DOCUMENTAÇÃO TÉCNICA DO CÓDIGO-FONTE

Projeto de Iniciação Científica (IC)

Coleta e Organização de Dados do Reddit para Análise de Sentimentos

1. Visão Geral do Projeto

Este projeto tem como objetivo coletar, filtrar, organizar e armazenar postagens do Reddit escritas em português, associadas a diferentes categorias de sentimento (positivo, negativo e neutro), com foco em temas relacionados à saúde mental, emoções e experiências humanas.

O código foi desenvolvido no contexto de um Projeto de Iniciação Científica voltado à Análise de Sentimentos em mídias sociais, utilizando dados reais, não estruturados e espontâneos.

A aplicação contempla as seguintes etapas do pipeline de dados:
• Coleta automatizada via API
• Limpeza e pré-processamento textual
• Filtragem por idioma
• Organização estruturada
• Geração de dataset pronto para análise estatística ou aprendizado de máquina

2. Tecnologias e Ferramentas Utilizadas

Linguagem: Python

Bibliotecas:
- praw: Coleta de dados via API do Reddit
- pandas: Estruturação em DataFrame e exportação Excel
- re: Expressões regulares para limpeza
- emoji: Conversão de emojis em texto
- langdetect: Identificação automática de idioma
- openpyxl: Escrita de arquivos .xlsx

3. Configuração Inicial do Código

Importações principais:
```
import praw
import pandas as pd
import re
```

import emoji
from langdetect import detect, LangDetectException

Remoção de caracteres inválidos:
ILLEGAL_CHARACTERS_RE = re.compile(r'[\000-\010]|[\013-\014]|[\016-\037]')

Essa expressão evita erros ao exportar para Excel.

4. Definição dos Subreddits

subreddits_alvo = ["conversas", "desabafos", "brasil"]

Critérios:
• Alta presença de textos em português
• Relatos pessoais e discussões emocionais
• Diversidade temática

5. Configuração da API do Reddit

reddit = praw.Reddit(
client_id='...',
client_secret='...',
user_agent='ICAnaliseSentimentosBot/0.1 by AnaliseSentimentosIC'
)

O PRAW gerencia autenticação OAuth, limites de requisição e tratamento de erros.

6. Estruturas de Controle

postsEncontrados = []
textos_vistos = set()
ids_vistos = set()

Uso de set() garante unicidade e eficiência na deduplicação.

7. Organização das Palavras-chave

palavrasChavesGrupos = {
"positivo": ["amo", "feliz", "alegre", "adoro"],

"negativo": ["raiva", "triste", "ódio", "ansioso"],
"neutro": ["terapia", "autoestima", "sentimento", "apoio"]
}

Cada palavra possui mínimo de 500 postagens para balanceamento.

8. Função de Limpeza de Texto

def limpar_texto(t):
t = t.replace("\n", " ").strip()
t = emoji.demojize(t)
t = ILLEGAL_CHARACTERS_RE.sub("", t)
return t.lower()

Responsável pela padronização e redução de ruído textual.

9. Estratégia de Coleta em Larga Escala

for submission in subreddit.new(limit=5000):

Percorre postagens recentes e aplica filtragem manual de palavras-chave, superando limitações da função search().

10. Filtragem por Idioma

try:
if detect(texto) != "pt":
continue
except LangDetectException:
continue

Garante coerência linguística do corpus.

11. Controle de Quantidade

if total_palavra >= 500:
break

Assegura balanceamento estatístico entre categorias.

12. Estrutura dos Dados Armazenados

Campos:
• data
• texto
• autor
• subreddit
• palavraChave
• categoria

13. Contagem Estatística

```
contagem.append({
"categoria": categoria,
"palavra": palavra,
"total": total_palavra
})
```

Utilizada para validação e análise exploratória.

14. Exportação para Excel

```
with pd.ExcelWriter(caminho, engine="openpyxl") as writer:
```

O arquivo final contém:
• Uma aba por categoria emocional
• Uma aba adicional com estatísticas

Facilita inspeção manual e integração com ferramentas de Machine Learning.