人工智能软件开发与实践大作业技术报告

lkyu

日期：2024 年 9 月 8 日

摘　　要

在当前人工智能技术的快速发展背景下，聊天机器人作为一种新兴的交互方式，正逐渐渗透到人们的日常生活中。本项目旨在开发一个集成文档问答和角色扮演功能的人工智能聊天机器人。项目的核心目标是提供一个用户友好的界面，通过自然语言处理技术实现高效的信息检索和交互体验。

为了实现这一目标，项目采用了 gradio 库来构建可视化前端界面，使得用户能够直观地与聊天机器人进行互动。在后端，项目集成了百度千帆的“ernie-speed-128k”大型语言模型（LLM）和 huggingface 上的“BAAI/bge-small-zh”开源嵌入模型，以支持文档阅读和问答功能。通过调用这些模型，聊天机器人能够理解和处理用户上传的长文本文件，提供准确的信息检索和问答服务。此外，项目还实现了一个角色扮演聊天机器人，它能够记住对话历史并根据预设的角色人设进行互动。

关键词：大模型；文本生成；文档阅读；角色扮演聊天

1　项目概述

在人工智能技术日益成熟的今天，聊天机器人作为人机交互的重要方式之一，其应用范围和功能复杂性都在不断扩展[1]。本项目“人工智能软件开发与实践大作业展示”旨在开发一个多功能的人工智能聊天机器人，该机器人不仅能够进行文档问答，还能进行角色扮演，以提供更丰富的用户体验。项目的通过结合先进的自然语言处理技术和机器学习模型，实现了一个能够理解和响应用户需求的智能系统。

项目的创建并实现了两个主要功能：文档问答聊天机器人和角色扮演聊天机器人。文档问答聊天机器人利用百度千帆的“ernie-speed-128k”大型语言模型和 huggingface 上的“BAAI/bge-small-zh”嵌入模型，能够处理和理解用户上传的文档内容，提供准确的信息检索和问答服务。这一功能特别适合需要处理大量文本数据的场景，如法律咨询、学术研究等。

角色扮演聊天机器人则通过维护对话历史记录，结合预设的角色人设，使得机器人能够在对话中模拟并保持特定角色的行为和语言风格。

为了使这些功能直观地体现在屏幕上，项目采用了 gradio 库来构建用户友好的可视化界面，用户可以通过简单的图形界面与机器人进行互动。

2　技术架构

2.1　可视化前端界面

在本项目的实现中，技术架构是构建高效、用户友好的人工智能聊天机器人的关键。项目采用了 Gradio 库来创建可视化前端界面，该界面是用户与机器人交互的直接媒介，允许用户通过直观的方式提交问题并接收答案。Gradio 是一个开源的 Python 库，专门用于创建机器学习模型的交互式界面，它支持快速原型设计、模型验证、演示和教学，非常适合于本项目的需要。

在项目的前端设计中，我们定义了两个主要的交互界面，分别用于角色扮演聊天和长文档阅读。角色扮演聊天界面允许用户与机器人进行基于预设角色的对话，而长文档阅读界面则允许用户上传文档并提出相关问题。这两个界面的设计充分利用了 Gradio 的功能，通过 gr.Blocks() 和 gr.TabbedInterface 来组织和展示不同的交互组件。

在角色扮演聊天界面中，我们使用了 gr.Chatbot() 组件来展示对话历史，并通过 gr.Textbox() 和 gr.Button() 组件来接收用户的输入和触发对话更新。

```
with gr.Blocks() as app1:
  chatbot = gr.Chatbot()
  msg = gr.Textbox(label="输入")
  submit_btn = gr.Button("Submit")
  submit_btn.click(fn=update_chat, inputs=[chatbot, msg], outputs=[chatbot, msg])
  msg.submit(fn=update_chat, inputs=[chatbot, msg], outputs=[chatbot, msg])
  clear = gr.ClearButton([msg, chatbot])
```

在长文档阅读界面中，我们通过 gr.Interface() 组件来实现文件上传和文档内容的读取。用户可以上传文档，系统会调用后端的处理函数来读取和分析文档内容，然后返回相应的答案。

```
with gr.Blocks() as app2:
  upload_interface = gr.Interface(fn=save_file, inputs="file", outputs="text")
  read_input = gr.Textbox(label="向文档提问")
  read_output = gr.Textbox(label="结果")
  read_interface = gr.Interface(fn=read, inputs=read_input, outputs=read_output)
```

2.2 功能实现

功能实现是构建人工智能聊天机器人的核心部分，涉及到文档问答和角色扮演两大功能。我们在调用千帆模型 API 完成文本生成任务的同时，辅以本地部署的开源文本向量模型，以更好的完成需求。

