

掌控心理学：使用 Mistral-7B 和 LangChain 构建专家 RAG

原创 大数据杂货铺 大数据杂货铺 2024-04-16 20:12 广东



欢迎来到大语言模型（LLM）时代，它们已成为不可或缺的数字伴侣，帮助我们完成从编码到烹饪冒险的各种任务。然而，我们必须承认，即使是最有能力的人也有局限性。尽管大语言模型拥有令人印象深刻的能力，但他们仍面临着两个重大挑战：

信息访问：大语言模型无法快速访问其训练集之外的数据。想象一下，你最喜欢的人工智能助手无法为你提供有帮助的答案，但却喋喋不休地告诉你如何获得问题的答案。

幻觉错误：这可能听起来令人惊讶，但大语言模型有时会产生看起来像是进入了想象领域的反应。这些不准确之处可能会导致混乱和沮丧。

我们都经历过大语言模型似乎陷入困境的时刻，由于其有限的训练数据而无法产生一致的反应。这就像有一个朋友擅长提供一般建议，但在寻求真正的帮助时却陷入困境。

例如，当你向顶级大语言模型提出这些深刻的问题时，像 ChatGPT-3.5 这样的人工智能往往会给出听起来像是来自自助小册子的答案。他们擅长反省数据模式，但缺乏提供个性化建议所需的真正的人类理解。因此，如果一个人正在寻求真正的支持，普通的大语言模型几乎没有什么帮助。



You

I constantly have the urge of getting into violence, am I normal and how do I get rid of this?



ChatGPT

Feeling the urge to engage in violence can be concerning, but it's essential to recognize that you're not alone, and there are ways to address these feelings. It's not uncommon for people to experience anger or aggression, but acting on these impulses in harmful ways can have serious consequences.

First and foremost, it's crucial to seek professional help from a therapist or counselor who can provide support and guidance in managing these urges. Therapy can help you explore the underlying causes of your violent impulses and develop healthy coping mechanisms to manage them.

In addition to therapy, consider practicing relaxation techniques such as deep breathing, meditation, or physical exercise to help manage feelings of anger and stress. Finding healthy outlets for your emotions, such as engaging in creative activities or participating in sports, can also be beneficial. 公众号 · 大数据杂货铺

<https://chat.openai.com/>

在人工智能的旋风世界中，每当这些充满了所有奇特的扩展训练数据的新模型下降时，我们几乎都会受到摆布。但是，嘿，如果我们想要一位像真正的心理学家一样提供建议的助理呢？我们不要忘记深入探讨这些深刻的问题。另外，如果我们值得信赖的大语言模型有更多的情报来精确地确定这些回答，那不是很好吗？

