

Assessment

Engenharia de Prompts para Ciência de Dados

Pedro Monteiro Vieira Bayma Azevedo

Imports

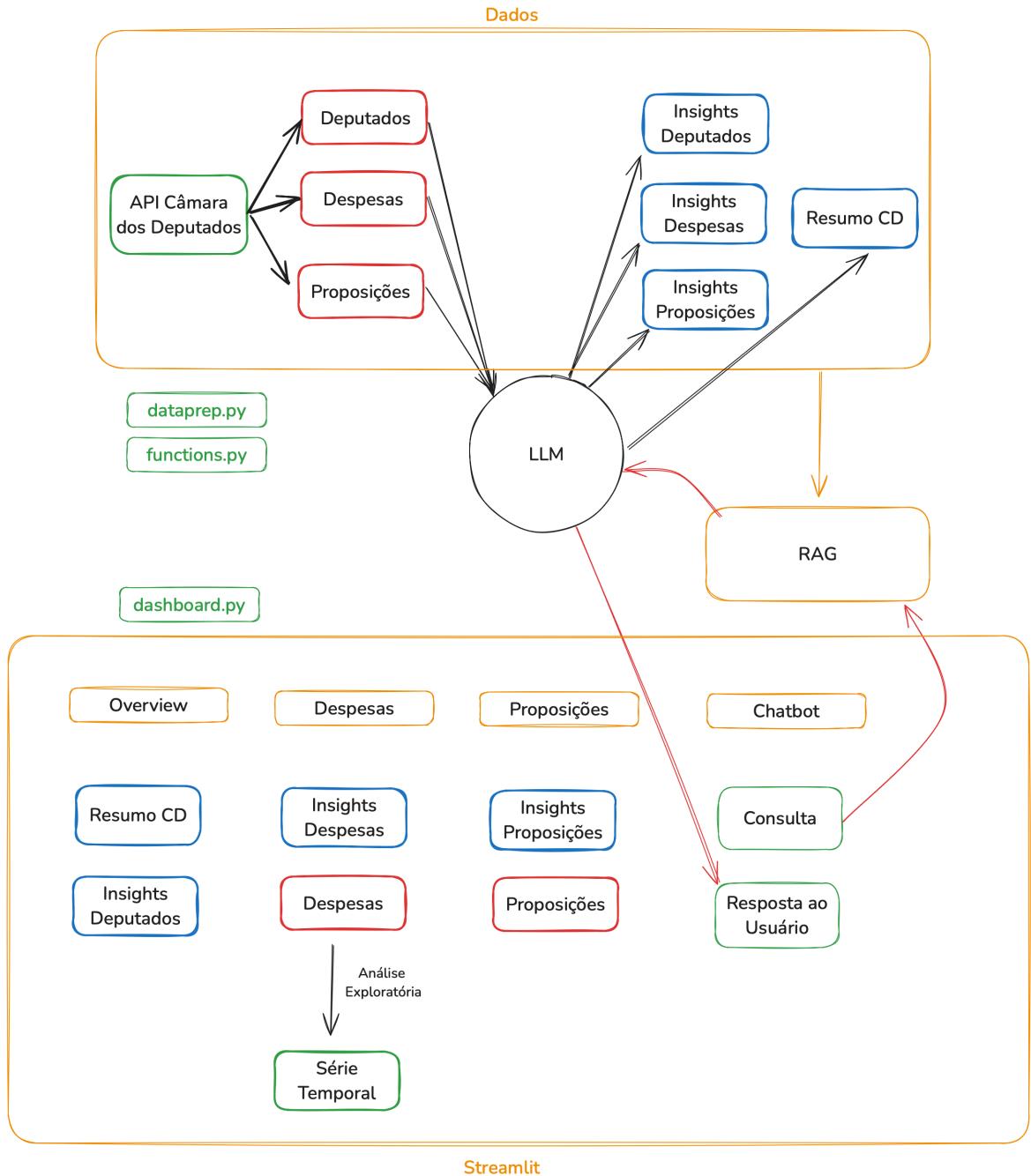
```
In [ ]: import os
from dotenv import load_dotenv
import google.generativeai as genai
import matplotlib.pyplot as plt
import json
from functions import *
import numpy as np
import faiss
import torch
from sentence_transformers import SentenceTransformer
from transformers import pipeline
from diffusers import DiffusionPipeline
import requests
import pandas as pd
```

```
In [3]: #loading api key
load_dotenv('.env')
```

Out[3]: True

Exercício 1

a



b

A arquitetura consiste em coletar dados a partir da API da Câmara dos Deputados e salvá-los em disco. Posteriormente obter e também salvar em disco insights e resumos utilizando para essa tarefa o LLM Gemini do Google.

Com esses dados em mãos foi criada aplicação em streamlit que carrega e apresenta ao usuário esses dados de acordo com o esquema acima, bem como a criação de uma série temporal.

Por fim, ainda com esses dados obtidos, foi criado um RAG que vetorizando os dados de forma a criar um repositório de conhecimento para que o chatbot possa consultar de forma a responder as perguntas do usuário.

C

Os LLMs, treinados a partir de redes neurais, conseguem identificar, a partir de um texto fornecido, contextos e compreender informações fornecidas a eles, provendo uma resposta adequada ao que foi questionado.

Dessa forma, e principalmente os LLMs treinados para lidar bem com textos, como o ChatGPT 4, conseguem utilizando o texto inicial, bem como o contexto capturado gerar uma versão resumida do texto preservando seus pontos principais, identificando ideias centrais e removendo informações redundantes.

Exercício 2

The screenshot shows a chat interface on the Câmara dos Deputados website. At the top, there's a header with the Câmara dos Deputados logo and the text "Câmara dos Deputados" and "Claude-3.5-Sonnet". On the right, there's a "New chat" button. Below the header, the Claude-3.5-Sonnet profile is displayed, showing a sunburst icon, the name "Claude-3.5-Sonnet", "By @poe", "20K followers", and buttons for "Chats", "View details", and "OFFICIAL". A message box contains the question: "Poderia me explicar em um texto de no máximo 2 parágrafos o que é a Câmara dos Deputados e qual sua função no Brasil?". The AI response follows:

A Câmara dos Deputados é uma das duas casas do Congresso Nacional brasileiro, sendo composta por 513 deputados federais eleitos pelo povo para mandatos de quatro anos. Os deputados são distribuídos proporcionalmente entre os estados e o Distrito Federal de acordo com suas populações, representando os interesses de seus eleitores e regiões.

Sua principal função é elaborar e votar leis que regulam a vida no país, além de fiscalizar o Poder Executivo quanto à administração e aos gastos públicos. A Câmara também tem poder para propor e aprovar emendas à Constituição, analisar medidas provisórias enviadas pelo presidente da República e julgar crimes de responsabilidade cometidos por autoridades como o próprio presidente, vice-presidente e ministros de Estado.



Gemini-1.5-Pro

By @poe

8.9K followers

[Chats](#)

Powered by gemini-1.5-pro-002. The multi-modal model from Google's Gemini family that balances model performance and speed. The model accepts text, image, and video input from the entire conversation and provides text output, with a restriction of one video per message. Ideal for generating text, answering questions, and providing summaries.

175 ⚡ per message · [View details](#)

OFFICIAL

Poderia me explicar em um texto de no máximo 2 parágrafos o que é a Câmara dos Deputados e qual sua função no Brasil?



Gemini-1.5-Pro @poe

A Câmara dos Deputados, juntamente com o Senado Federal, compõe o Congresso Nacional, o órgão que exerce o Poder Legislativo no Brasil. Ela representa o povo brasileiro, sendo seus membros eleitos pelo sistema proporcional, com cada estado e o Distrito Federal tendo direito a um número de deputados proporcional à sua população. A principal função da Câmara é elaborar e votar leis que afetam o país como um todo, incluindo temas como orçamento, segurança pública, educação, saúde e economia. Além disso, a Câmara também tem o poder de fiscalizar o Poder Executivo, abrindo investigações e, em casos extremos, iniciando processos de impeachment contra o presidente da República, ministros de Estado e outros altos funcionários.

A Câmara dos Deputados tem um papel fundamental na democracia brasileira, atuando como um espaço de debate e representação dos diversos interesses da população. Através do processo legislativo, os deputados discutem, propõem, alteram e aprovam leis que visam o bem comum. A fiscalização do Poder Executivo também é crucial para garantir a transparência e a responsabilidade na administração pública. A composição proporcional da Câmara busca assegurar que todas as regiões e segmentos da sociedade brasileira tenham voz na tomada de decisões que afetam o país.



GPT-4o

By @poe

44K followers

[Chats](#)

OpenAI's most powerful model, GPT-4o, leveraging the latest Nov 2024 model snapshot, which provides more natural, engaging & tailored writing and overall provides more thorough, insightful responses. Stronger than GPT-3.5 in quantitative questions (math and physics), creative writing, and reasoning.

289 ⚡ per standard message · [View details](#)

OFFICIAL

Poderia me explicar em um texto de no máximo 2 parágrafos o que é a Câmara dos Deputados e qual sua função no Brasil?



GPT-4o @poe

A Câmara dos Deputados é uma das duas casas do Poder Legislativo federal no Brasil, compondo o Congresso Nacional ao lado do Senado Federal. Formada por representantes eleitos pelo povo, os deputados federais são escolhidos por meio de voto direto a cada quatro anos. O número de deputados por estado é proporcional à sua população, garantindo uma representação nacional. Essa casa legislativa está situada em Brasília, no Palácio do Congresso Nacional.

A principal função da Câmara dos Deputados é elaborar, discutir e votar leis que regem o país, além de fiscalizar os atos do Poder Executivo, garantindo que o governo atue conforme a Constituição. Ela também tem o papel de analisar e aprovar propostas de emenda à Constituição, projetos de lei, medidas provisórias e orçamento federal. Além disso, a Câmara pode instaurar processos de impeachment contra o presidente da República e outras autoridades públicas, exercendo um papel crucial na garantia da democracia e da transparência governamental.

Respostas

a

O Claude tem como principais vantagens suas características mais voltadas para

tarefas complexas e analíticas, bem como tarefas que envolvam código e escrita.

Como desvantagem podemos apontar o tamanho da janela de contexto que é menor que o de outros LLMs e atualmente estar em versão de testes.

O Gemini Pro tem como principal vantagem sua capacidade multimodal que lida bem com textos, imagens, etc, e o custo menor quando comparado por exemplo ao GPT 4o. Com desvantagem podemos apontar sua menos consistência nas respostas e na capacidade de compreender com precisão comandos quando comparado com outros modelos.

