

CPS769 - Introdução à Inteligência Artificial e Aprendizagem Generativa

Este repositório contém as listas de exercícios da disciplina CPS769 - Introdução à Inteligência Artificial e Aprendizagem Generativa, do Programa de Engenharia de Sistemas e Computação (PESC) do Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia (COPPE/UFRJ).

Questão 1

O objetivo deste trabalho é treinar o que foi mostrado em classe nas últimas aulas, isto é, como usar chatbots para chamada a outras ferramentas ou banco de dados.

Especificamente, você trabalhará com as técnicas de chamada de função aprendidas em aula utilizando uma base de dados que fornece informações sobre o tempo, disponibilizada no Kaggle pelo INMET.

Nesta tarefa, você deverá implementar (e rodar) os passos necessários para interpretar e responder perguntas em linguagem natural.

1. Implemente o que for necessário para responder perguntas como:

- (a) "Qual o dia mais quente do ano de 2001"?
- (b) "Qual o dia mais quente do ano de 2021"?
- (c) "Qual a média da temperatura no ano de 2021"?

2. Tente também responder perguntas complexas como:

- (a) "Qual a média do mês de janeiro de 2010"?
- (b) "Janeiro de 2021 fez frio?"?
- (c) "Compare a média de temperatura do ano de 2001 com a de 2021"?

```
In [ ]: from langchain_core.pydantic_v1 import BaseModel, Field
        from langchain_openai import ChatOpenAI
        from langchain_core.prompts import ChatPromptTemplate
        from env import OPENAI_API_KEY
        import pandas as pd
        import pandasql as psq1

        # Inicialização do modelo de linguagem
        llm = ChatOpenAI(model="gpt-4o-mini",
                        openai_api_key=OPENAI_API_KEY,
                        )

        class WeatherQuery(BaseModel):
            """Estrutura para consultas meteorológicas."""
```

```

    csv_file: str = Field(description="O nome do arquivo CSV a ser lido")
    query: str = Field(description="A query SQL para consultar os dados no arquivo")

class NaturalLanguageResponse(BaseModel):
    """Estrutura para a resposta em linguagem natural."""
    resposta: str = Field(description="A resposta em linguagem natural baseada na query")

# Exemplo de sistema para orientar a LLM
system = """You are an expert in meteorological data analysis. You understand the data and can answer queries.

Here are some examples of queries and their structured responses:

example_user: Qual foi o dia com a maior temperatura em 2002?
example_assistant: {"csv_file": "weather_2002.csv", "query": "SELECT DATA, MAX(TEMPERATURA) FROM weather_2002.csv"}

example_user: Qual foi o dia com a temperatura mais alta em 2020?
example_assistant: {"csv_file": "weather_2020.csv", "query": "SELECT DATA, MAX(TEMPERATURA) FROM weather_2020.csv"}

example_user: Qual foi a média das temperaturas no ano de 2020?
example_assistant: {"csv_file": "weather_sum_2020.csv", "query": "SELECT AVG(TEMPERATURA) FROM weather_2020.csv"}

example_user: Qual foi a média da temperatura em janeiro de 2011?
example_assistant: {"csv_file": "weather_2011.csv", "query": "SELECT AVG(TEMPERATURA) FROM weather_2011.csv"}

example_user: O mês de janeiro de 2020 foi frio?
example_assistant: {"csv_file": "weather_2020.csv", "query": "SELECT MIN(TEMPERATURA) FROM weather_2020.csv"}

example_user: Como se comparam as médias de temperatura de 2000 e 2019?
example_assistant: {"csv_file": "weather_sum_all.csv", "query": "SELECT strftime('%Y', DATA), AVG(TEMPERATURA) FROM weather_sum_all.csv"}

prompt = ChatPromptTemplate.from_messages([("system", system), ("human", "{input}")]

few_shot_structured_llm = prompt | llm.with_structured_output(WeatherQuery)

def process_weather_query(pergunta):
    response = few_shot_structured_llm.invoke(pergunta)

    # Determinar quais colunas ler com base no tipo de arquivo
    if 'sum' in response.csv_file:
        colunas = ['DATA (YYYY-MM-DD)', 'temp_avg']
        colunas_novas = ['DATA', 'TEMP']
    else:
        colunas = ['DATA (YYYY-MM-DD)', 'Hora UTC', 'TEMPERATURA DO PONTO DE ORV']
        colunas_novas = ['DATA', 'HORA', 'TEMP']

    # Ler o arquivo CSV limitando às colunas especificadas e renomeá-las
    df = pd.read_csv(f"archive/{response.csv_file}", usecols=colunas)
    df.columns = colunas_novas

    # Executar a query SQL usando pandasql
    result = psql.sqldf(response.query, {'df': df})
    return result, response

# Nova função para gerar a resposta em linguagem natural
def generate_natural_language_response(pergunta, result):
    # Converte o resultado da consulta em um formato de string compreensível
    result_str = result.to_string(index=False)

    # Prompt para gerar a resposta em linguagem natural
    nl_system = """You are a helpful assistant who provides responses in natural

