

Título: Análise de Texto em PDF com Unigrama, Bigrama e Trigrama

Descrição do Desafio: Desenvolva um programa em Python que seja capaz de:

1. Receber um arquivo PDF carregado pelo usuário.
2. Extrair o texto do PDF.
3. Realizar a tokenização do texto para criar unigrama, bigrama e trigrama.

Regras e Requisitos:

- Utilize a biblioteca PyPDF2 para extrair o texto do PDF.
- Utilize a biblioteca nltk para a tokenização e geração dos n-gramas.
- O programa deve lidar com exceções e exibir mensagens de erro amigáveis caso o texto não possa ser extraído.
- Exiba as 10 primeiras entradas de cada tipo de n-grama (unigrama, bigrama e trigrama) no console ou em uma interface gráfica (como o Streamlit).

Critérios de Avaliação:

- **Corretude:** O código deve extrair e processar corretamente o texto do PDF.
- **Legibilidade:** O código deve ser bem organizado e comentado.
- **Trabalho com Erros:** Implementação de tratamento de erros para arquivos que não possam ser processados corretamente.
- **Simplicidade e Eficiência:** O código deve ser simples e eficiente para que possa ser mantido facilmente.

Entrega:

- O código deve ser enviado em uma aplicação Google Colab.

Exemplo de Saída Esperada:

- Unigramas: ['este', 'é', 'um', 'exemplo', 'de', ...]
- Bigramas: [('este', 'é'), ('é', 'um'), ('um', 'exemplo'), ...]
- Trigramas: [('este', 'é', 'um'), ('é', 'um', 'exemplo'), ...]

Texto do Desafio:

“A programação é uma arte que mistura lógica, criatividade e inovação. Cada linha de código é como uma peça em um quebra-cabeça, colaborando para criar algo maior e funcional. Com o tempo, desenvolvedores aprendem a apreciar a elegância de algoritmos bem estruturados e a satisfação de resolver problemas complexos. O mundo moderno está repleto de aplicações que facilitam nossas vidas, desde simples aplicativos de mensagens até sistemas de inteligência artificial capazes de tomar decisões autônomas. Por trás de cada tecnologia, há desenvolvedores comprometidos em transformar ideias em realidade.”