Por fim, o GPT 4o tem como vantagens o tamanho de sua janela de contexto, bem como sua performance de forma geral, podendo atuar em vários contextos. Como desvantagem, temos o custo elevado quando comparado a outros modelos e a necessidade de assinatura.

b

Todas as respostas foram corretas. A resposta do Claude e do GPT foram extremamente próximas, tanto em conteúdo quanto em tamanho e temas abordados. Por sua vez a resposta do Gemini foi mais abrangente incluindo outras informações e foi um pouco maior que as outras duas.

c

Eu elejo a resposta do Claude como a mais próxima do esperado por ter atendido melhor aos requisitos de ser curta e trazer conteúdo que descrevia bem, de forma mais geral, o papel da Câmara dos Deputados.

d

poderia formar essa resposta como yaml com a chave "overview_summary"?
obrigado

Claude-3.5-Sonnet Poe

yaml

Copy

overview_summary:
description: |

A Câmara dos Deputados é uma das duas casas do Congresso Nacional brasileiro, sendo composta por 513 deputados federais eleitos pelo povo para mandatos de quatro anos. Os deputados são distribuídos proporcionalmente entre os estados e o Distrito Federal de acordo com suas populações, representando os interesses de seus eleitores e regiões.

Sua principal função é elaborar e votar leis que regulam a vida no país, além de fiscalizar o Poder Executivo quanto à administração e aos gastos públicos. A Câmara também tem poder para propor e aprovar emendas à Constituição, analisar medidas provisórias enviadas pelo presidente da República e julgar crimes de responsabilidade cometidos por autoridades como o próprio presidente, vice-presidente e ministros de Estado.

sem esse description

Claude-3.5-Sonnet Poe

yaml

Copy

overview_summary: A Câmara dos Deputados é uma das duas casas do Congresso Nacional brasileiro, sendo composta por 513 deputados federais eleitos pelo povo para mandatos de quatro anos. Os deputados são distribuídos proporcionalmente entre os estados e o Distrito Federal de acordo com suas populações, representando os interesses de seus eleitores e regiões. Sua principal função é elaborar e votar leis que regulam a vida no país, além de fiscalizar o Poder Executivo quanto à administração e aos gastos públicos. A Câmara também tem poder para propor e aprovar emendas à Constituição, analisar medidas provisórias enviadas pelo presidente da República e julgar crimes de responsabilidade cometidos por autoridades como o próprio presidente, vice-presidente e ministros de Estado.

Exercício 3

a - código dentro do arquivo dataprep.py

b

In [17]: pizzaPrompt = """

Contexto

Você é um especialista em ciência de dados.

Eu posso um arquivo parquet com os dados dos deputados do brasil. Nesses siglaPartido, do partido de cada deputado.

Instrução

Favor gerar código em python que crie um gráfico de pizza com o total e o Utilize as bibliotecas pandas, e matplotlib.

No fim, o código deve salvar o gráfico em um arquivo png com o nome de "docs/distribuicao_deputados.png". Existente docs utilizando o seguinte path: "docs/distribuicao_deputados.py"

Favor retornar somente o código em python sem nenhum comentário adicional

.....

```
genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(pizzaPrompt)
print(response.text)
```

```
```python
import pandas as pd
import matplotlib.pyplot as plt

Certifique-se de que o arquivo parquet esteja no mesmo diretório ou especifique o caminho completo
df = pd.read_parquet("deputados.parquet")

Contagem de deputados por partido
partidos = df['siglaPartido'].value_counts()

Criando o gráfico de pizza
plt.figure(figsize=(10, 10))
plt.pie(partidos, labels=partidos.index, autopct='%.1f%%', startangle=140)
plt.title('Distribuição de Deputados por Partido')
plt.axis('equal') # Para garantir que o gráfico seja um círculo

Salvando o gráfico
plt.savefig("docs/distribuicao_deputados.png")

plt.show()
```
```

```

```
In [20]: import pandas as pd
import matplotlib.pyplot as plt

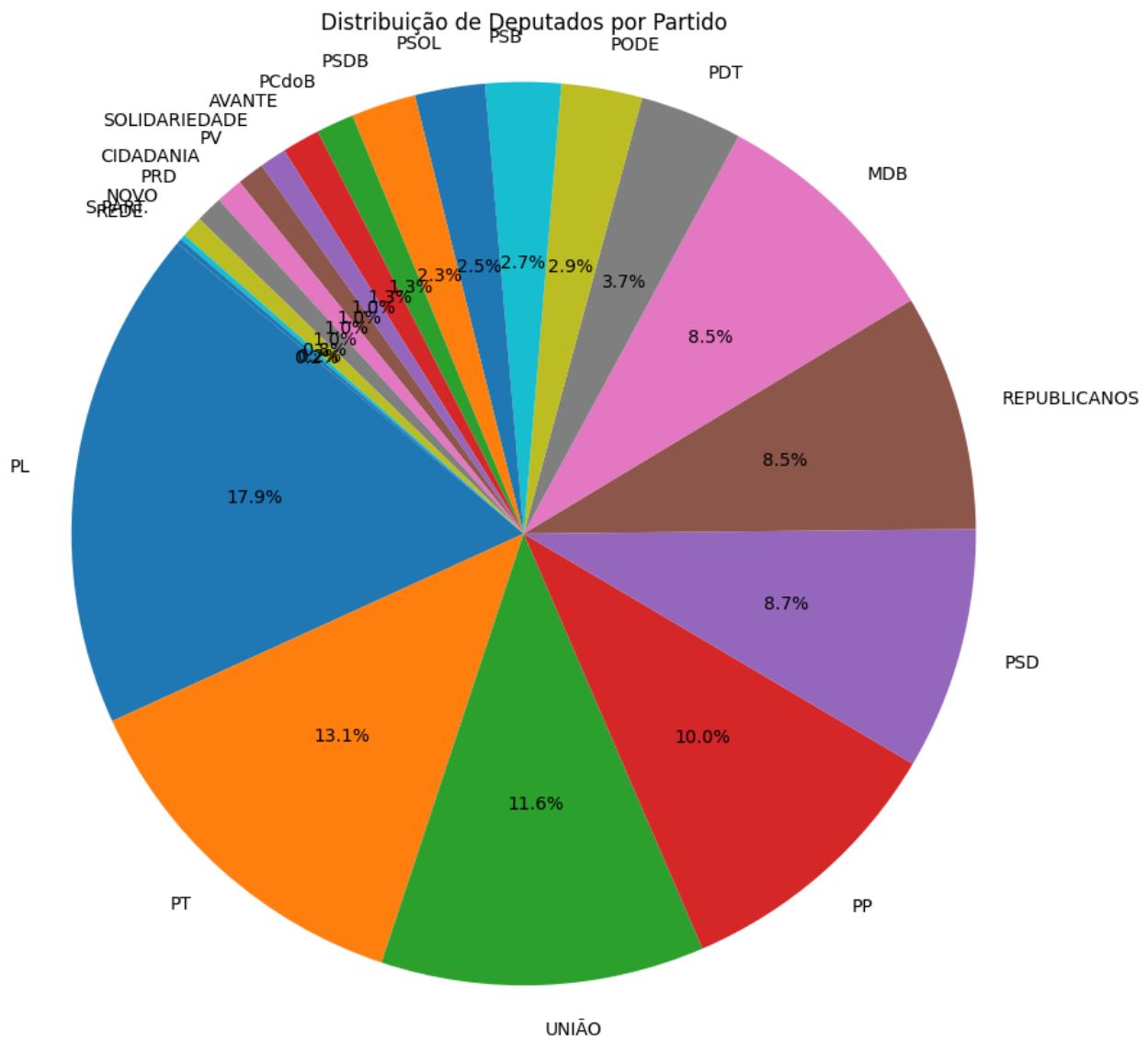
Certifique-se de que o arquivo parquet esteja no mesmo diretório ou especifique o caminho completo
df = pd.read_parquet("./data/deputados.parquet")

Contagem de deputados por partido
partidos = df['siglaPartido'].value_counts()

Criando o gráfico de pizza
plt.figure(figsize=(10, 10))
plt.pie(partidos, labels=partidos.index, autopct='%.1f%%', startangle=140)
plt.title('Distribuição de Deputados por Partido')
plt.axis('equal') # Para garantir que o gráfico seja um círculo

Salvando o gráfico
plt.savefig("docs/distribuicao_deputados.png")

plt.show()
```



C

```
In [45]: df = pd.read_parquet("./data/deputados.parquet")
partidos = df['siglaPartido'].value_counts()
partidos_texto = str(partidos.to_dict())

insightPartidosPrompt = f"""
Contexto
Você é um especialista em ciência de dados.
Eu posso ter um arquivo texto com a distribuição dos deputados por partido.

Instrução
Favor gerar insights a partir do dado {partidos_texto} sobre a distribuição por
partido e como isso pode impactar o dia a dia da câmara dos deputados.

Favor retornar a análise em um formato json com a seguinte estrutura:

```json

```

```

    "insight1": "Insight 1",
    "insight2": "Insight 2",
    "insight3": "Insight 3"

    ...
    ...

genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(insightPartidosPrompt)
print(response.text)

```json
{
 "insight1": "O PL detém uma posição de força significativa na Câmara dos Deputados, com 93 deputados, representando quase um terço do total. Isso concede ao partido um poder considerável na aprovação de projetos de lei e na formação de alianças, potencialmente levando a uma agenda legislativa mais alinhada com suas prioridades e influenciando fortemente o rumo das votações.",

 "insight2": "A fragmentação partidária é evidente, com diversos partidos possuindo representações menores. A presença de diversos partidos com poucos deputados (como NOVO, REDE, S.PART.) dificulta a formação de coalizões estáveis e pode resultar em maior instabilidade política e negociações mais complexas para a aprovação de propostas, demandando maior capacidade de articulação política dos líderes partidários.",

 "insight3": "O PT, apesar de ter uma expressiva bancada com 68 deputados, está em posição de oposição ao PL. A dinâmica entre esses dois partidos de grande porte moldará significativamente o cenário político da Câmara. A capacidade de negociação e a construção de consensos entre esses dois blocos serão cruciais para a governabilidade e a aprovação de leis importantes. A formação de blocos de apoio ou oposições poderá gerar impactos significativos no processo de aprovação de projetos de lei e na definição da agenda legislativa. O sucesso de projetos de lei e a eficiência da Câmara dependerão em grande parte da capacidade de esses partidos de negociarem e encontrarem pontos em comum, ou se houver capacidade de criação de coalizões mais amplas."
}
```

```

```

In [ ]: clean_response= response.text.replace(```json\n", '') .replace("\n```", '')
insights = json.loads(clean_response)

with open("./data/insights_distribuicao_deputados.json", "w") as file:
    json.dump(insights, file, indent=4, ensure_ascii=False)

```

Resposta

O prompt define a instrução a ser passada ao LLM e será formada por alguns componentes. A persona, um desses componentes é responsável por prover o papel e contextualizar o LLM com relação a como se portar e qual deve ser o foco de suas

respostas junto ao usuário. Os dados servem como a base de informação que o LLM recebe para poder trabalhar e prover a resposta e por fim, os exemplos servem para demonstrar e contextualizar o LLM da maneira que a resposta deve ser dada, de forma a permitir que sejam respeitados padrões e formatos.