```

Here are some examples:

example_user: Qual o dia mais quente do ano de 2001?

example_assistant: O dia mais quente do ano de 2001 foi [data] com uma temperatu

example_user: Qual a média da temperatura no ano de 2021?

example_assistant: A média da temperatura no ano de 2021 foi de [temperatura méd

example_user: Compare a média de temperatura do ano de 2001 com a de 2021?

example_assistant: A média de temperatura no ano de 2001 foi de [temperatura méd

```
nl_prompt = ChatPromptTemplate.from_messages([("system", nl_system), ("human", nl_prompt)])
nl_structured_llm = nl_prompt | llm.with_structured_output(NaturalLanguageResponse)

response = nl_structured_llm.invoke({"input": f"Pergunta: {pergunta}\nResultado: "})
return response.resposta

# Exemplos de uso
perguntas = [
    "Qual o dia mais quente do ano de 2001?",
    "Qual o dia mais quente do ano de 2021?",
    "Qual a média da temperatura no ano de 2021?",
    "Qual a média do mês de janeiro de 2010?",
    "Janeiro de 2021 fez frio?",
    "Compare a média de temperatura do ano de 2001 com a de 2021?"
]

for pergunta in perguntas:
    print("-----")
    print("Pergunta: ", pergunta)
    result, response = process_weather_query(pergunta)
    print("Query SQL: ", response.query)
    resposta = generate_natural_language_response(pergunta, result)
    print("Resposta: ", resposta)
```

Pergunta: Qual o dia mais quente do ano de 2001?

Query SQL: `SELECT DATA, MAX(TEMP) as Max_Temperature FROM df GROUP BY DATA ORDER BY Max_Temperature DESC LIMIT 1`

Resposta: O dia mais quente do ano de 2001 foi 4 de setembro, com uma temperatura de 40.5 graus Celsius.

Pergunta: Qual o dia mais quente do ano de 2021?

Query SQL: `SELECT DATA, MAX(TEMP) as Max_Temperature FROM df GROUP BY DATA ORDER BY Max_Temperature DESC LIMIT 1`

Resposta: O dia mais quente do ano de 2021 foi 4 de setembro, com uma temperatura de 40.5 graus Celsius.

Pergunta: Qual o dia mais quente do ano de 2021?

Query SQL: `SELECT DATA, MAX(TEMP) as Max_Temperature FROM df GROUP BY DATA ORDER BY Max_Temperature DESC LIMIT 1`

Resposta: O dia mais quente do ano de 2021 foi 15 de setembro, com uma temperatura de 45,0 graus Celsius.

Pergunta: Qual a média da temperatura no ano de 2021?

Query SQL: `SELECT AVG(TEMP) as Avg_Temperature FROM df`

Resposta: A média da temperatura no ano de 2021 foi de 23,2 graus Celsius.

Pergunta: Qual a média do mês de janeiro de 2010?

Query SQL: `SELECT AVG(TEMP) as Avg_Temperature FROM df WHERE strftime('%m', DATA) = '01'`

Resposta: A média da temperatura no mês de janeiro de 2010 foi de 20.28 graus Celsius.

Pergunta: Janeiro de 2021 fez frio?

Query SQL: `SELECT MIN(TEMP) as Min_Temperature FROM df WHERE strftime('%m', DATA) = '01'`

Resposta: Sim, janeiro de 2021 fez frio, com a temperatura mínima registrada de -10,0 graus Celsius.

Pergunta: Compare a média de temperatura do ano de 2001 com a de 2021?

Query SQL: `SELECT strftime('%Y', DATA) as Ano, AVG(TEMP) as Avg_Temperature FROM df WHERE strftime('%Y', DATA) IN ('2001', '2021') GROUP BY Ano`

Resposta: A média de temperatura no ano de 2001 foi de 23.31 graus Celsius, enquanto em 2021 foi de 23.20 graus Celsius.