现在，微调似乎是一种可能的解决方案，但它并不全是彩虹和独角兽，它有一系列风险和复杂性：

模型漂移：当我们不断用新数据微调模型时，模型可能会偏离原始性能，从而可能产生意想不到的结果。

成本和复杂性：除了技术障碍之外，迭代微调过程还需要大量的计算资源和专业知识，这意味着巨额投资。

这就是 RAG 发挥作用的地方。在本系列中，我们将开始探索 RAG，涵盖：

了解 RAG：检索增强生成模型到底是什么？它们如何运作？

使用 Mistral 7b 实施：通过 HuggingFace 和 LangChain 等平台利用 Mistral 7b 构建 RAG 的分步指南。

现实世界的应用：通过实际例子见证小型 RAG 的实际应用，包括创建一个能够像人类心理学家一样深入研究深层问题的心理学助理。

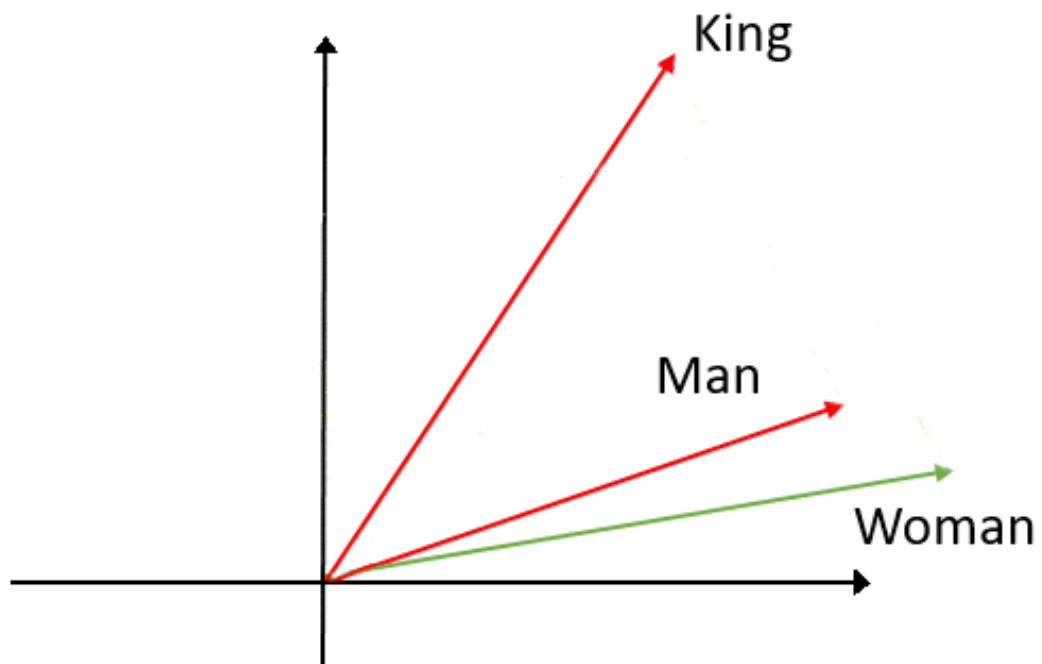
在深入研究错综复杂的技术之前，让我们确保我们对一些关键概念达成共识：

1. 文本嵌入

文本嵌入就像计算机大脑中的文字魔法。它们将单词、短语甚至整个句子转化为数字，帮助计算机理解它们的含义。可以将其想象为将文字转换为只有计算机可以读取的密码。我们有 Word2Vec 和 GloVe 等老式方法，也有 BERT 等新颖的方法，它们都发挥着神奇的作用来理解文本。

想象一下：你有"国王"+"男人"+"女人"。这就像试图在世界上寻找"女王"一样。令人惊讶的是，这个数学技巧常常让我们准确无误，显示出文本嵌入在理解单词关系方面有多么出色。

文本嵌入已经取得了长足的进步。我们从基本方法开始，但现在我们有了像 BERT 这样的转换器来改变一切。他们就像街区里的酷孩子一样，利用注意力来理解上下文中的单词。此次升级意味着他们可以处理更大的文本块并更好地理解它们。



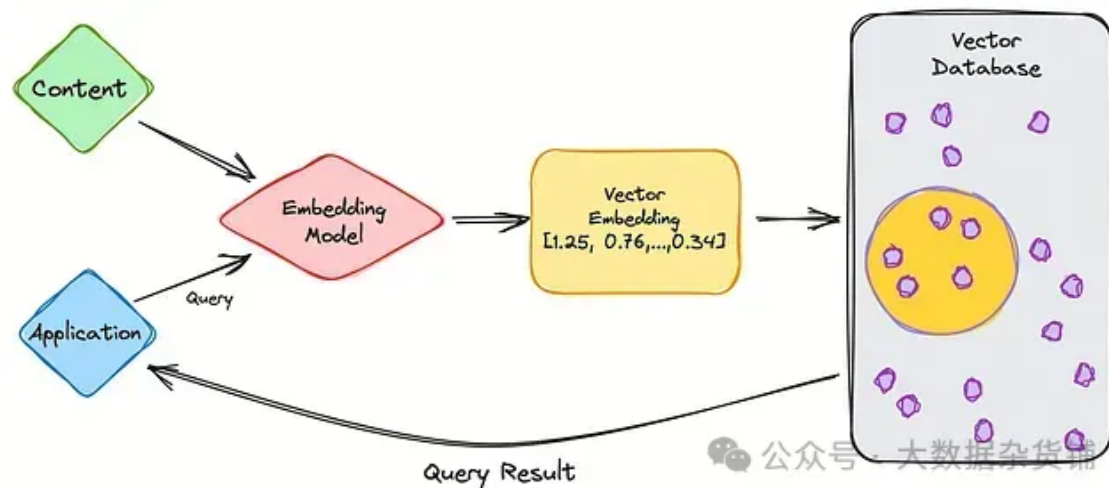
"国王"、"女王"、"男人"和"女人"之间语义关系的二维图形描述。

以下是来自 OpenAI 和 Cohere 的一些重要资源，可以帮助您更好地理解文本嵌入。

2. 矢量数据库

那么，您知道处理大量复杂数据是多么令人头痛吗？传统数据库有时无法解决如此复杂的问题。但随后，矢量数据库出现了，可以拯救世界！他们就像数据世界的超级英雄，轻松处理所有棘手的高维数据点。