Exercício 4

a - código dentro do arquivo dataprep.py

b

```
In [74]: df_despesas = pd.read_parquet("./data/serie_despesas_diárias_deputados.parquet")
despesas_texto = str(df_despesas.to_dict())
despesas_texto
```

```
Out[74]: "{'id': {1617: 141411, 4613: 220662, 4025: 141464, 5738: 178896, 857: 220605}, 'nome': {1617: 'Dagoberto Nogueira', 4613: 'Lucas Ramos', 4025: 'José Airton Félix Cirilo', 5738: 'Newton Cardoso Jr', 857: 'Bandeira de Mello'}, 'tipo_despesa': {1617: 'MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE PARLAMENTAR', 4613: 'MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDAD E PARLAMENTAR', 4025: 'MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE PAR LAMENTAR', 5738: 'COMBUSTÍVEIS E LUBRIFICANTES.', 857: 'MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE PARLAMENTAR'}, 'valor_documento': {1617: 9000.0, 4613: 1100.0, 4025: 5558.24, 5738: 200.03, 857: 517.0}, 'data_documento': {1617: '2024-07-03T00:00:00', 4613: '2024-10-04T00:00:00', 4025: '2024-08-27T00:00:00', 5738: '2024-07-31T00:00:00', 857: '2024-07-17T00:00:00'}}"
```

```
In [76]: df_despesas = pd.read_parquet("./data/serie_despesas_diárias_deputados.parquet")
despesas_texto = str(df_despesas.to_dict())

prompt1 = f"""
### Contexto

Você é um especialista em ciência de dados.
Eu posso usar um arquivo parquet com as despesas dos deputados que contém some
id, nome, tipo_despesa, valor_documento, data_documento conforme amostra

{despesas_texto}

### Instrução
Favor sugerir 3 análises simples que possam ser feitas somente com os dad
......

genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(prompt1)
print(response.text)
```

Com os dados fornecidos, podemos realizar algumas análises simples e descritivas. Eis três sugestões:

1. Despesa total por deputado:

Esta análise visa determinar o total gasto por cada deputado. Isso fornece uma visão geral imediata de quem teve as maiores despesas.

* **Método:** Agrupar os dados por `nome` e somar o `valor_documento` para cada deputado.

* **Resultado:** Uma tabela ou gráfico mostrando o nome de cada deputado e o total gasto por ele. Essa análise pode ser complementada com a visualização de um gráfico de barras, ordenando os deputados pelo valor total gasto, para melhor percepção visual.

* **Interpretação:** Identifica os deputados com os maiores gastos, permitindo investigações posteriores sobre a natureza dessas despesas.

2. Despesa média por tipo de despesa:

Esta análise calcula a despesa média para cada `tipo_despesa`. Isso ajuda a identificar quais tipos de despesas são mais caras em média.

* **Método:** Agrupar os dados por `tipo_despesa` e calcular a média do `valor_documento` para cada grupo.

* **Resultado:** Uma tabela mostrando cada tipo de despesa e sua média de custo. Um gráfico de barras seria também uma ótima visualização.

* **Interpretação:** Permite identificar os tipos de despesas que consomem mais recursos em média, indicando possíveis áreas de otimização ou necessidade de maior transparência.

3. Distribuição das despesas ao longo do tempo:

Esta análise explora a variação das despesas ao longo do tempo.

* **Método:** Converter a coluna `data_documento` para um formato de data apropriado (se necessário) e então agrregar as despesas por períodos (ex: mês, trimestre). Podemos calcular a soma das despesas para cada período.

* **Resultado:** Uma tabela ou gráfico de linha mostrando a evolução das despesas totais ao longo do tempo. A granularidade (dia, mês, trimestre) deve ser escolhida de acordo com a quantidade de dados disponíveis.

* **Interpretação:** Identifica tendências temporais nas despesas, revelando se há picos ou períodos de maior ou menor gasto. Isso pode indicar fatores sazonais ou eventos específicos que influenciam as despesas.

Estas três análises são relativamente simples de implementar utilizando bibliotecas como Pandas (Python) ou dplyr (R) e podem fornecer insights valiosos sobre os dados das despesas dos deputados. A escolha da melhor visualização para cada análise dependerá do tamanho do dataset e do público alvo.

```
In [83]: prompt2 = f'''
```

```
### Contexto
```

Você é um especialista em ciência de dados.
Eu posso um arquivo parquet com as despesas dos deputados que contém some
id, nome, tipo_despesa, valor_documento, data_documento conforme amostra

{despesas_texto}

Instrução

Favor gerar o código python para a seguinte análise:

1. Despesa total por deputado:

Esta análise visa determinar o total gasto por cada deputado. Isso forne

* **Método:** Agrupar os dados por `nome` e somar o `valor_documento` par

* **Resultado:** Uma tabela ou gráfico mostrando o nome de cada deputado

* **Interpretação:** Identifica os deputados com os maiores gastos, permi

Requisitos:

O código python deve considerar que o arquivo parquet possui o seguinte p

O código deve salvar a tabela final em um arquivo csv com o nome "despesa em ./data/despesa_total_por_deputado.csv"

O código deve mostrar o gráfico de barras com a análise.

A saída deve ser somente o código em python sem comentários adicionais.

.....

```
genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(prompt2)
print(response.text)
```

```

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

Carrega o arquivo parquet
df = pd.read_parquet('./data/serie_despesas_diárias_deputados.parquet')

Calcula a despesa total por deputado
despesa_total = df.groupby('nome')['valor_documento'].sum().reset_index()

Salva a tabela em um arquivo CSV
despesa_total.to_csv('./data/despesa_total_por_deputado.csv', index=False)

Cria o gráfico de barras
plt.figure(figsize=(12, 6))
sns.barplot(x='nome', y='valor_documento', data=despesa_total, order=despesa_total.sort_values('valor_documento', ascending=False).nome)
plt.xticks(rotation=90)
plt.xlabel('Deputado')
plt.ylabel('Total Gasto')
plt.title('Despesa Total por Deputado')
plt.tight_layout()
plt.show()

```

```

```

In [84]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

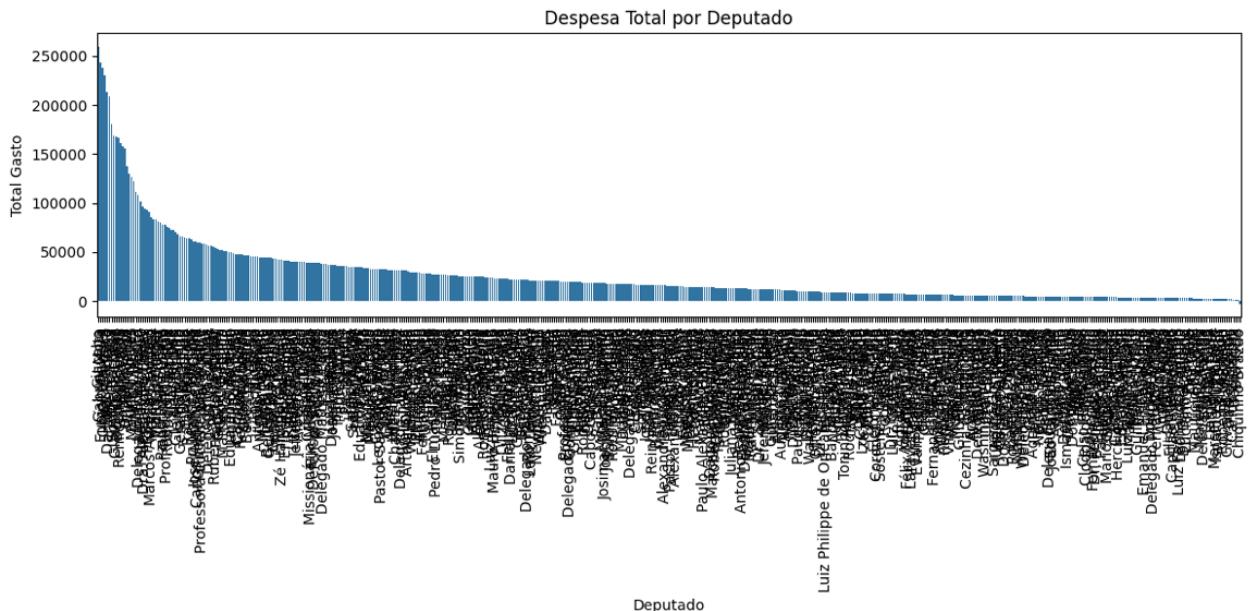
# Carrega o arquivo parquet
df = pd.read_parquet('./data/serie_despesas_diárias_deputados.parquet')

# Calcula a despesa total por deputado
despesa_total = df.groupby('nome')['valor_documento'].sum().reset_index()

# Salva a tabela em um arquivo CSV
despesa_total.to_csv('./data/despesa_total_por_deputado.csv', index=False)

# Cria o gráfico de barras
plt.figure(figsize=(12, 6))
sns.barplot(x='nome', y='valor_documento', data=despesa_total, order=despesa_total.sort_values('valor_documento', ascending=False).nome)
plt.xticks(rotation=90)
plt.xlabel('Deputado')
plt.ylabel('Total Gasto')
plt.title('Despesa Total por Deputado')
plt.tight_layout()
plt.show()

```



In [86]: prompt3 = f""

Contexto

Você é um especialista em ciência de dados.

Eu posso um arquivo parquet com as despesas dos deputados que contém some id, nome, tipo_despesa, valor_documento, data_documento conforme amostra

{despesas_texto}

Instrução

Favor gerar o código python para a seguinte análise:

****2. Despesa média por tipo de despesa:****

Esta análise calcula a despesa média para cada `tipo_despesa`. Isso ajud

* ****Método:**** Agrupar os dados por `tipo_despesa` e calcular a média do `valor` para cada tipo.
* ****Resultado:**** Uma tabela mostrando cada tipo de despesa e sua média de valor.
* ****Interpretação:**** Permite identificar os tipos de despesas que consomem mais recursos.

Requisitos:

O código python deve considerar que o arquivo parquet possui o seguinte p

O código deve salvar a tabela final em um arquivo csv com o nome "despesa em ./data/despesa_média_por_tipo_de_despesa.csv

O código deve mostrar o gráfico de barras com a análise.

A saída deve ser somente o código em python sem comentários adicionais.

11