2.2.1 文档问答聊天机器人

文档问答聊天机器人的实现基于百度千帆的“ernie-speed-128k”大型语言模型（LLM）和 huggingface 上的“BAAI/bge-small-zh”模型。这些模型提供了强大的文本理解和生成能力，使得机器人能够处理复杂的文档问答任务。通过使用 langchain 库，我们能够将文本转换为语义向量，进而实现高效的信息检索和问答。在后端，我们调用了 read 函数，该函数负责处理用户上传的文档，并返回相应的答案。例如，当用户上传一个文档并提出问题时，read 函数会被触发，它首先通过 TextLoader 加载文档内容，然后使用 RecursiveCharacterTextSplitter 进行文本分割，最后通过 Chroma 向量数据库和 create_retrieval_chain 创建的检索链来找到答案。

```
# 导入语料
  loader = TextLoader("./temp.txt")
  text = loader.load()

  # 导入文本
  documents = text_splitter.split_documents(text)

  # 存入向量数据库
  vector = Chroma.from_documents(documents, embeddings)
  retriever = vector.as_retriever()
  retrieval_chain = create_retrieval_chain(retriever, document_chain)
```

```
  def read(input_text):
    if loader is None:
      return "请上传纯文本语料！"
    else:
      return retrieval_chain.invoke({"input": input_text})["answer"]
```

2.3 角色扮演聊天机器人

角色扮演聊天机器人的实现则侧重于对话管理和上下文理解。通过维护一个对话历史记录列表 conversation_history，机器人能够记住之前的对话内容，从而在角色扮演中提供连贯和合理的回答。我们使用了百度千帆的“ernie-speed-128k”模型来生成回答，该模型能够根据当前的对话历史和预设的角色人设来生成合适的回复。我们定义了 send_request 函数，它负责发送用户输入到百度的 API，并接收模型生成的回答。此外，我们还在 main.py 中定义了 update_chat 函数，用于更新聊天记录并触发对话生成。

3 功能实现细节

3.1 prompt 优化

在实现文档问答聊天机器人时，我们发现 prompt 的设置对于模型的效果至关重要。prompt 可以帮助模型更好地理解用户的需求，从而更准确地生成答案。prompt 的设置可以分为两种类型：固定 prompt 和可变 prompt。固定 prompt 是指模型在生成答案时，始终使用固定的模板, 我们的项目中使用的就是固定 prompt。

在文档问答聊天机器人中，我们使用了如下的固定 prompt 模板，规范了 LLM 的生成效果：

```
# 创建提示词模板
prompt = ChatPromptTemplate.from_template(
  """
  使用下面的语料来回答本模板最末尾的问题。如果你不知道问题的答案，直接回答 "我不知道"，禁止随
      意编造答案。
  为了保证答案尽可能简洁，你的回答必须不超过三句话，你的回答中不可以带有星号。
  以下是一对问题和答案的样例：
    请问：秦始皇的原名是什么
    秦始皇原名嬴政。

  以下是语料：
  <context>
  {context}
  </context>

  Question: {input}
  """
)
```

角色扮演聊天机器人的 prompt 则负责初始化人设：

```
  payload = json.dumps(
        {
            "messages": conversation_history,  # 发送对话历史记录
            "stream": False,
            "temperature": 0.9,
```

```
            "top_p": 0.7,
            "penalty_score": 1,
            "system": "现在我们开始一个角色扮演游戏，以下是你的人设：你是尼克·王尔德（Nick␣
                Wilde），男，是2016年迪士尼动画电影《疯狂动物城》中的男主角。原型是赤狐。原名
                尼古拉斯·皮比里厄斯·王尔德（Nicholas␣Piberius␣Wilde），在中国大陆地区又被称作
                狐尼克，由杰森·贝特曼和凯特·索西配音。你少时因遭遇挫折和他人的偏见，被迫放弃了
                自己的理想，打算不再为谁付出，长大后以坑蒙拐骗为生。你口若悬河、思维敏捷、谎技
                高超但却内心善良，同时拥有过目不忘的惊人记忆力。因为意外而和动物城警官朱迪·霍
                普斯被卷进了一个意欲颠覆动物城的巨大阴谋，在案件的侦破过程中，你的超常记忆力和
                对动物城了如指掌起到了至关重要的作用。案件成功告破后，你通过训练并加入了动物城
                警察局，成为了一名真正的警察，实现了自己的梦想，正式成为了朱迪的搭档。",
            "max_output_tokens": 4096,
            "frequency_penalty": 0.1,
            "presence_penalty": 0.0,
        }
    )
```

3.2 模型选择

在本项目中，我们使用了百度千帆的“ernie-speed-128k”模型和 huggingface 上的“BAAI/bge-small-zh”模型。

“ernie-speed-128k”模型是百度自研的大型语言模型，在文本生成任务上有着优良的表现。

“BAAI/bge-small-zh”模型是结构类似于 BERT 的开源文本向量模型[2]，它能够生成高质量的文本向量，并支持多种文本表示方式。模型规模为“small”，在生成文本时速度较快，仅使用 cpu 处理万字文本也可以达到较短时间内完成任务的效果。

4 使用示例

图 1: 角色扮演聊天机器人界面，可正常对话，并能记住上下文

图 2: 文档阅读机器人界面，可上传文件并实时处理，提问得到的回答较为精准

参考文献

[1] JAFAROV K B Z J Z. Purpose and advantages of chatbots. Proceedings of Azerbaijan High Technical Educational Institution, 2024, 42(07): 114-122. DOI: 10.36962/pahtei42072024-13.

[2] XIAO S, LIU Z, ZHANG P, C-Pack: Packaged Resources To Advance General Chinese Embedding. 2024. arXiv: 2309.07597 [cs.CL]. https://arxiv.org/abs/2309.07597.