想象一下：您的数据点全部布置在网格上，每个数据点都由其独特的属性表示。这有点像根据口味而不是颜色或形状来组织一堆不同的水果。唯一的区别是文本的这些属性是使用文本嵌入（向量）来表示的。



图片来源: KDnuggets

矢量数据库使用余弦相似度等很酷的技巧来计算不同数据点的相似程度，就像 Google 根据它们与您的查询的匹配程度向您显示搜索结果一样。这一切都是为了理解混乱的数据并帮助您快速找到所需的内容！

对于好奇的人来说，以下是一些了解矢量数据库更多信息的好资源

3. RAG 揭幕

但 RAG 到底是什么？检索增强生成模型充当桥梁，为语言模型（LLM）提供外部数据访问权限，以生成富含上下文洞察的响应。无论是借鉴最近的新闻、讲座记录，还是在我们的案例中，借鉴顶级心理学文献，RAG 都使大语言模型能够以新发现的深度和相关性做出回应。

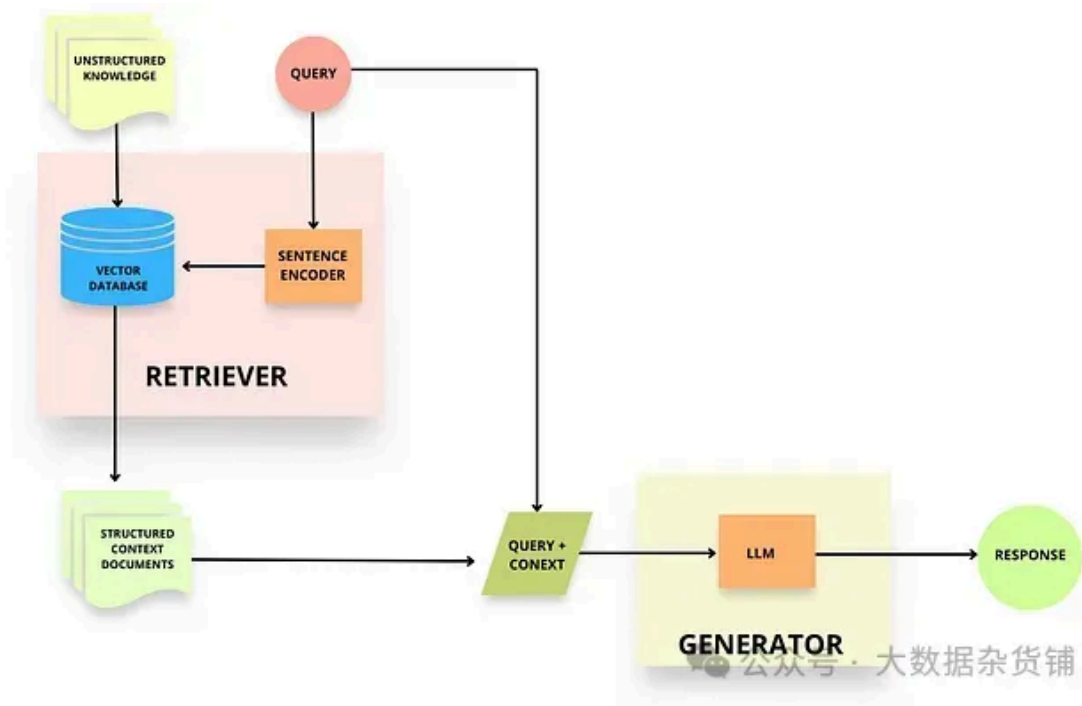
您问它们如何发挥作用？将 RAG 想象为配备向量搜索机制的大语言模型。以下是该过程的简化分解：