```
genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(prompt3)
print(response.text)
```

```
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

Carrega o arquivo parquet
df = pd.read_parquet('./data/serie_despesas_diárias_deputados.parquet')

Calcula a despesa média por tipo de despesa
despesa_media_por_tipo = df.groupby('tipo_despesa')['valor_documento'].mean().reset_index()

Salva a tabela em um arquivo CSV
despesa_media_por_tipo.to_csv('./data/despesa_média_por_tipo_de_despesa.csv', index=False)

Cria o gráfico de barras
plt.figure(figsize=(12, 6))
sns.barplot(x='tipo_despesa', y='valor_documento', data=despesa_media_por_tipo)
plt.xticks(rotation=45, ha='right')
plt.xlabel('Tipo de Despesa')
plt.ylabel('Despesa Média')
plt.title('Despesa Média por Tipo de Despesa')
plt.tight_layout()
plt.show()

```

```

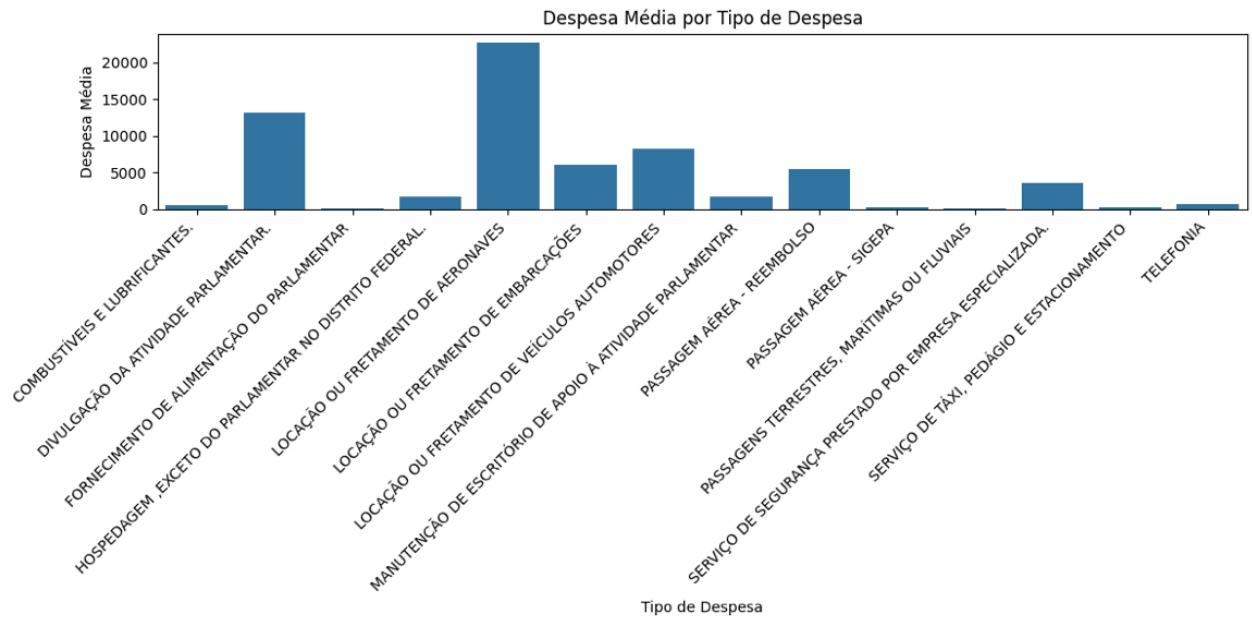
```
In [87]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Carrega o arquivo parquet
df = pd.read_parquet('./data/serie_despesas_diárias_deputados.parquet')

# Calcula a despesa média por tipo de despesa
despesa_media_por_tipo = df.groupby('tipo_despesa')['valor_documento'].mean()

# Salva a tabela em um arquivo CSV
despesa_media_por_tipo.to_csv('./data/despesa_média_por_tipo_de_despesa.csv')

# Cria o gráfico de barras
plt.figure(figsize=(12, 6))
sns.barplot(x='tipo_despesa', y='valor_documento', data=despesa_media_por_tipo)
plt.xticks(rotation=45, ha='right')
plt.xlabel('Tipo de Despesa')
plt.ylabel('Despesa Média')
plt.title('Despesa Média por Tipo de Despesa')
plt.tight_layout()
plt.show()
```



```
In [88]: prompt4 = f""""
```

Contexto

Você é um especialista em ciência de dados.

Eu posso um arquivo parquet com as despesas dos deputados que contém some `id`, `nome`, `tipo_despesa`, `valor_documento`, `data_documento` conforme amostra

{despesas_texto}

Instrução

Favor gerar o código python para a seguinte análise:

3. Distribuição das despesas ao longo do tempo:

Esta análise explora a variação das despesas ao longo do tempo.

* **Método:** Converter a coluna `data_documento` para um formato de data e então agregar as despesas por períodos (ex: mês, trimestre). Podemos c
* **Resultado:** Uma tabela ou gráfico de linha mostrando a evolução das despesas ao longo do tempo. A granularidade (dia, mês, trimestre) deve ser escolhida de acordo com a necessidade.

Requisitos:

O código python deve considerar que o arquivo parquet possui o seguinte schema:

O código deve salvar a tabela final em um arquivo csv com o nome "despesas_ao_longo_do_tempo.csv" em ./data/despesas_ao_longo_do_tempo.csv

O código deve mostrar o gráfico com a análise.

A saída deve ser somente o código em python sem comentários adicionais.

.....

```
genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(prompt4)
print(response.text)
```

```

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

Carrega o arquivo parquet
df = pd.read_parquet('./data/serie_despesas_diárias_deputados.parquet')

Converte a coluna 'data_documento' para datetime
df['data_documento'] = pd.to_datetime(df['data_documento'])

Agrupa as despesas por mês e soma os valores
df['mes'] = df['data_documento'].dt.to_period('M')
despesas_mensais = df.groupby('mes')['valor_documento'].sum().reset_index()

Converte o periodo para datetime para melhor visualização no gráfico
despesas_mensais['mes'] = despesas_mensais['mes'].dt.to_timestamp()

Salva a tabela em um arquivo CSV
despesas_mensais.to_csv('./data/despesas_ao_longo_do_tempo.csv', index=False)

Cria o gráfico de linha
plt.figure(figsize=(12, 6))
sns.lineplot(x='mes', y='valor_documento', data=despesas_mensais)
plt.xlabel('Mês')
plt.ylabel('Total de Despesas')
plt.title('Distribuição das Despesas ao Longo do Tempo')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```
```

```

In [89]:

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

Carrega o arquivo parquet
df = pd.read_parquet('./data/serie_despesas_diárias_deputados.parquet')

Converte a coluna 'data_documento' para datetime
df['data_documento'] = pd.to_datetime(df['data_documento'])

Agrupa as despesas por mês e soma os valores
df['mes'] = df['data_documento'].dt.to_period('M')
despesas_mensais = df.groupby('mes')['valor_documento'].sum().reset_index()

Converte o periodo para datetime para melhor visualização no gráfico
despesas_mensais['mes'] = despesas_mensais['mes'].dt.to_timestamp()

Salva a tabela em um arquivo CSV
despesas_mensais.to_csv('./data/despesas_ao_longo_do_tempo.csv', index=False)

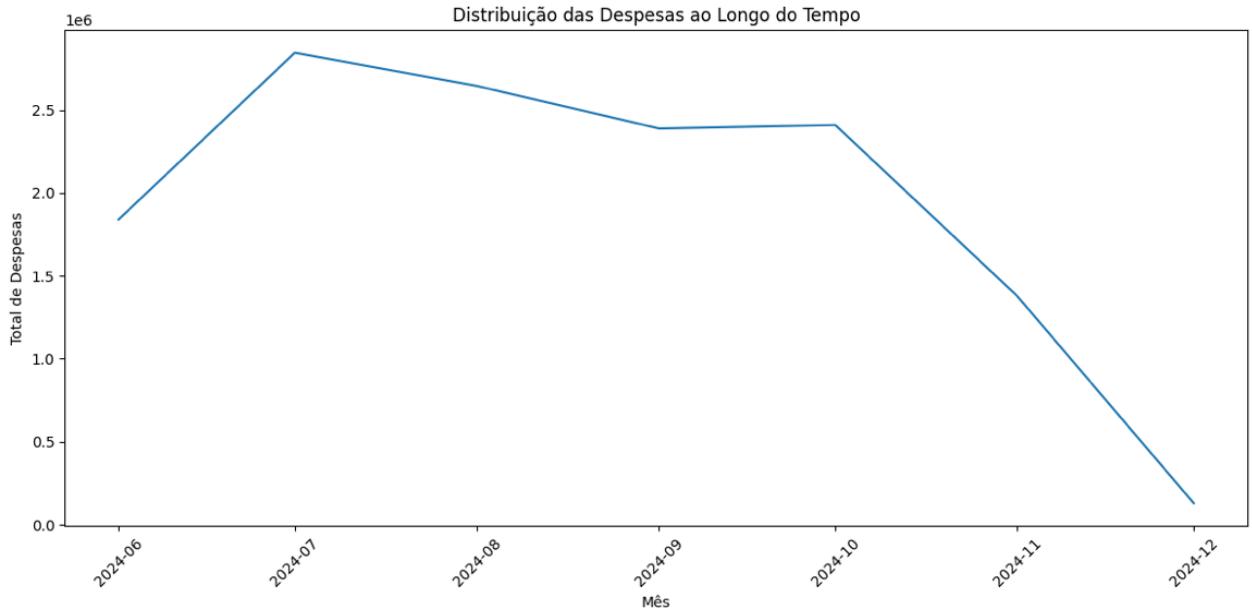
Cria o gráfico de linha

```

```

plt.figure(figsize=(12, 6))
sns.lineplot(x='mes', y='valor_documento', data=despesas_mensais)
plt.xlabel('Mês')
plt.ylabel('Total de Despesas')
plt.title('Distribuição das Despesas ao Longo do Tempo')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



## C

```

In [90]: df_despesas_medias = pd.read_csv("./data/despesa_média_por_tipo_de_despesa.csv")
despesas_medias = str(df_despesas_medias .to_dict())

df_despesas_totais = pd.read_csv("./data/despesa_total_por_deputado.csv")
despesas_totais = str(df_despesas_totais .to_dict())

df_despesas_tempo = pd.read_csv("./data/despesas_ao_longo_do_tempo.csv")
despesas_tempo = str(df_despesas_tempo .to_dict())

```

```

In [94]: generatedKnowledgePrompt = f"""

Contexto

Você é um especialista em ciência de dados.

