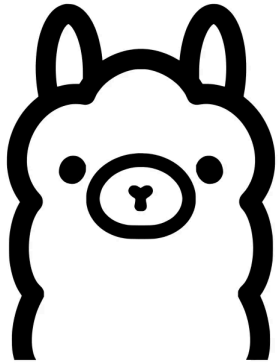

Ollama

本地快速上手大型语言模型

(Llama 3.1, Mistral, Gemma 2, ...)

多模态调用 - 分析图片

操作步骤

- 安装需要的类库
- 编写图片分析代码

操作演示

课堂实验

执行脚本

```
# BakLLaVA 是一个多模态模型，由 Mistral 7B 基本模型和 LLaVA 架构增强组成
ollama pull bakllava

# 安装图片处理类库
pip install Pillow==10.4.0
```

main.py

```python
import time, pprint, warnings, base64, io
from langchain_ollama import ChatOllama
from langchain_core.tools import tool
from langchain_core.messages import HumanMessage
from PIL import Image

warnings.simplefilter("ignore")


def evalEndTime(start_time):
    end_time = time.time()  # 获取结束时间
    execution_time = "(程序运行时间：%.2f 秒)" % (
        end_time - start_time
    )  # 计算程序运行时间
    return execution_time


def convert_to_base64(image_path):
    pil_image = Image.open(image_path)
    buffered = io.BytesIO()
    pil_image.save(buffered, format="JPEG")
    img_str = base64.b64encode(buffered.getvalue()).decode("utf-8")
    return img_str


print("=" * 100)
start_time = time.time()  # 获取开始时间

# 创建Ollama
llm_bakllava = ChatOllama(
    model="bakllava",  # 多模态模型
    temperature=0.5,
)
print(">", llm_bakllava)

llm = ChatOllama(
    # model="llama3.1:8b",
    model="gemma2:9b",
    temperature=0.5,
)
print(">", llm)

# 读取图片
image_data = convert_to_base64("image.jpg")
# 消息数组
messages = [
    HumanMessage(
        content=[
            {
                "type": "text",
                "text": "Please provide a detailed description of the content in this image.",
            },
            {
                "type": "image_url",
                "image_url": {"url": f"{image_data}"},
            },
        ]
    )
]

################################################################################
# 识别图片
result = llm_bakllava.invoke(messages)
print("-" * 100)
print(result.content)

# 翻译中文
result = llm.invoke(
    f"""把下面的内容翻译成中文

{result.content}
"""
)
print("-" * 100)
print(result.content)

print()
```

下课时间

课件下载： https://github.com/komavideo/lesson_ollama