

Manual de Instalação

Chat-BTG

Índice

1. Introdução	3
1.1. Solução	3
2. Instalação do modelo	
3. Implantação no Streamlit	5
4. API da OpenAI	

1. Introdução

1.1. Solução

Diante do que foi apresentado como objetivo, a principal necessidade de negócio identificado pelo Banco BTG Pactual é compreender as opiniões, sentimentos e necessidades dos clientes expressos nas redes sociais, em especial, para fins de prova de conceito, na plataforma Instagram.

Neste sentido, pretendemos entregar um modelo de análise de sentimentos que faça a detecção de palavras-chave nos comentários das redes sociais, utilizando o Processamento de Linguagem Natural (PLN).

Este projeto visa suprir a necessidade de entender o cliente, o que ele deseja e o que não deseja em relação ao banco e suas campanhas. A análise de sentimento permite ao banco compreender melhor o cliente, adaptando-se e oferecendo uma experiência cada vez mais satisfatória.

2. Instalação do modelo

O BERT (Bidirectional Encoder Representations from Transformers) é um modelo de linguagem baseado em Transformers, arquitetura que é baseada em mecanismos de "atenção" e relevância. Isto permite que o modelo capture relações de dependência entre palavras, similaridade e contexto de maneira eficiente e paralela.

Primordialmente, destaca-se a instalação do framework PyTorch, utilizado na criação e treinamento de redes neurais. Em seguida, a instalação da biblioteca associada "Transformers" fornece implementações pré-treinadas do BERT

Consequente, é ideal a instalação das classes e utilitários fornecidos pelo PyTorch "TensorDataset, DataLoader, RandomSampler, SequentialSampler", a fim de facilitar a manipulação e treinamento do modelo.

Já a instalação do otimizador "AdamW", deve ser realizada logo em seguida.

Por fim, instala-se a biblioteca do scikit-learn "train_test_split" que divide os dados em conjuntos de treinamento e teste.

A biblioteca **pandas** é comumente utilizada em programas Python e novamente deve ser instalado para fazer manipulações básicas dos dados.

```
!pip install torch
!pip install transformers
!pip install pandas
```

```
import torch
from transformers import BertTokenizer, BertForSequenceClassification
from torch.utils.data import TensorDataset, DataLoader, RandomSampler, SequentialSampler
from torch.optim import AdamW
from sklearn.model_selection import train_test_split
import pandas as pd
```

3. Implantação no Streamlit

Pré-Requisitos: Configuração de um ambiente virtual e a instalação do Python em seu sistema.

Primeiramente, é necessário preparar o ambiente no qual você irá trabalhar. Em seguida, você deve instalar o Streamlit utilizando o comando `!pip install streamlit`

Após a instalação, crie um arquivo Python no diretório do seu projeto e importe as bibliotecas necessárias. No ambiente Chat-BTG, as seguintes bibliotecas estão instaladas:

Bibliotecas instaladas no Chat-BTG:

```
import streamlit as st
import pandas as pd
import plotly.express as px
import base64
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from io import BytesIO
from collections import Counter
import emoji
```

Em seguida, crie a estrutura básica do Streamlit no seu arquivo. Aqui está um exemplo de código estruturado para o Chat-BTG:

```
def read_csv(upload_file):
    if upload_file is not None:
        df = pd.read_csv(upload_file)
        return df
    else:
        return None
```

Finalmente, navegue até o diretório onde o seu arquivo está localizado utilizando o terminal e execute o seguinte comando para iniciar o Streamlit localmente:

Comando de execução:

```
streamlit run app.py
```

4. API da OpenAI

1. Geração de uma API Key na plataforma da OpenAI:

Primeiramente, deve ser feita uma conta no site da OpenAI (<https://platform.openai.com/>) e na seção 'view API keys' é possível gerar uma chave para uso no arquivo .py.

2. Instalação do pandasai e da openai no ambiente virtual criado, com os comandos:

```
!pip install pandasai
pip install openai
```

3. Importação e preparação da variável a ser utilizada para aplicação do PandasAI:

```
from pandasai import PandasAi
from pandasai.llm.openai import OpenAI
```

```
llm = OpenAI(api_token='token da api')
```

```
pandas_ai = PandasAI(llm)
```

4. Criação de uma estrutura que recebe um input do usuário e gera uma resposta baseada na base de dados inserida:

```
uploaded_file = st.file_uploader("Insira o arquivo CSV",
type='csv')
```

```
if uploaded_file is not None:
```

```
    df = pd.read_csv(uploaded_file)
```

```
    st.write(df.head(3))
```

```
prompt = st.text_area("Coloque a análise desejada:")
```

```
    # Generate output
```

```
    if st.button("Generate"):
```

```
        if prompt:
```

```
            # call pandas_ai.run(), passing dataframe and prompt
```

```
            with st.spinner("Gerando a resposta..."):
```

```
                st.write(pandas_ai.run(df, prompt))
```

```
        else:
```

```
            st.warning("Por favor, coloque uma análise.")
```