**Documentação do Código: Análise Estatística de Texto**

Este código é uma aplicação web interativa construída com o **Streamlit** para realizar análise estatística de textos fornecidos pelo usuário. A análise inclui contagem de palavras, contagem de caracteres, identificação das palavras mais frequentes, contagem de sentenças e parágrafos, e geração de gráficos e nuvens de palavras. O código permite que o usuário forneça o texto de duas formas: digitando ou enviando um arquivo.

**Importação de Bibliotecas**

python

CopiarEditar

import streamlit as st

import pandas as pd

import json

import re

import matplotlib.pyplot as plt

import seaborn as sns

from collections import Counter

from wordcloud import WordCloud

import PyPDF2

import nltk

from nltk.corpus import stopwords

from unidecode import unidecode

* **streamlit**: Biblioteca utilizada para criar a interface gráfica da aplicação web.
* **pandas**: Usada para manipulação de dados, especialmente para ler arquivos CSV.
* **json**: Para manipulação de arquivos JSON.
* **re**: Biblioteca de expressões regulares para manipulação e limpeza do texto.
* **matplotlib** e **seaborn**: Bibliotecas para criação de gráficos.
* **collections.Counter**: Para contar as palavras no texto e obter as mais frequentes.
* **wordcloud**: Para gerar uma nuvem de palavras visual.
* **PyPDF2**: Para ler e extrair texto de arquivos PDF.
* **nltk**: Biblioteca para processamento de linguagem natural.
* **unidecode**: Usada para remover acentos dos caracteres.

**Funções Auxiliares**

**Função remove\_stopwords()**

python

CopiarEditar

def remove\_stopwords(words):

stop\_words = set(stopwords.words("portuguese")) # Usando as stopwords em português

return [word for word in words if word.lower() not in stop\_words]

Esta função recebe uma lista de palavras e remove as **stopwords** (palavras comuns que não agregam muito significado, como "de", "e", "a", etc.). Utiliza o conjunto de stopwords em português disponível na biblioteca **NLTK**.

**Função clean\_text()**

python

CopiarEditar

def clean\_text(text):

text = re.sub(r'[^\w\s]', '', text) # Remove caracteres especiais

text = unidecode(text) # Remove acentos

text = text.lower() # Coloca tudo em minúsculo

return text

A função clean\_text realiza três operações principais:

1. **Remove caracteres especiais** (pontuação, símbolos) usando expressões regulares.
2. **Remove acentos** das palavras utilizando a função unidecode.
3. **Converte todo o texto para minúsculas** para normalizar as palavras.

**Função text\_analysis()**

python

CopiarEditar

def text\_analysis(text):

text = clean\_text(text)

words = text.split()

words = remove\_stopwords(words)

word\_count = len(words)

char\_count = sum(len(word) for word in words)

word\_freq = Counter(words)

most\_common = word\_freq.most\_common(10)

sentences = re.split(r'[.!?]', text)

sentences = [s.strip() for s in sentences if s.strip()]

sentence\_count = len(sentences)

paragraphs = text.split("\n")

paragraphs = [p.strip() for p in paragraphs if p.strip()]

paragraph\_count = len(paragraphs)

avg\_word\_length = char\_count / word\_count if word\_count > 0 else 0

avg\_sentence\_length = word\_count / sentence\_count if sentence\_count > 0 else 0

return word\_count, char\_count, most\_common, sentence\_count, paragraph\_count, avg\_word\_length, avg\_sentence\_length, word\_freq

Esta função realiza a análise estatística do texto:

1. **Limpeza do texto**: O texto é passado pela função clean\_text() para remover caracteres especiais, acentos e converter para minúsculas.
2. **Contagem de palavras**: O texto é dividido em palavras e as stopwords são removidas.
3. **Contagem de caracteres**: Soma o número total de caracteres sem contar os espaços.
4. **Frequência das palavras**: Utiliza Counter para contar as ocorrências de cada palavra e obter as 10 palavras mais frequentes.
5. **Contagem de sentenças**: Divide o texto em sentenças usando expressões regulares e conta quantas sentenças existem.
6. **Contagem de parágrafos**: Divide o texto em parágrafos e conta quantos parágrafos existem.
7. **Cálculo da média**:
   * **Comprimento médio das palavras**: Calcula a média do número de caracteres por palavra.
   * **Comprimento médio das sentenças**: Calcula a média do número de palavras por sentença.

**Exemplo de Retorno da Função**

A função retorna vários valores para que possam ser apresentados na interface:

* **word\_count**: Número total de palavras.
* **char\_count**: Número total de caracteres (sem espaços).
* **most\_common**: As 10 palavras mais frequentes.
* **sentence\_count**: Número de sentenças.
* **paragraph\_count**: Número de parágrafos.
* **avg\_word\_length**: Comprimento médio das palavras.
* **avg\_sentence\_length**: Comprimento médio das sentenças.
* **word\_freq**: Frequência de todas as palavras no texto.

**Interface de Usuário com Streamlit**

**Definição do Título e Descrição**

python

CopiarEditar

st.title("📊 Análise Estatística de Texto")

st.write("Forneça um texto para análise estatística. Escolha entre escrever ou enviar um arquivo.")

Aqui, o título e a descrição são definidos para a interface da aplicação.

**Entrada de Texto ou Arquivo**

python

CopiarEditar

input\_choice = st.radio("Escolha como deseja fornecer o texto:", ("Escrever o texto", "Enviar um arquivo"))

O usuário pode escolher entre digitar o texto ou enviar um arquivo. O st.radio cria uma opção de rádio para escolher o método de entrada.

**Processamento do Texto**

Dependendo da escolha do usuário, o texto é capturado da seguinte forma:

* Se o usuário escolher "Escrever o texto", o texto é inserido via campo de texto (st.text\_area).
* Se o usuário escolher "Enviar um arquivo", o código processa o arquivo enviado (texto, CSV, JSON ou PDF).

**Exibição dos Resultados da Análise**

Após o processamento do texto, a análise estatística é realizada pela função text\_analysis(). Os resultados são apresentados na interface, incluindo:

* **Número total de palavras e caracteres**.
* **Média de comprimento das palavras e sentenças**.
* **Número de sentenças e parágrafos**.

Além disso, a função gera uma tabela com as palavras mais frequentes.

**Geração de Gráficos**

O usuário pode escolher o tipo de gráfico a ser gerado para visualizar a distribuição de palavras:

* **Gráfico de Barras**: Exibe a frequência das 10 palavras mais comuns.
* **Gráfico de Pizza**: Exibe as porcentagens de ocorrência das palavras mais frequentes.
* **Nuvem de Palavras**: Gera uma nuvem de palavras visualmente atraente, onde as palavras mais frequentes aparecem em tamanho maior.

A escolha do gráfico é feita através de um st.radio com as opções "Gráfico de Barras", "Nuvem de Palavras" e "Gráfico de Pizza".

**Conclusão**

Este código cria uma ferramenta interativa para análise estatística de textos, permitindo que os usuários insiram texto manualmente ou carreguem arquivos de diferentes formatos (TXT, CSV, JSON, PDF) para análise. Ele oferece insights sobre o conteúdo textual, como a frequência das palavras, número de sentenças e parágrafos, e também permite a visualização desses dados por meio de gráficos e nuvens de palavras.