Em sua resposta considere as análises prévias realizadas com os dados de

1. Despesas médias por tipo de despesa:

{despesas_medias}

2. Despesa total por deputado:

{despesas_totais}

3. Distribuição das despesas ao longo do tempo:

```

```

{despesas_tempo}

Eu posso um arquivo parquet com as despesas dos deputados que contém some
id, nome, tipo_despesa, valor_documento, data_documento conforme amostra

{despesas_texto}

Instrução
Favor gerar insights a partir dos dados fornecidos e análises previamente

Favor retornar a análise em um formato json com a seguinte estrutura:

```json
{
    "insight1": "Insight 1",
    "insight2": "Insight 2",
    "insight3": "Insight 3"
}
```
.....
```
genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(generatedKnowledgePrompt)
print(response.text)

```json
{
 "insight1": "As despesas com 'DIVULGAÇÃO DA ATIVIDADE PARLAMENTAR' apresentam a maior média (R$ 13.072,26), indicando uma alta concentração de recursos nessa área. É necessário investigar se esse valor está alinhado com as práticas de transparência e se há uma distribuição equitativa entre os deputados.",
 "insight2": "Há uma grande disparidade nas despesas totais por deputado, variando de valores muito baixos a valores extremamente altos (como o deputado Alberto Fraga com R$ 180.949,25). Essa discrepância exige uma análise mais aprofundada para entender as causas, considerando fatores como tamanho da equipe, região representada e tipo de atividade parlamentar desempenhada. A análise de outliers pode ser crucial para identificar possíveis anomalias.",
 "insight3": "As despesas demonstram uma sazonalidade, com picos nos meses de julho e agosto de 2024, e uma queda significativa em novembro e dezembro. Isso sugere a possibilidade de fatores externos influenciando as despesas, como o calendário político ou a disponibilidade de recursos orçamentários. Análises mais detalhadas, considerando a desagregação das despesas por tipo e deputado ao longo do tempo, são necessárias para confirmar esta hipótese e identificar possíveis tendências."
}
```

```

```

In [95]: clean_response= response.text.replace(```json\n`, '').replace("\n```", '')
insights = json.loads(clean_response)

with open("./data/insights_despesas_deputados.json", "w") as file:

```

```
json.dump(insights, file, indent=4, ensure_ascii=False)
```

Exercício 5

a - código dentro do arquivo dataprep.py

b

```
In [2]: df_proposicoes = pd.read_parquet("./data/proposicoes_deputados.parquet")  
proposicoes = str(df_proposicoes[['ementa','tema','id']].to_dict())
```

```
In [3]: summary_prompt = f"""\n    ### Contexto\n    Você é um assessor parlamentar que deve preparar um resumo das proposições\n    ### Instrução\n    Resuma as proposições dos deputados constantes no seguintes dados:\n    {proposicoes}
```

Você deve retornar a resposta em um formato JSON com a seguinte estrutura

```
```json\n    "id1": "Resumo 1",\n    "id2": "Resumo 2",\n    "id3": "Resumo 3"\n    ...  
....\n\nproposition_summarizer = ChunkSummary(\n model_name = "gemini-1.5-flash",\n apikey = os.environ["GEMINI_KEY"],\n text = summary_prompt,\n window_size = 100,\n overlap_size = 25\n)\n\nproposition_summary = proposition_summarizer.summarize()
```

```
Summarizing text
Summarizing chunk 1 from 1
Final summarization
```

```
In [5]: clean_summary= proposition_summary.replace("```json\\n", '').replace("\\n```
summary = json.loads(clean_summary)
```

```
with open("./data/sumarizacao_proposicoes.json", "w") as file:
 json.dump(summary, file, indent=4, ensure_ascii=False)
```

## Exercício 6

```
In [18]: # Criar aplicação streamlit
chain_of_thoughts_1 = """

Objetivo
Criar código de uma aplicação streamlit sobre dados de deputados da Câmara com 3 abas: Overview, Despesas e Proposições

Prompt: "Forneça um código em python de uma aplicação streamlit com 3 abas: Overview, Despesas e Proposições
.....

Aba Overview
chain_of_thoughts_2 = """

Objetivo
Criar código de uma aplicação streamlit sobre dados de deputados da Câmara

Prompt: "Forneça um código em python para a aba de Overview que possua um geral da aplicação streamlit, informando as abas, o objetivo e o contexto
.....

Texto summarizado
chain_of_thoughts_3 = """

Objetivo
Ler o arquivo com o texto summarizado sobre a Câmara dos deputados e inserir

Prompt: "Forneça um código em python para a aba de Overview que leia o arquivo ./data/config.yaml, obtenha o texto da chave overview_summary e exiba esse resumo da Câmara dos Deputados.
.....

Criar aplicação com grafico pizza
chain_of_thoughts_4 = """

Objetivo
Inserir imagem da distribuição dos Deputados por partido na aplicação, na

Prompt: "Forneça um código em python para a aba de Overview que leia o arquivo ./docs/distribuicao_deputados.png, e insira essa imagem na aba de Overview
.....

Insights
```

```
chain_of_thoughts_5 = """
Objetivo
Ler o arquivo com os insights sobre as despesas dos deputados e inserir e
```

Prompt:

leia o arquivo no path ./data/insights\_despesas\_deputados.json, obtenha os insights e exiba esses insights na aba de Overview

o json possui a seguinte estrutura:

```
```json  
{  
    "insight1": "Insight 1",  
    "insight2": "Insight 2",  
    "insight3": "Insight 3"  
}  
...  
....
```

```
In [16]: genai.configure(api_key=os.environ["GEMINI_KEY"])  
model = genai.GenerativeModel("gemini-1.5-flash")  
response = model.generate_content(chain_of_thoughts_1)  
print(response.text)
```

Este código requer que você tenha instalado as bibliotecas `streamlit`, `pandas` e `requests`. Você pode instalá-las usando `pip install streamlit pandas requests`. A API da Câmara dos Deputados pode mudar, então verifique a documentação deles para garantir que os endpoints ainda são válidos. Este exemplo usa dados simulados, pois acesso direto à API requer autenticação e manipulação de grandes conjuntos de dados que podem ser complexos para um exemplo.

```
```python  
import streamlit as st
import pandas as pd
import requests # Necessário para acessar APIs (se usar dados reais)

Dados simulados – substitua por dados reais da API da Câmara dos Deputados
overview_data = {
 'Deputado': ['João Silva', 'Maria Santos', 'Pedro Souza'],
 'Partido': ['PT', 'PSDB', 'MDB'],
 'Estado': ['SP', 'RJ', 'MG'],
}

despesas_data = {
 'Deputado': ['João Silva', 'Maria Santos', 'Pedro Souza', 'João Silva'],
 'Data': ['2024-01-15', '2024-01-20', '2024-01-25', '2024-02-10'],
 'Descrição': ['Passagens Aéreas', 'Diárias', 'Combustível', 'Telefonia'],
 'Valor': [1000, 500, 200, 150],
```

```

}

proposicoes_data = {
 'Deputado': ['João Silva', 'Maria Santos', 'Pedro Souza'],
 'Tipo': ['Projeto de Lei', 'Requerimento', 'Sugestão Legislativa'],
 'Número': ['PL-1234', 'REQ-5678', 'SL-9012'],
 'Assunto': ['Educação', 'Saúde', 'Meio Ambiente'],
}

overview_df = pd.DataFrame(overview_data)
despesas_df = pd.DataFrame(despesas_data)
proposicoes_df = pd.DataFrame(proposicoes_data)

#Função para obter dados da API (a ser implementada se usar dados reais)
def get_data_from_api(endpoint):
response = requests.get(endpoint)
response.raise_for_status() # Raise HTTPError for bad responses (4xx or 5xx)
return response.json()

st.set_page_config(page_title="Dados Deputados", page_icon=":bar_chart:", layout="wide")

st.title("Dados da Câmara dos Deputados")

selected_tab = st.sidebar.radio("Selecione uma aba:", ("Overview", "Despesas", "Proposições"))

if selected_tab == "Overview":
 st.header("Overview dos Deputados")
 st.dataframe(overview_df)

elif selected_tab == "Despesas":
 st.header("Despesas dos Deputados")
 st.dataframe(despesas_df)
 # Adicione aqui gráficos ou outras visualizações das despesas

elif selected_tab == "Proposições":
 st.header("Proposições dos Deputados")
 st.dataframe(proposicoes_df)
 # Adicione aqui gráficos ou outras visualizações das proposições

```

```

Lembre-se de substituir os dados simulados pelos dados reais obtidos da API da Câmara dos Deputados. Você precisará consultar a documentação da API para entender como acessar e formatar os dados. A adição de gráficos (usando `st.bar_chart`, `st.line_chart`, etc.) e outras visualizações tornará a aplicação mais informativa. Considere também a adição de filtros e recursos de busca para uma melhor experiência do usuário.

```
In [19]: genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(chain_of_thoughts_2)
```

```
print(response.text)

```python
import streamlit as st

def overview_page():
 """
 Cria a aba Overview da aplicação Streamlit.
 """
 st.title("Análise de Dados de Deputados da Câmara dos Deputados")

 st.markdown(
 """
 Esta aplicação Streamlit visa facilitar a análise de dados de deputados da Câmara dos Deputados.
 Através de uma interface intuitiva, você poderá explorar informações relevantes sobre os parlamentares, permitindo uma compreensão mais aprofundada do cenário político brasileiro.
 """

 Objetivo:

O principal objetivo desta aplicação é disponibilizar dados públicos de forma acessível e organizada, facilitando a análise e a compreensão do trabalho dos deputados. Isso contribui para a transparência e o engajamento cívico, permitindo a população um melhor acompanhamento das atividades parlamentares.

 Contexto:

Os dados utilizados nesta aplicação são provenientes de fontes públicas e oficiais da Câmara dos Deputados. A aplicação está em constante desenvolvimento e novas funcionalidades serão adicionadas futuramente.