数据库设置：使用编码文档填充矢量数据库。

查询编码：使用句子转换器将输入查询转换为向量。

相关上下文检索：根据查询从向量数据库中检索相关上下文。

提示大语言模型：利用检索到的上下文和原始查询，指导大语言模型生成充满上下文深度的响应。



高级 RAG 架构

4. 心理专家 RAG

在打开代码编辑器之前，让我们花点时间了解一下为什么 RAG 在心理学领域具有巨大的潜力。想象一下，拥有一个数字伴侣不仅能够理解您的查询，还能从庞大的心理学文献库中获取相关见解。借助 RAG，我们将进入一个只需敲击几下键盘即可获得个性化心理健康支持和同理心反应的世界。现在让我们进入代码部分！

4.1 入门：安装和设置库

好吧，让我们解决这些依赖关系。我们需要大量库来启动我们的 RAG 之旅。别担心，我们已经为您提供了一个简单的 pip install 命令：

```
!pip install -q torch datasets
!pip install -q accelerate==0.21.0 \
    peft==0.4.0 \
    bitsandbytes==0.40.2 \
    trl==0.4.7
!pip install -q langchain \
    sentence_transformers \
    faiss-cpu \
    pypdf
!pip install -U transformers
```

4.2 导入行业工具

现在我们的库已经就位，让我们引入重型火炮。我们将使用 PyTorch、Transformers 和一点 LangChain 魔法来编织我们的 RAG 杰作。


```

import os
import torch
from transformers import (
    AutoTokenizer,
    AutoModelForCausalLM,
    BitsAndBytesConfig,
    pipeline
)

from transformers import BitsAndBytesConfig

from langchain.text_splitter import CharacterTextSplitter

from langchain.embeddings.huggingface import HuggingFaceEmbeddings
from langchain.vectorstores import FAISS

from langchain.prompts import PromptTemplate
from langchain.schema.runnable import RunnablePassthrough
from langchain.llms import HuggingFacePipeline
from langchain.chains import LLMChain
import transformers
import nest_asyncio

nest_asyncio.apply()
from tqdm.notebook import tqdm
from langchain_community.document_loaders import PyPDFLoader

```

4.3 加载模型和分词器

啊，我们 RAG 应用程序的核心和灵魂 - Mistral-7B。让我们启动模型和分词器，但是等等，让我们高效地将其全部安装在单个 GPU 上。我们在这里所做的一切都是为了优化！

```

# Load Tokenizer
model_name='mistralai/Mistral-7B-Instruct-v0.2'

tokenizer = AutoTokenizer.from_pretrained(model_name)
tokenizer.pad_token = tokenizer.eos_token
tokenizer.padding_side = "right"
print("Tokenizer loaded !!")

## bitsandbytes parameters
# Activate 4-bit precision base model loading
use_4bit = True

# Compute dtype for 4-bit base models
bnb_4bit_compute_dtype = "float16"

# Quantization type (fp4 or nf4)

```

```

bnb_4bit_quant_type = "nf4"

# Activate nested quantization for 4-bit base models (double quantization)
use_nested_quant = False

compute_dtype = getattr(torch, bnb_4bit_compute_dtype)

bnb_config = BitsAndBytesConfig(
    load_in_4bit=use_4bit,
    bnb_4bit_quant_type=bnb_4bit_quant_type,
    bnb_4bit_compute_dtype=compute_dtype,
    bnb_4bit_use_double_quant=use_nested_quant,
)

# Check GPU compatibility with bfloat16
if compute_dtype == torch.float16 and use_4bit:
    major, _ = torch.cuda.get_device_capability()
    if major >= 8:
        print("=" * 80)
        print("Your GPU supports bfloat16: accelerate training with bf16=True")
        print("=" * 80)

# Load Model
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,
)

```

让我们在一个基本问题上测试我们的普通模型

```

def get_vanilla_response(question, tokenizer, model):
    inputs = tokenizer.encode_plus(f"[INST] {question} [/INST]", return_tensors="pt")['input_ids']

    generated_ids = model.generate(inputs,
                                   max_new_tokens=1000,
                                   do_sample=True)

    output = tokenizer.batch_decode(generated_ids)
    return output[0]

question = "What changes do I need in my personality to make friends easily?"
print(get_vanilla_response(question = question, tokenizer = tokenizer, model = model))

```

以下是我们得到的响应，运行时响应可能会略有变化

Question: What changes do I need in my personality to make friends easily?

