GenAI with LangchainGo & Ollama 🚀

Running AI Models Locally in Go

Golang Meetup | Helsinki

Maryum Hamid

Introduction

Software Engineer at Zalando

Preowned premise of Zalando

C++ -> Python -> Go

Ollama

Docker for LLMs

Fully local inference

Works offline (perfect for privacy-focused applications)

Written in Go, uses llama.cpp

Ollama demo

```
> ollama pull mistral

> ollama run mistral

>>> What is Ollama?

>>> /bye
```

Supported Models

Llama by Meta

Mistral

Gemma by Google

Phi by Microsoft

Qwen by Alibaba

DeepSeek

LangChainGo

Framework for working with LLMs

Inspired by LangChain (Python)

Abstraction over AI models

Works with multiple LLMs

```
go get github.com/tmc/langchaingo
```

Ollama + LangchainGo

```go
package main
import (
    "context"
    "fmt"
    "github.com/tmc/langchaingo/llms/ollama"
    "log"
)

func main() {
    llm, err := ollama.New(ollama.WithModel("mistral:latest"))
    if err != nil {
        log.Fatal(err)
    }

    resp, err := llm.Call(context.Background(), "Tell me a joke about Golang.")
    if err != nil {
        log.Fatal(err)
    }

    fmt.Println(resp)
}
```

Memory (Remembers Past Messages)

LLMs to remember past interactions

Maintains Conversation Context

Buffer Memory, Conversation Memory, Vector-based Memory

Chatbot Demo With Memory

```
err = chatMemory.AddUserMessage(context.Background(), input)
if err != nil {
    continue
}

messages, err := chatMemory.Messages(context.Background())
if err != nil {
    continue
}

var conversationHistory string
for _, msg := range messages {
    conversationHistory += fmt.Sprintf("%s: %s\n", msg.GetType(), msg.GetContent())
}

response, err := model.Call(context.Background(), conversationHistory, llms.WithTemperature(0.7))
if err != nil {
    continue
}

err = chatMemory.AddAIMessage(context.Background(), response)
if err != nil {
```

Image Recognition

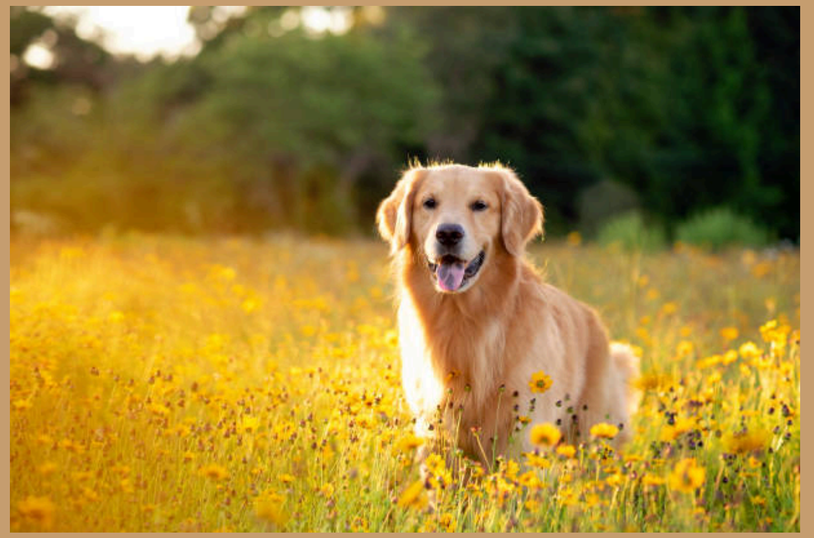
```
llm, err := ollama.New(ollama.WithModel("llava"))
    if err != nil {
        log.Fatal("Error initializing OpenAI:", err)
    }

messages := []llms.MessageContent{
    {
        Role: llms.ChatMessageType("human"),
        Parts: []llms.ContentPart{
            llms.TextPart("Recognize the object in the image"),
            llms.BinaryPart("dog.jpeg", imageData),
        },
    },
}

out, err := llm.GenerateContent(ctx, messages)
if err != nil {
    log.Fatal("Error generating response:", err)
}
```

LLaVA (Large Language and Vision Assistant) is a multimodal model.

Image Recognition Demo



```
AI Response:  The image shows a golden retriever dog standing in a field of yellow flowers,
likely dandelions. The dog is looking towards the camera with its tongue out, and it appears
to be enjoying the sunny day. The background suggests it might be late afternoon or early evening,
given the warm lighting and the shadows cast by the dog and the flowers.
```

Thank you!

https://www.linkedin.com/in/maryum-hamid-6657231aa/