 Abas disponíveis:

- **Overview:** Esta página fornece uma visão geral da aplicação.
- **Dados Demográficos:** (Adicionar mais abas conforme desenvolvimento) Explore dados demográficos dos deputados, como idade, gênero, partido e estado.
- **Despesas:** (Adicionar mais abas conforme desenvolvimento) Analise as despesas dos deputados, incluindo informações detalhadas sobre verbas e gastos.
- **Proposições:** (Adicionar mais abas conforme desenvolvimento) Acompanhe a atividade legislativa dos deputados, com informações sobre proposições apresentadas e votadas.

)
```

#Exemplo de como integrar a função na aplicação Streamlit completa:

```

if __name__ == "__main__":
 st.set_page_config(page_title="Análise de Dados de Deputados", page_ic
on=":bar_chart:", layout="wide")

 pages = {
 "Overview": overview_page,
 # Adicione outras páginas aqui... ex: "Dados Demográficos": demog
raphic_page, etc.
 }

 selected_page = st.sidebar.radio("Navegue pelas páginas:", tuple(page
s.keys()))
 pages[selected_page]()
```

```

Para usar este código:

1. **Instale o Streamlit:** `pip install streamlit`
2. **Salve o código:** Salve o código acima como um arquivo Python (ex: `app.py`).
3. **Execute o Streamlit:** Abra seu terminal, navegue até o diretório onde salvou o arquivo e execute `streamlit run app.py`.

Isso abrirá a aplicação Streamlit no seu navegador. Você verá a aba "Overview" com o título, descrição e as informações sobre as outras abas (que ainda precisam ser implementadas). Você precisará adicionar o código para as outras abas ("Dados Demográficos", "Despesas", "Proposições") para que a aplicação fique completa. Lembre-se de incluir a leitura e tratamento dos dados da Câmara dos Deputados nessas outras abas. Considere usar bibliotecas como `pandas` para manipulação de dados.

```
In [20]: genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(chain_of_thoughts_3)
print(response.text)
```

Este código lê o arquivo YAML, extrai o texto do resumo e o exibe. Ele assume que você está usando uma biblioteca GUI como Tkinter ou PyQt para criar a aba "Overview". Adapte o código de exibição de acordo com a sua biblioteca GUI específica.

****1. Instalando PyYAML:****

Você precisará instalar a biblioteca PyYAML para ler arquivos YAML. Use o pip:

```
```bash
pip install pyyaml
```

```

****2. Código Python (com Tkinter):****

```
```python
import yaml
import tkinter as tk
from tkinter import scrolledtext
```

```

def load_overview_summary(filepath):
 """Carrega o resumo do arquivo YAML."""
 try:
 with open(filepath, 'r', encoding='utf-8') as file:
 config = yaml.safe_load(file)
 return config.get('overview_summary', "Resumo não encontrado.") # Retorna mensagem padrão se a chave não existir
 except FileNotFoundError:
 return "Arquivo de configuração não encontrado."
 except yaml.YAMLError as e:
 return f"Erro ao ler o arquivo YAML: {e}"

```

```

def create_overview_tab(root, filepath="./data/config.yaml"):
 """Cria a aba Overview com o resumo."""
 overview_tab = tk.Frame(root)
 overview_tab.pack(fill="both", expand=True)

 summary_text = load_overview_summary(filepath)

 text_area = scrolledtext.ScrolledText(overview_tab, wrap=tk.WORD, width=80, height=20)
 text_area.insert(tk.END, summary_text)
 text_area.config(state=tk.DISABLED) # Impede edição do texto
 text_area.pack(expand=True, fill="both")

```

```

#Exemplo de uso:
root = tk.Tk()
root.title("Câmara dos Deputados")
create_overview_tab(root)
root.mainloop()
```

```

3. Arquivo config.yaml (./data/config.yaml):

Crie um arquivo chamado `config.yaml` na pasta `./data/` com o seguinte conteúdo:

```

```yaml
overview_summary: |
 Este é um resumo da Câmara dos Deputados.
 Aqui você pode inserir informações relevantes sobre a Câmara,
 suas funções, composição, e histórico.
 Adicione mais detalhes conforme necessário.
```

```

Explicação do código:

- * **`load_overview_summary()`:** Lê o arquivo YAML, utilizando `yaml.safe_load()` para segurança. Trata erros de arquivo não encontrado e erros do YAML. Retorna o texto do resumo ou uma mensagem de erro.
- * **`create_overview_tab()`:** Cria um frame para a aba Overview e usa `scrolledtext.ScrolledText` para exibir o texto do resumo em uma área de text

o com barra de rolagem. O `state=tk.DISABLED` impede que o usuário edite o texto.
* **Exemplo de uso:** Cria a janela principal do Tkinter e chama `create_overview_tab` para adicionar a aba Overview.

Para usar com outra biblioteca GUI (ex: PyQt):

Você precisará adaptar a parte `create_overview_tab()`. Em vez de `tkinter`, use os widgets da sua biblioteca escolhida para criar a aba e exibir o texto. A função `load_overview_summary()` permanece a mesma. Por exemplo, com PyQt você poderia usar um `QTextEdit`.

Lembre-se de ajustar o caminho do arquivo `config.yaml` se ele estiver em um local diferente. Este exemplo fornece uma base sólida; você pode expandir para incluir recursos adicionais, como formatação de texto, links e imagens na sua aba de Overview.

```
In [21]: genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(chain_of_thoughts_4)
print(response.text)
```

O código Python para inserir uma imagem na aba "Overview" depende totalmente da estrutura e da biblioteca que você está usando para construir sua aplicação. Não existe uma solução universal em Python para isso. Precisamos de mais informações sobre seu framework (ex: Streamlit, Dash, Flask, Tkinter, etc.).

No entanto, posso te fornecer exemplos para alguns frameworks populares:

1. Streamlit:

Streamlit é ótimo para construir rapidamente aplicações de visualização de dados.

```
```python
import streamlit as st
from PIL import Image

st.title("Distribuição de Deputados por Partido")

image = Image.open("./docs/distribuicao_deputados.png")
st.image(image, caption="Distribuição de Deputados por Partido", use_column_width=True)
````
```

Este código abre a imagem usando a biblioteca PIL (Pillow) e então a exibe no Streamlit usando `st.image`. `use_column_width=True` faz com que a imagem ocupe toda a largura da coluna.

2. Dash (Plotly):

Dash é mais robusto e permite criar aplicações mais complexas.

```
```python
import dash
from dash import html, dcc
from dash.dependencies import Input, Output

app = dash.Dash(__name__)

app.layout = html.Div([
 html.H1("Distribuição de Deputados por Partido"),
 html.Img(src="./docs/distribuicao_deputados.png", alt="Distribuição de Deputados", style={'width': '100%'}),
])

if __name__ == '__main__':
 app.run_server(debug=True)
```
```

Aqui, usamos `html.Img` para incorporar a imagem diretamente no layout do Dash. `style={'width': '100%'}` garante que a imagem se ajuste à largura do container.

3. Flask (com um pouco mais de trabalho):

Flask é um framework web mais genérico, então você precisará renderizar um template HTML.

```
```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def overview():
 return render_template('overview.html')

if __name__ == "__main__":
 app.run(debug=True)
```
```

Você precisaria criar um arquivo `templates/overview.html`:

```
```html
<h1>Distribuição de Deputados por Partido</h1>

```
```

Você precisará colocar a imagem em uma pasta `static` na raiz do seu projeto Flask.

Lembre-se: Substitua `"./docs/distribuicao_deputados.png"` pelo camin

ho correto para sua imagem. Você pode precisar instalar as bibliotecas necessárias: `pip install streamlit pillow dash` ou apenas `pip install flask`.

Para usar qualquer um desses exemplos, você precisará adaptar o código ao seu projeto existente, considerando a estrutura de pastas e o funcionamento da sua aplicação. Forneça mais detalhes sobre seu framework para obter uma resposta mais precisa.

```
In [22]: genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(chain_of_thoughts_5)
print(response.text)
```

Este código assume que você está usando uma biblioteca de interface de usuário (UI) e que você tem uma função `updateOverviewTab` que aceita uma lista de insights como entrada e os exibe na aba Overview. Adapte o código à sua biblioteca UI específica. Também assume que o arquivo JSON existe e está corretamente formatado. Tratamento de erros (arquivo não encontrado, JSON inválido) deve ser adicionado para uma solução robusta.

```
```python
import json

def update_overview_tab(filepath="./data/insights_despesas_deputados.json"):
 """
 Lê o arquivo JSON, extrai os insights e atualiza a aba Overview.
 """
 try:
 with open(filepath, 'r', encoding='utf-8') as f: # Encoding especificado para lidar com caracteres especiais
 data = json.load(f)
 insights = list(data.values()) # Obtém todos os valores (insights) do dicionário

 # Aqui, adapte para sua biblioteca UI
 # Exemplo genérico, substitua updateOverviewTab pela função correta da sua biblioteca
 updateOverviewTab(insights)

 except FileNotFoundError:
 print(f"Erro: Arquivo não encontrado em {filepath}")
 # Lidar com o erro, por exemplo, exibindo uma mensagem de erro na UI
 updateOverviewTab(["Erro: Arquivo não encontrado"]) # Exemplo de tratamento de erro na UI
 except json.JSONDecodeError:
 print(f"Erro: Arquivo JSON inválido em {filepath}")
 updateOverviewTab(["Erro: Arquivo JSON inválido"]) # Exemplo de tratamento de erro na UI
 except Exception as e:
 print(f"Erro inesperado: {e}")
 updateOverviewTab(["Erro inesperado"]) # Exemplo de tratamento de erro na UI
```

```
Exemplo de uso:
update_overview_tab()
```
```

```
**Exemplo com bibliotecas UI populares (adapte ao seu caso):**
```

```
**Tkinter (Python):**
```

```
```python  
import tkinter as tk
import json

def updateOverviewTab(insights):
 overview_text.delete("1.0", tk.END) #Limpa o texto anterior
 for insight in insights:
 overview_text.insert(tk.END, insight + "\n")

root = tk.Tk()
overview_text = tk.Text(root)
overview_text.pack()

update_overview_tab()

root.mainloop()
```
```

```
**PyQt5:**
```

```
```python  
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QLabel
import json

def updateOverviewTab(insights):
 for insight in insights:
 label = QLabel(insight)
 layout.addWidget(label)
 window.setLayout(layout)
 window.show()

app = QApplication(sys.argv)
window = QWidget()
layout = QVBoxLayout()

update_overview_tab()

sys.exit(app.exec_())
```
```

Lembre-se de instalar as bibliotecas necessárias (`pip install PyQt5` ou já ter o Tkinter, que é padrão no Python). Substitua os exemplos de `updateOverviewTab` pelo método correto da sua biblioteca UI para atualizar a aba Overview com a lista de insights. A estrutura geral permanece a mesma. É crucial o tratamento de exceções para lidar com possíveis erros na leitura ou processamento do arquivo JSON.