Response:

Making friends can be a challenge for some people, and it often requires some adjustments to our personality traits and social behaviors. Here are some suggestions for changes that may help you make friends more easily:

1. Be open and approachable: People are more likely to be drawn to those who are friendly and welcoming. Smile often, make eye contact, and show genuine interest in others.
2. Be a good listener: Most people love to talk about themselves and share their stories. Be an attentive listener and show empathy when others speak. This will make them feel valued and appreciated.
3. Be patient and persistent: Making friends takes time and effort. Don't get discouraged if it takes a while to build a strong connection. Keep putting yourself out there and be persistent.
4. Be optimistic: Having a positive attitude and a good sense of humor can help attract people to you. People are more likely to want to be around those who make them feel good.
5. Be reliable and consistent: People value those who are dependable. Keep your commitments and be consistent in your actions and behaviors.
6. Be authentic to yourself: Don't try to be someone you're not just to fit in. People are drawn to authenticity and genuine personality. Be yourself and let your unique qualities shine through.
7. Be willing to try new things: Being open to new experiences and trying new things can help broaden your horizons and expand your social circle. This can lead to meeting new people and making new friends. Keep in mind that making friends is a two-way street, so be open to forming genuine connections and being a good friend in return.

这个答案很好，但仍然很笼统，缺乏人情味。现在让我们尝试使用 RAG 来改善这一点。

4.4 打造 RAG 管道

有趣的来了。将它们放在一起创建我们的 RAG 管道。我们将使用 LangChain 来无缝地协调检索和生成过程。

5. 创建文本生成管道

```
# Create a text generation pipeline
text_generation_pipeline = pipeline(
    model=model,
    tokenizer=tokenizer,
```



```
task="text-generation",
temperature=0.2,
repetition_penalty=1.1,
return_full_text=True,
max_new_tokens=10000,
)

mistral_llm = HuggingFacePipeline(pipeline=text_generation_pipeline)
```

现在让我们来看看我们将用来创建心理助手的数据。

5.1 心理学书籍清单

- 丹尼尔·卡尼曼 (Daniel Kahneman) 的《思考，快与慢》：这本书深入研究了两种思维系统，提供了对认知偏见和决策过程的见解。
- 罗伯特·B·西奥迪尼 (Robert B. Cialdini) 的《影响力：说服心理学》：西奥迪尼探索了影响力的原理，揭示了影响人类行为的心理触发因素。
- 丹·艾瑞利 (Dan Ariely) 的《可预测的非理性》：艾瑞利的书探讨了影响决策的非理性行为，为人类心理学提供了有价值的观点。
- 丹尼尔·戈尔曼 (Daniel Goleman) 的《情商》：这本书强调了情商在理解自己和他人方面的重要性，这是人类心理学的一个重要方面。
- 艾略特·阿伦森 (Elliot Aronson) 的《社会动物》：阿伦森的书涵盖了社会心理学的广泛主题，提供了对社会背景下人类行为的见解。
- 理查德·H·塞勒 (Richard H. Thaler) 和卡斯·R·桑斯坦 (Cass R. Sunstein) 的《助推》：塞勒和桑斯坦结合心理学和经济学的见解，讨论了微妙的助推如何影响决策。

```

# add Book paths from Google Drive
pdf_paths = ['/content/drive/MyDrive/Blogs/psychology-gpt/Dan Ariely - Predictably Irrati
              '/content/drive/MyDrive/Blogs/psychology-gpt/Daniel Goleman - Emotional Inte
              '/content/drive/MyDrive/Blogs/psychology-gpt/Daniel Kahneman - Thinking, Fas
              '/content/drive/MyDrive/Blogs/psychology-gpt/Elliot Aronson - The Social Ani
              '/content/drive/MyDrive/Blogs/psychology-gpt/Richard H. Thaler, Prof. Cass R
              '/content/drive/MyDrive/Blogs/psychology-gpt/Robert B. Cialdini - Influence_

docs = []
for pdf_path in tqdm(pdf_paths):
    loader = PyPDFLoader(pdf_path)
    pages = loader.load_and_split()
    docs = docs+pages[8:-10]

# Load chunked documents into the FAISS index
db = FAISS.from_documents(docs,HuggingFaceEmbeddings(model_name='BAAI/bge-base-en-v1.5'))
retriever = db.as_retriever()

