

Simple Image to Text using Langchain, Ollama, LM studio Apis

Mã script này sử dụng các mô hình ngôn ngữ Ollama và OpenAI để thực hiện phân tích hình ảnh và dịch văn bản. Đầu tiên, nó khởi tạo các mô hình với các tham số cụ thể như URL cơ sở và tên mô hình.

Mô hình chính `ChatOllama` được sử dụng để hiểu nội dung của một tệp hình ảnh (`castle.png`) được đọc bất đồng bộ bằng cách sử dụng Node.js's `fs.promises`.

Sau khi phân tích hình ảnh, mô hình `ChatOpenAI` xử lý văn bản được trích xuất. Nó không chỉ thực hiện dịch thuật mà còn xử lý các nhiệm vụ liên quan đến hiểu và tạo ngôn ngữ.

Script xây dựng một thông điệp chứa cả văn bản và dữ liệu hình ảnh. Dữ liệu hình ảnh được mã hóa thành định dạng Base64 để xử lý bởi các mô hình. Sau khi gọi mô hình `ChatOpenAI` với thông điệp đã xây dựng, nó lấy nội dung của hình ảnh dưới dạng văn bản.

Tiếp theo, một mẫu dịch thuật được định nghĩa sử dụng `PromptTemplate` từ `@langchain/core/prompts`. Mẫu này chỉ ra cách dịch văn bản từ một ngôn ngữ sang ngôn ngữ khác, trong trường hợp này từ tiếng Anh sang Tiếng Việt. Nội dung đã dịch được định dạng và sau đó chuyển đến một phiên bản khác của `ChatOllama`, thực hiện việc dịch thuật thực tế.

Thay vì sử dụng URL API từ xa như trước đây, `node module openai` đã thay đổi URL API thành đường dẫn localhost path. Điều này có thể ảnh hưởng đến cách script liên kết với mô hình ngôn ngữ OpenAI.

Cuối cùng, script ghi cả nội dung hình ảnh gốc và bản dịch sang tiếng Việt vào console. Qua quá trình này, script minh họa cách sử dụng các mô hình khác nhau để thực hiện các tác vụ phức tạp như phân tích hình ảnh và dịch văn bản một cách đồng bộ.

```
import { Ollama } from "@langchain/ollama";
import { ChatOllama } from "@langchain/ollama";
import { HumanMessage } from "@langchain/core/messages";
import { PromptTemplate } from "@langchain/core/prompts";
import { ChatPromptTemplate } from "@langchain/core/prompts";
```

```
import fs from "fs/promises";
import { ChatOpenAI, OpenAI } from "@langchain/openai";

const chatModel = new ChatOllama({
  baseUrl: "http://ollama.kyluat.zzz:11434", // Default value
  model: "llama3.2:1b",
});

const run = async () => {
  // from image
  const imageData = await fs.readFile("./castle.png");
  const chatLMModel = new ChatOpenAI({
    baseUrl: "http://192.168.1.111:1234/v1", // Default value
    apiKey: "sk-proj-asdsd--blabah_weUA",
    model: "llava-v1.5-7b",
  });

  const messages = [
    new HumanMessage({
      content: [
        {
          type: "text",
          text: `What's in this image?`,
        },
        {
          type: "image_url",
          image_url: {
            url: "data:image/png;base64," + imageData.toString("base64"),
          },
        },
      ],
    }),
  ];

  const res = await chatLMModel.invoke(messages);
  const imageContent = res.content;

  console.log("Image Content: ", imageContent);

  // translate to vietnamese
```

```
const template =
  "You are a helpful assistant that translates {input_language} to {output_language}.";
const humanTemplate = "{text}";
const chatPrompt = ChatPromptTemplate.fromMessages([
  ["system", template],
  ["human", humanTemplate],
]);
const multiplePromptResult = await chatPrompt.formatMessages({
  input_language: "English",
  output_language: "Vietnamese",
  text: imageContent,
});
const res2 = await chatModel.invoke(multiplePromptResult);
console.log("Vietnamese: ", res2.content);
};

run();
```

Revision #1

Created 15 January 2025 06:48:11 by Son.KyLuat

Updated 15 January 2025 06:53:10 by Son.KyLuat