Exercício 7

```
In [25]: df_despesas = pd.read_parquet("./data/serie_despesas_diárias_deputados.parquet")
despesas_texto = str(df_despesas.to_dict())

deputados = pd.read_parquet("./data/serie_despesas_diárias_deputados.parquet")
deputados
```

Out[25]:

| | id | nome | tipo_despesa | valor_documento | data_documento |
|------|-----------|----------------|---|------------------------|-----------------------|
| 0 | 220593 | Abilio Brunini | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 149.90 | 2024-08-21T00:00:00 |
| 1 | 220593 | Abilio Brunini | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 149.00 | 2024-10-05T00:00:00 |
| 2 | 220593 | Abilio Brunini | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 149.90 | 2024-09-07T00:00:00 |
| 3 | 220593 | Abilio Brunini | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 149.90 | 2024-07-13T00:00:00 |
| 4 | 220593 | Abilio Brunini | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 1050.00 | 2024-06-24T00:00:00 |
| ... | ... | ... | ... | ... | ... |
| 7743 | 220552 | Zucco | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 720.00 | 2024-10-09T00:00:00 |
| 7744 | 220552 | Zucco | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 81.00 | 2024-06-18T00:00:00 |
| 7745 | 220552 | Zucco | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 81.00 | 2024-07-31T00:00:00 |
| 7746 | 220552 | Zucco | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 1210.99 | 2024-08-21T00:00:00 |
| 7747 | 220552 | Zucco | MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE | 679.80 | 2024-09-17T00:00:00 |

7748 rows × 5 columns

```
In [30]: batchPrompt = f"""\n
```

Objetivo
Criar o código de uma aplicação streamlit sobre dados de deputados da Câmara

Parte do código já está feito. Precisamos preenchê-lo com as informações

Prompt: "1. Forneça um código que na Aba despesas faça o seguinte:

1.1. Leia o arquivo no path ./data/insights_despesas_deputados.json

o json possui a seguinte estrutura:

```
```json
```

```
"insight1": "Insight 1",
"insight2": "Insight 2",
"insight3": "Insight 3"
```

1.2. Leia o arquivo no path ./data/serie\_despesas\_diárias\_deputada a partir da coluna nome crie um st.selectbox para seleção do depu

1.3. Crie um gráfico de barras com a série temporal usando matplotlib a partir do st.selectbox. Segue amostra dos dados que devem ser l

```
{despesas_texto}
```

2. Forneça um código que na Aba proposições faça o seguinte:

2.1 Leia o arquivo no path data/proposicoes\_deputados.parquet e e como um dataframe

2.2. Leia o arquivo no path ./data/sumarizacao\_proposicoes.json e o json possui a seguinte estrutura:

```
```json
```

```
"id1": "Resumo 1",
"id2": "Resumo 2",
"id3": "Resumo 3"
```

```
....
```

```
.....
```

```
In [31]: genai.configure(api_key=os.environ["GEMINI_KEY"])
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content(batchPrompt)
print(response.text)
```

```
```python
```

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import json
```

```

import pyarrow.parquet as pq

st.set_page_config(page_title="Dados Deputados", page_icon=":bar_chart:", layout="wide")

--- DESPESAS ---
st.markdown("<h1 style='text-align: center;'>Despesas</h1>", unsafe_allow_html=True)
col1, col2 = st.columns(2)

with col1:
 st.subheader("Insights Gerais:")
 try:
 with open("./data/insights_despesas_deputados.json", "r") as f:
 insights_despesas = json.load(f)
 for key, value in insights_despesas.items():
 st.write(f"{key}: {value}")
 except FileNotFoundError:
 st.error("Arquivo insights_despesas_deputados.json não encontrado.")

with col2:
 st.subheader("Série Temporal de Despesas:")
 try:
 df_despesas = pq.read_table("./data/serie_despesas_diárias_deputados.parquet").to_pandas()
 deputados = df_despesas['nome'].unique()
 selected_deputado = st.selectbox("Selecione o Deputado:", deputados)

 df_deputado = df_despesas[df_despesas['nome'] == selected_deputado]
 df_deputado['data_documento'] = pd.to_datetime(df_deputado['data_documento'])
 df_deputado = df_deputado.sort_values('data_documento')

 fig, ax = plt.subplots(figsize=(10,6))
 ax.bar(df_deputado['data_documento'], df_deputado['valor_documento'])
 ax.set_xlabel("Data")
 ax.set_ylabel("Valor")
 ax.set_title(f"Despesas do Deputado {selected_deputado}")
 ax.tick_params(axis='x', rotation=45)
 st.pyplot(fig)

 except FileNotFoundError:
 st.error("Arquivo serie_despesas_diárias_deputados.parquet não encontrado.")
 except Exception as e:
 st.exception(e)

--- PROPOSIÇÕES ---

```

```

st.markdown("<h1 style='text-align: center;'>Proposições</h1>", unsafe_allow_html=True)
col3, col4 = st.columns(2)

with col3:
 st.subheader("Dados das Proposições:")
 try:
 df_proposicoes = pq.read_table("./data/proposicoes_deputados.parquet").to_pandas()
 st.dataframe(df_proposicoes)
 except FileNotFoundError:
 st.error("Arquivo proposicoes_deputados.parquet não encontrado.")

with col4:
 st.subheader("Sumarização das Proposições:")
 try:
 with open("./data/sumarizacao_proposicoes.json", "r") as f:
 summarizacao_proposicoes = json.load(f)
 for key, value in summarizacao_proposicoes.items():
 st.write(f"{key}: {value}")
 except FileNotFoundError:
 st.error("Arquivo summarizacao_proposicoes.json não encontrado.")