```

```

# Create prompt template
prompt_template = """
### [INST] Instruction: Answer the question based on your human psychology knowledge, you

{context}

### QUESTION:
{question} [/INST]
"""

# Create prompt from prompt template
prompt = PromptTemplate(
    input_variables=["context", "question"],
    template=prompt_template,
)

# Create llm chain
llm_chain = LLMChain(llm=mistral_llm, prompt=prompt)

rag_chain = (
    {"context": retriever, "question": RunnablePassthrough()}
    | llm_chain
)

```

5.2 让我们测试一下我们的 RAG

设置就足够了；让我们测试一下我们的 RAG！我们将提出一些发人深省的问题，看看效果如何。

```
# testing on the same question for a comparison
question = "What changes do I need in my personality to make friends easily?"
op = rag_chain.invoke(question)
print(op['text'])
```

Question: What changes do I need in my personality to make friends easily?

Response:

Based on the information provided in the documents, here are some suggestions for developing social skills to make friends more easily:

1. Be attuned to others' needs and feelings: Strive for a balance between being true to yourself and being sensitive to others' needs. This will help you avoid becoming a social chameleon, who may win popularity but sacrifices true satisfaction.
2. Honesty and authenticity: Be genuine in your interactions with others. People appreciate authenticity and are more likely to form lasting relationships with those who are true to themselves.
3. Active listening: Pay attention to what others are saying and respond appropriately. This shows respect and interest in their thoughts and experiences.
4. Empathy: Try to understand others' perspectives and emotions. This will help you connect with them on a deeper level and build stronger relationships.
5. Adaptability: Be flexible in your interactions with others. This means being able to adjust your behavior to fit different social situations and adapt to the needs and preferences of the people around you.
6. Positive body language: Use open and positive body language to convey warmth and approachability. This includes maintaining eye contact, smiling, and using appropriate facial expressions.
7. Approachability: Make yourself available and approachable to others. This means being open to new experiences and meeting new people, and creating opportunities for social interaction.
8. Practice active engagement: Engage fully in conversations and activities with others. This means being present in the moment and actively participating in the interaction, rather than just going through the motions.

By focusing on these areas, you can develop the social skills necessary to make friends more easily and build strong, meaningful relationships.

6. 好处

因此，当我们比较这两个答案时，我们注意到他们在如何提出提高交友社交技能的建议方面存在一些相似之处和不同之处。

现在，基于 RAG 的答案是什么？这就像一个朋友为你提供了整个分解-----详细，并为每项技能提供了具体的例子。这都是关于实用的技巧和策略，为您提供清晰的步骤来提升您的社交游戏水平。另一方面，普通的答案更广泛一些，发生了哲学上的转变。

从本质上讲，RAG 通过利用外部数据源增强基本语言模型的功能来提高响应的质量和深度，这使其对于需要特定领域专业知识任务（例如提供社交技能和心理学建议）特别有价值。

对于那些想要进行更多实验的人来说，这里是入门笔记本。如果您有任何疑问，请通过我的 LinkedIn 账号联系我，我很乐意提供帮助！

7. 参考

Josh Starmer 对词嵌入的深入介绍

Huggingface 上的顶级嵌入模型

LangChain RAG Cookbook

原文链接：<https://ai.plainenglish.io/mastering-psychology-building-an-expert-rag-with-mistral-7b-and-langchain-39bd5a5bdb4c>

RAG 12 LLM 15 langchain 2

RAG · 目录

上一篇

提高大型语言模型（LLM）性能的四大数据清理技术

下一篇

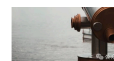
深入研究向量数据库

[阅读原文](#)

喜欢此内容的人还喜欢

OpenSearch 与 Elasticsearch：7 个主要差异及如何选择





【工具箱】Cryptal探索密码学工具箱的无限可能
Coder小Q



基于SpringBoot的班级回忆录系统
小猿编程秘籍

