e sua quantidade
```

```
paper_id        0
title           0
abstract    19888
text            0
dtype: int64
```

```
# Limpando dados Nulo (NaN) em todas as colunas do dataframe
# O método "inplace" caso seja "TRUE" aplica as alterações do dataframe de forma automática
corona_df.dropna(inplace=True)
```

```
# Removendo registro do tipo Nulo(NaN) - VERIFICANDO NOVAMENTE
corona_df.isnull().sum() # identificando em qual campo tem dados Nulo (NaN) e sua quantidade
```

```
paper_id    0
title       0
abstract    0
text        0
dtype: int64
```

```
# Verificando o shape após a remoção de registros Nulo (NaN)
corona_df.shape
```

```
(31748, 4)
```

4. Conclusões e passos futuros

Os dados são cruciais para os modelos de ML. Sem bons dados, não existe um bom modelo. Na maioria dos casos, os dados coletados podem ser usados por algoritmos de ML somente após serem pré-processados. Fomos capazes de iniciar a limpeza de dados, etapa inicial do pré-processamento, transformando o CORD-19 em um objeto do tipo DataFrame da biblioteca Pandas. Com isso, foi possível aplicar suas funções e métodos para detectar valores faltantes (NaN) e vazios, bem como eliminá-los, evitando possíveis exceções no dataset CORD-19.

Os próximos passos envolverão, então, identificar e apagar valores duplicados, criar uma função para utilizar uma biblioteca do spaCY para fazermos o processamento de textos médicos e posteriormente remover palavras *Stopwords*[5] para reduzir a dimensionalidade.

Referências

Afzal, M., Alam, F., Malik, K. M., and Malik, G. M. (2020). Clinical context–aware biomedical text summarization using deep neural network: Model development and validation. *Journal of medical Internet research*, 22(10):e19810.

Lalmuanawma, S., Hussain, J., and Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review. *Chaos, Solitons & Fractals*, page 110059.

Ridzuan, F. and Zainon, W. M. N. W. (2019). A review on data cleansing methods for big data. *Procedia Computer Science*, 161:731–738.

Shuja, J., Alanazi, E., Alasmary, W., and Alashaikh, A. (2020). Covid-19 open source data sets: A comprehensive survey. *Applied Intelligence*, pages 1–30.

[5]Palavras que possuem apenas significado sintático dentro da sentença, porém não traz informações relevantes sobre o seu sentido.