```

```

****Para executar este código:****

1. **Instale as bibliotecas necessárias:** `pip install streamlit pandas matplotlib pyarrow`
2. **Crie a estrutura de pastas:** Crie as pastas `data` dentro do mesmo diretório onde você salvará o script Python.
3. **Coloque os arquivos de dados:** Coloque os arquivos `insights_despesas_deputados.json`, `serie_despesas_diárias_deputados.parquet`, `proposicoes_deputados.parquet`, e `sumarizacao_proposicoes.json` dentro da pasta `data`. Certifique-se de que os dados estejam no formato correto (JSON e Parquet).
4. **Execute o Streamlit:** `streamlit run seu_nome_do_arquivo.py` (substitua `seu_nome_do_arquivo.py` pelo nome do seu arquivo Python).

Lembre-se de adaptar os caminhos dos arquivos caso necessário. Se você tiver problemas com a leitura dos arquivos parquet, verifique se a versão do pyarrow é compatível com a versão do pandas e do streamlit. Pode ser necessário instalar uma versão específica do pyarrow para resolver problemas de compatibilidade.

Resposta

O resultado do código gerado pela técnica batch-prompting se mostrou mais adequado do que o gerado pelo técnica chain-of-thoughts. Acredito que devido ao tamanho dos prompts, a complexidade da tarefa e ao modelo escolhido (mais simples), o LLM não soube lidar muito bem com o contexto envolvido e nem reconhecer com precisão os prompts anteriores. Por sua vez, quando todas as

instruções foram colocadas em somente um prompt, ele capturou o contexto mais facilmente e gerou um código mais assertivo.

Exercício 8

a

A técnica self-ask pode ser utilizada nesse contexto para poder indicar ao LLM um caminho de raciocínio que facilite a obtenção da resposta desejada a partir da divisão em subperguntas a serem resolvidas sequencialmente. Isso se torna bastante importante no contexto do exercício uma vez que temos diferentes bases de dados com diferentes informações, assim o self ask pode ajudar o modelo a entender mais facilmente como deve proceder para chegar a uma resposta desejada

b

1.

Dados da Câmara dos Deputados

Overview Despesas Proposições Chatbot

Chatbot Câmara dos Deputados

Hi >

Qual é o partido político com mais deputados na câmara?

O banco de dados fornecido contém informações sobre deputados e suas respectivas siglas partidárias, mas não apresenta o número total de deputados por partido. Portanto, não é possível responder à pergunta "Qual é o partido político com mais deputados na câmara?" com base apenas nesses dados. Seriam necessárias informações adicionais, como a contagem de deputados por partido, para responder corretamente.

O modelo não conseguiu identificar o partido com mais deputados na câmara. Isso ocorreu pois o número (k) de resultados mais próximos do vetor da pergunta não foi o suficiente para obter resultados que o LLM pudesse interpretar para obter essa resposta apesar de termos estipulado $k = 100$ de forma a aumentar a precisão das respostas. Valores maiores deixaram a aplicação significativamente mais lenta.

2.

Dados da Câmara dos Deputados

Overview Despesas Proposições Chatbot

Chatbot Câmara dos Deputados

Hi



Qual é o deputado com mais despesas na câmara?



Baseado nos dados fornecidos, o deputado com mais despesas é **Gustinho Ribeiro**, com um valor total de despesas de 77571.13. É importante notar que o dataset contém apenas o valor total das despesas e não detalha o tipo de despesa para todos os deputados.

O modelo não conseguiu identificar o deputado com a maior despesa corretamente. Isso ocorreu pois o número (k) de resultados mais próximos do vetor da pergunta não foi o suficiente para obter resultados que o LLM pudesse interpretar para obter essa resposta apesar de termos estipulado $k = 100$ de forma a aumentar a precisão das respostas. Valores maiores deixaram a aplicação significativamente mais lenta. Nesse caso, para saber qual o deputado mais gastou, não basta um resultado específico, não basta um resultado probabilístico/estocástico, característico desse tipo de aplicação, seria necessário um modelo determinístico que tivesse acesso a todas as informações coletadas. Devido as limitações de processamento local e da API do google estipulamos esse limite como $k=100$, para que o LLM conseguisse responder essa pergunta com precisão, precisaria ter acesso a no mínimo 519 valores com as despesas de todos os deputados.

3.

Dados da Câmara dos Deputados

Overview Despesas Proposições Chatbot

Chatbot Câmara dos Deputados

|Hi



Qual é o tipo de despesa mais declarada pelos deputados da câmara?



Baseado nos dados fornecidos, o tipo de despesa mais declarado pelos deputados é **COMBUSTÍVEIS E LUBRIFICANTES**. Há um número significativamente maior de registros para esta categoria em comparação com "PASSAGEM AÉREA - SIGEPA" e "DIVULGAÇÃO DA ATIVIDADE PARLAMENTAR".

O modelo conseguiu identificar corretamente uma vez que exige menor precisão para responder a essa pergunta.

4.

Chatbot Câmara dos Deputados

Hi



Quais são as informações mais relevantes sobre as proposições que falam de Economia?



As proposições relacionadas à Economia no banco de dados tratam dos seguintes assuntos:

- **Suspensão temporária de pagamento de dívidas de municípios em situação de emergência ou calamidade pública:** (PLP 249/2007) Altera a Lei de Responsabilidade Fiscal para permitir essa suspensão.
- **Alteração da Lei nº 5.764/1971:** (PL 6692/2013) Modifica a lei para permitir que cargos de diretoria em cooperativas sejam ocupados por não associados.
- **Aumento de repasses de receitas de loterias para Previdência Social, Assistência Social e SUS:** (PL 3127/2015) Propõe aumentar a porcentagem destinada a estes setores.
- **Priorização no processamento de boletos bancários:** (PL 7474/2017) Propõe estabelecer a cronologia da percepção de pagamentos como critério preferencial na liquidação de boletos.
- **Prevenção da utilização do sistema financeiro para ilícitos:** (PL 7877/2017) Altera a Lei nº 9.613/1998 para tornar mais eficiente a prevenção.
- **Destinação da renda líquida da Mega Sena para a Federação Brasileira das Associações de Síndrome de Down:** (PL 6987/2017) Propõe a destinação específica desses recursos.
- **Alteração da data de vencimento de contratos de crédito:** (PL 9255/2017) Permite ao consumidor alterar a data de vencimento de um contrato de crédito.
- **Modificação da Lei nº 7.827/1989:** (PL 532/2015) Modifica artigos da lei, sem especificar o conteúdo detalhado da alteração.
- **Destinação de 1% da arrecadação das Loterias Federais para o INCA:** (PL 3987/2015) Propõe a destinação de recursos para o Instituto Nacional do Câncer.

As demais proposições com o tema "Economia" listadas, apesar de se relacionarem com o tema, não apresentam informações suficientes em suas emendas para uma descrição mais detalhada. A maioria das proposições com tema "Economia" parece focar em redirecionamento de recursos públicos ou alterações em legislações relacionadas ao sistema financeiro e arrecadação.

O modelo conseguiu identificar corretamente os dados dentro do RAG visto que são poucos registros dentro do total. Assim, com uma precisão menor, dentro dos k = 100, foi possível isolar esses casos e trazer a resposta correta

Chatbot Câmara dos Deputados

Hi



Quais são as informações mais relevantes sobre as proposições que falam de 'Ciência, Tecnologia e Inovação'?



As proposições relacionadas a "Ciência, Tecnologia e Inovação" no banco de dados abordam diversos aspectos:

- **Incentivos fiscais:** Há proposições que buscam reduzir impostos na aquisição de equipamentos de informática para famílias de baixa renda (PL 486/2019) e ampliar o crédito financeiro para investimentos em P&D&I em setores específicos, como tecnologias da informação e comunicação e semicondutores (PL 13/2020). Este último também propõe a criação de um programa nacional para o setor de semicondutores ("Brasil Semicon"). Outra proposição (PL 3946/2012) visa permitir a aquisição de produtos ou processos inovadores gerados por políticas de fomento à pesquisa e desenvolvimento.
- **Financiamento de pesquisa:** Algumas proposições tratam do financiamento de pesquisas em áreas específicas, como o desenvolvimento de diagnósticos, vacinas e medicamentos para o tratamento de epidemias e câncer (PL 4812/2016). Outra propõe a instituição de um prêmio para incentivar a pesquisa e a aplicação de conhecimentos e tecnologia para o desenvolvimento humano (PL 1872/2011).
- **Inovação e tecnologia:** Há propostas que focam no incentivo e desenvolvimento de startups e empresas de inovação e tecnologia (PL 3432/2019), e na extensão de benefícios da Lei de Informática a jogos eletrônicos (PL 899/2011). Uma proposição também busca condicionar o acesso aos benefícios da Lei de Informática à inclusão de ferramentas de tradução para Libras em dispositivos tecnológicos (PL 3503/2019).
- **Aplicação de recursos em Ciência e Tecnologia:** Uma proposição (PL 691/2007) altera uma lei para determinar a aplicação de recursos do Fundo de Universalização dos Serviços de Telecomunicações em educação e ciência e tecnologia.
- **Modernização de Leis:** Há propostas que visam alterar leis existentes para integrar aspectos de ciência, tecnologia e inovação, como a PL 8252/2014, que altera as Leis nºs 10.973/2004, 8.666/1993 e 12.462/2011. A ementa não detalha as alterações propostas.

Em resumo, as proposições demonstram um interesse em fomentar a Ciência, Tecnologia e Inovação através de incentivos financeiros, direcionamento de recursos, e atualização da legislação para contemplar as novas tecnologias e demandas do setor. A falta de detalhes nas ementas de algumas proposições dificulta uma análise mais aprofundada.

O modelo conseguiu identificar corretamente os dados dentro do RAG visto que são poucos registros dentro do total. Assim, com uma precisão menor, dentro dos k = 100, foi possível isolar esses casos e trazer a resposta correta

Exercício 9

a

Stable Diffusion

Arquitetura e Funcionamento:

Baseado no modelo de difusão latente para gerar imagens. A ideia é começar a imagem com um ruído e ao longo de etapas incrementais ir ajustando a imagem ao desejado pelo usuário.

Vantagens:

Modelo open-source

Exige menos requisitos computacionais para operar.

Pode ser executado localmente, permitindo que seja feito um fine tuning.

Desvantagens:

Ainda exige grande poder de processamento, dependendo de GPUs

Menor capacidade de gerar imagens complexas que necessitem de pequenos ajustes.

Configuração mais complexa.

DALL-E

Arquitetura e Funcionamento:

Baseado em transformers que prevêem os tokens visuais (pequenas partes da imagem) a partir dos tokens de texto (prompt)

Vantagens:

A combinação entre imagem e texto permite imagens que capturam muito bem o que foi solicitado.

Possui uma boa capacidade para gerar imagens complexas respondendo bem a pequenos ajustes

Desvantagens:

Custo para utilizar em escala, visto que é proprietário

Menos refinado que o MidJourney

Mid Journey

Arquitetura e Funcionamento:

Baseado em arquitetura híbrida e proprietária com alto foco na estética da arte.

Vantagens:

Produz imagens mais refinadas e bonitas.

Possui integração com o discord o que facilita o uso.

Desvantagens:

Funciona somente via discord

Menor precisão para descrições técnicas detalhadas.

b

```
In [25]: with open("./data/sumarizacao_proposicoes.json", "r") as file:  
    summary = json.load(file)  
  
prop_1 = "create na imagem based on this: " + summary['293703']  
prop_2 = "create na imagem based on this: " + summary['498056']  
  
prop_1, prop_2
```

```
Out[25]: ('create na imagem based on this: Inclui problemas do consumo de drogas  
no currículo escolar.',  
 'create na imagem based on this: Inclui bolsas de pós-graduação no PROU  
NI.')
```

```
In [28]: prompt_1 = f"""\n  
Não elabora imagens muito complexas,  
  
Usando um estilo visual similar ao de van gogh e com uma composição que re  
um tema sobre educação,  
"create na imagem based on this: " + {prop_1}]\n"""\n  
  
prompt_2 = f"""\n  
Não elabora imagens muito complexas,  
  
Usando um estilo visual clássico com uma composição que remeta a  
um tema sobre educação,  
"create na imagem based on this: " + {prop_2}]\n"""\n  
  
prompt_1
```

```
Out[28]: '\n\nNão elabora imagens muito complexas,\n\nUsando um estilo visual sim  
ilar ao de van gogh e com uma composição que remeta a \num tema sobre ed  
ucation, \n"create na imagem based on this: " + create na imagem based on  
this: Inclui problemas do consumo de drogas no currículo escolar.]\n'
```

```
In [30]: pipeline = DiffusionPipeline.from_pretrained("CompVis/stable-diffusion-v1  
                torch_dtype=torch.float16,  
                variant="fp16",  
                use_safetensors=True).to("mp  
for i in range(3):
```

```
for j, prompt in enumerate([prompt_1, prompt_2]):  
    image = pipeline(prompt).images[0]  
    image.save(f"./images/imagem-{j}-{i}.png")
```

```
Loading pipeline components...: 100%|██████████| 7/7 [00:01<00:00,  5.64i  
t/s]  
Token indices sequence length is longer than the specified maximum sequenc  
e length for this model (84 > 77). Running this sequence through the model  
will result in indexing errors  
The following part of your input was truncated because CLIP can only handl  
e sequences up to 77 tokens: ['ículo escolar .']'  
100%|██████████| 50/50 [03:38<00:00,  4.38s/it]  
The following part of your input was truncated because CLIP can only handl  
e sequences up to 77 tokens: ['ouni .']'  
100%|██████████| 50/50 [13:04<00:00, 15.69s/it]  
The following part of your input was truncated because CLIP can only handl  
e sequences up to 77 tokens: ['ículo escolar .']'  
100%|██████████| 50/50 [08:28<00:00, 10.18s/it]  
The following part of your input was truncated because CLIP can only handl  
e sequences up to 77 tokens: ['ouni .']'  
100%|██████████| 50/50 [06:11<00:00,  7.42s/it]  
The following part of your input was truncated because CLIP can only handl  
e sequences up to 77 tokens: ['ículo escolar .']'  
100%|██████████| 50/50 [05:17<00:00,  6.34s/it]  
The following part of your input was truncated because CLIP can only handl  
e sequences up to 77 tokens: ['ouni .']'  
100%|██████████| 50/50 [09:46<00:00, 11.72s/it]
```

Resposta

De forma geral a geração de imagens não foi tão acertiva quanto o esperado, em parte pois o pipeline aceitava somente 77 tokens, o que limitou bastante a qualidade do prompt a ser repassado ao modelo. Ainda, em um dos prompts foi solicitado que o moelo entregasse imagens no estido de Van Gogh, respeitando o tema da proposição, todavia ele somente se ateve a questão do Van Gogh.

Das 6 imagens, uma se mostrou mais adequada pois trouxe uma imagem com pessoas e uma lâmpada, fazendo referência ao tema de educação.

Outro ponto de limitação que impediu a geração de novas imagens foi o tempo necessário para a execução do código. Localmente as 6 imagens foram geradas em aproximadamente 50 minutos. Comparativamente, a opção de geração via colab gratuito estava levando 40 minutos por imagem.

As imagens podem ser vistas a seguir:



