## ## 引言

伴随着深度学习技术在自然语言处理方面取得的重大突破，对话式人工智能（Chatbot）已经不再是科幻小说中的遥不可及的愿景，而是已然转变为了现实世界中的一个重要组成部分，逐步融入到我们每个人生活、工作以及人机交互过程之中。尤其是在数字化转型的大趋势下，这些智能化的机器人助手已经不仅仅局限于家庭、办公室等场所，更是在各个行业领域中发挥着举足轻重的作用，他们就像专业的咨询师，致力于解决特定领域内的各种问题，为广大用户提供更为便捷、高效的服务体验。

然而，随着社会发展和用户需求的不断变化，仅限于单一领域的聊天机器人已经无法满足人们日益增长的多元化需求。在此背景之下，一种能够跨越不同行业领域、整合多种功能特性的综合性聊天机器人助手显得尤为必要且迫切。

本篇文章旨在设计并实现一款基于langchain框架的综合聊天机器人，利用该框架将大规模语言模型、智能体以及Prompt工程有机地结合起来，赋予其问答、文本分析、网络搜索以及数据库查询等多重功能，同时借助streamlit可视化框架与用户进行高效、直观的交互。经过实际测试验证，采用langchain框架构建的聊天机器人具备更为丰富多样的功能特性，能够依据用户提出的问题进行深入分析和精准选择，从而提供更为真实可信的解答。

关键词：langchain、大规模语言模型、聊天机器人、Prompt工程、智能体

#### #### 背景概述

时间步入二十一世纪第三个十年之际，以先进的“预训练＋微调”自然语言处理（NLP）模式崭露头角的生成式人工智能ChatGPT成为了全球科技界瞩目的焦点。得益于搭载大规模语言模型（LLM）的ChatGPT，它通过海量的模型参数、卓越的计算性能，以及与人类反馈相融合的强化学习算法，使得其语言生成能力始终保持在顶尖水平。这一强大实力使得ChatGPT在各种复杂场景下均能精准解读语义，并生成高品质的文本内容，从而保证了与用户之间的持续且稳定的对话过程。

作为ChatGPT的升级版本，GPT-4在人机交互的广度和深度上都取得了显著突破，成功实现了多模态交互，即支持文本和图像两种形式的输入。这一创新不仅使机器有能力阅读文本，而且可以深入理解图像所蕴含的意义，为用户带来更加丰富多彩的交互体验。

纵观媒介发展历程，我们不难发现，人类与媒介的交互方式正在逐渐由文字向视觉转变，从简单的交互行为演变为沉浸式的体验，从现实世界延伸至虚拟空间。因此，伴随着通用人工智能技术的日新月异，视觉化的多元化互动、沉浸式的深度体验以及虚实环境的融合必将成为人机交流的重要组成部分，进而推动人机交流能力的大幅度提升。

如今，聊天机器人已悄然融入各行各业的日常运营之中。无论是在银行、电子商务，还是医疗保健、教育领域，聊天机器人凭借其高效便捷的特点，为用户带来了前所未有的服务体验。然而，尽管聊天机器人的应用领域广泛，但综合性聊天机器人仍然较为罕见。

根据相关统计数据显示，全球范围内有超过数百万家企业已经采纳了聊天机器人技术，这些机器人主要应用于客户服务、销售推广、市场研究等多个领域。以银行业为例，众多银行纷纷推出了自家的聊天机器人，用以提供账户查询、转账汇款、信用卡申请等基本服务。而在电子商务领域，聊天机器人则更多地被用于商品咨询、订单跟踪、售后服务等环节。

然而，尽管这些聊天机器人在一定程度上满足了用户的需求，但它们的功能往往仅限于特定领域，难以实现跨行业的综合运用。相比之下，综合性聊天机器人具有更高的智能化水平和适应性，能够同时处理来自不同行业、不同领域的信息，为用户提供更为全面、个性化的服务。

#### #### 意义

近年来，尽管智能化聊天机器人领域已经取得了瞩目的进展，但是实施此类综合性的聊天机器人项目仍然面临诸多严峻的挑战。其中，最为突出的问题在于，这种综合型的机器人开发流程过于繁琐且充满不确定性，这也使得聊天机器人的数量依然比较稀缺。特别是对于具备大规模语言模式结构的机器人来说，尽管其功能强大，但在应对复杂多变的聊天情境时，却无法做到全面兼顾。此外，这类机器人还普遍存在无法直接访问互联网上或者数据库中的真实信息的局限性，这无疑限制了它们在实际应用环境下的表现。

在开发综合性聊天机器人的过程中，一个至关重要的难点就是如何高效地整合多个智能代理（Agent）。每一个智能代理都有可能负责处理特定的任务或领域，例如问答、推荐、情感分析等等。然而，要想将这些智能代理无缝地整合到一个统一的聊天机器人系统之中，并且保证它们能够协同工作、共享信息和数据，这无疑是一项极具挑战性的任务。

Langchain框架作为大语言模型应用开发领域的一颗璀璨新星，为解决上述问题提供了全新的思路。该框架通过精心设计的工具和组件集，大大简化了开发人员构建基于大语言模型的智能系统的过程。本篇论文正是利用Langchain框架的强大功能，设计并实现了一款集成性极高的聊天机器人，旨在深入挖掘大语言模型在聊天机器人技术中的巨大应用潜力，进而进一步提高聊天机器人的性能水平，并拓展其应用范围。

### ### 国内外研究现状

#### #### 大模型应用现状

国外Streamlit企业携手业界知名企业Snowflake于2023年共同公布了《大语言模型应用程序2023年度概述》（简称《State of LLM Apps 2023》）的最新研究成果。

本报告针对Streamlit社区云平台上托管的共计29,183款应用程序进行深入分析，其中包含了由17,926位开发者独立设计并创建的优秀作品。这一庞大规模的样本群体，无疑为我们揭示大语言模型（LLM）应用程序开发领域的新兴趋势、创新工具以及实际应用案例提供了宝贵的参考依据。尽管国内大模型应用程序在Streamlit社区云平台上的数量相对较少，但对于我们洞察海外大模型应用发展动态仍具有重要意义。以下是本报告的核心要点：

首先，OpenAI在大语言模型应用程序市场中占据着绝对的领导地位，高达73%的受访者表示选择使用其GPT模型。得益于OpenAI在GPT研究方面的卓越贡献、出色的输出质量、灵活易用的API接口等优势，使得其成为了大语言模型（LLM）应用程序开发领域的行业标杆。尤其是其率先推出的ChatGPT及大型Transformer模型，极大地激发了全球范围内开发者们的创新热情和无限想象空间。

其次，随着人工智能技术的不断进步，未来将呈现出更加多元化的发展趋势，其中55%的受访者表示倾向于采用多智能体（Multi-Agent）架构。LangChain和LlamaIndex等领先的Agent和工具编排框架应运而生，旨在进一步提升大语言模型（LLM）的功能表现。通过整合多个Agent以实现对LLM功能的高效管理和优化，如改善AI推理效果、消除偏见问题以及集成外部数据源等。

值得注意的是，大部分应用程序并未直接依赖于向量数据库和搜索服务，仅有18%的受访者表示会使用向量检索功能。然而，借助向量数据库和向量搜索技术，应用程序能够有效处理大规模非结构化数据集（涵盖文本、图像、视频或音频等多种类型）的分类任务。尽管此项功能尚未得到广泛应用，但其强大的搜索能力和更为精准的上下文理解，无疑将为推荐系统带来显著提升。

此外，近年来聊天机器人应用逐渐崛起，已有31%的受访者表示正在使用此类产品，且这一比例仍在持续上升之中。聊天机器人的出现为用户提供了一种迭代式完善答案的途径，从而为与大语言模型（LLM）进行流畅、贴近人性化的对话交流创造了广阔空间。相比之下，传统的LLM应用程序往往采用单一目标的文本输入方式，无法实现对话优化。

#### #### 聊天机器人现状

在论述聊天机器人的发展现状时，我们必须深入探讨其发展演变历程。根据不同的历史时期及其特点，我们可以将它大致划分为如下四个阶段。

第一阶段，该阶段跨越了20世纪60至80年代，在这个时期，我们主要依赖于“词典+规则”的自然语言处理模式。两个具有代表性的例子是约瑟夫·维森鲍姆的ELIZA以及肯尼思·科尔比的PARRY，它们通过对特定关键词的匹配来对人类提出的问题做出回应，但是这种交流方式显得过于机械化和程序化。在此之后，诸如瑞克特、专家系统、尤内克斯顾问等聊天机器人相继问世，虽然它们大多沿袭了这一理念，但由于早期技术条件的限制，例如使用简单的编程语言、有限的数据库以及基于关键词匹配的回复策略，导致人机交流的效果受到了极大的限制。

第二阶段，该阶段涵盖了20世纪90年代至21世纪初，以勒布纳人工智能奖的设立为重要标志，这一时期聊天机器人的智能化进程得到了显著加快。在这段时间里，基于统计模型的聊天机器人如阿尔伯特一号、爱丽丝和埃尔伯特等应运而生。值得一提的是，爱丽丝凭借其高度智能化的表现，曾三度荣获勒布纳奖。然而，尽管爱丽丝通过大量规则的运用提高了对话功能，但她所建立的基于AIML的闲聊系统却难以维持较长时间的对话，因此并未完全通过图灵测试。

第三阶段，该阶段始于21世纪初，一直延续至本世纪20年代，这期间智能聊天机器人迎来了私人用户数字助理的蓬勃发展，如Siri、Cortana、Google Assistant和Alexa等。这些机器人基于深度学习模型，融合了语音识别和信息检索技术，从而能够提供更加个性化的服务。它们实现了双向交流，能够主动响应用户的需求并推荐相关信息，但相较于真正的人类对话能力而言，仍然存在一定的差距。对于那些无法直接回答的问题，它们可能仅仅提供网页链接，这无疑会影响到人机交流的效率。

第四阶段，也就是本世纪20年代，以采用“预训练+微调”模式的ChatGPT为代表，这标志着生成式人工智能领域取得了重大突破。ChatGPT背后的LLM模型拥有极其庞大的参数规模和强大的计算能力，再加上人类反馈的强化学习技术，使得它能够不断地提升语言生成能力，实现高质量且一致性的文本输出，并且能够与用户保持稳定的对话关系。在此基础之上，GPT-4进一步拓展了多模态功能，支持文本和图片的输入，从而深化了人机交流的体验。展望未来，随着通用人工智能技术的不断进步，视觉化、沉浸式、虚实融合的交互方式将会逐渐成为人机交流的关键要素，这预示着人机交流能力将迎来前所未有的爆发式增长。

综上所述，聊天机器人经历了长时间的发展，随着大模型的诞生，迎来了爆发式的增长。

总的来说，聊天机器人历经了漫长的发展历程，如今伴随着大模型的诞生，终于迎来了爆发式的增长。

### 国外研究现状

研究现状这部分需要修改 这里肯定有问题

在全球范围内，关于聊天机器人的研发工作广受各界瞩目。众多享有盛誉的科技巨头及顶尖科研机构纷纷投身其中，持续深入探寻这一领域的前沿成果。

尤具代表性的例子当属OpenAI所推出的ChatGPT，作为人工智能领域第四代技术的杰出代表，ChatGPT借助海量的文本数据进行预训练，同时结合人类反馈进行精细调整，从而实现更为自然且流畅的人机对话效果。目前，ChatGPT已成功融入各类应用软件之中，为广大用户带来个性化的解答与交流体验。

同样值得一提的是谷歌旗下的Google Assistant，作为第三代智能助手的代表之作，其融合了聊天机器人的核心功能，支持用户通过语音或文字方式与其进行互动交流。Google Assistant不仅能解答各类疑问、提供丰富资讯，还可协助用户管理日常行程、操控智能家居设备等，全方位满足用户的多样化需求。

除此之外，微软的Cortana亦是一款备受赞誉的智能助手，用户可通过Windows操作系统及移动设备轻松访问使用。Cortana能够协助用户妥善安排日程、发送电子邮件、搜索所需信息、获取实时天气及交通状况等，助力用户更高效地完成各项任务。近年来，微软更进一步推出了Copilot，作为Cortana的升级版，Copilot在功能及性能方面均有显著提升。

**Thoppilan等人推出了LaMDA系列聊天机器人LLM，最多可达1370亿个参数，其重点在于安全性（通过对人类注释的监督微调）和事实依据（通过访问外部知识源）。值得注意的是，较小的LaMDA模型（20亿参数）经过微调后，其对话质量和安全性/依据评分方面的表现与较大的LaMDA模型（1370亿参数）相似，而后者未经微调。LaMDA模型作为Bard聊天机器人服务的一部分发布。然而，Bard的最新版本现在使用了PaLM 2 LLM 。**

**Glaese等人提出了一款名为Sparrow的聊天机器人，基于具有700亿参数的Chinchilla LLM，并使用RLHF（第2.9节）针对23个规则进行微调，使得模型更加实用、准确、无害。Sparrow还结合了外部知识，使用检索模型通过Google搜索查询提供证据。RLHF方法在输出偏好和规则违反率方面优于仅使用对话提示和监督微调的方法。**

**同样，OpenAI使用监督微调和RLHF（第2.9节）来专门训练GPT-3.5 LLM用于对话的ChatGPT聊天机器人。GPT-4 [398]是ChatGPT Plus聊天机器人的底层模型，但其训练和架构细节尚未公开。**

**Shuster等人介绍了基于OPT-175 LLM的1750亿参数聊天机器人BlenderBot-3，采用监督微调。BlenderBot-3通过执行互联网搜索和检索从先前输出生成的基于文本的长期记忆的模块来结合外部知识，以提高长时间交互中的性能表现。**

**Köpf等人发布了OpenAssistant Conversations数据集，这是一个人类标注的交互数据集，并使用它来对Pythia和LLaMA模型（最多300亿参数）进行聊天机器人应用的指令微调。为了帮助对齐最终模型，该数据集是根据指导方针生成的，以确保响应礼貌、有用、简洁、友好和安全意识。LLaMA 30B版本目前用于HuggingChat聊天机器人应用。**

**微调聊天机器人的一个关键挑战是创建高质量对话的广泛训练数据集。为了解决这个问题，Chen等人展示了如何使用现有的LLM（OPT 30B）根据少量专家编写的示例生成高质量合成对话数据集。众包工人评估生成的对话在以下指标上与现有的人生成的数据集相当：有趣、连贯、自然和一致。Chen等人展示了合成数据集可用于微调聊天机器人（BlenderBot 400M），并且只比使用人生成的数据集进行微调的性能略低。**

**聊天机器人旨在具有广泛的通用性，这也使评估其全部能力范围变得困难。Kocon ́等人使用涵盖多样化能力集**

### 国内研究现状

在我国，基于langchain框架的综合性聊天机器人研究正呈现出蓬勃发展的态势。伴随着诸如GPT-3、GPT-4等大规模语言模型（LLM）技术的不断突破，以及langchain框架的日益完善，越来越多的学者和开发人员开始致力于运用这些尖端技术打造高性能、智能化的聊天机器人。

国内诸多知名高等学府及科研机构，如清华大学、北京大学、中国科学院等，已在langchain框架的基础上开展了大量的研究与实践工作，提出了诸多富有创新性的思路与解决方案。他们不仅专注于提升聊天机器人的语言理解与生成能力，更积极探索如何将聊天机器人广泛应用于教育、医疗、金融等多个实际场景之中。

此外，国内一批互联网企业，如百度、腾讯、阿里巴巴等，也纷纷加大投入力度，全力开发基于langchain框架的聊天机器人。这些聊天机器人不仅具备卓越的智能水平，而且能够紧密贴合用户的实际需求，提供个性化的优质服务。

### ### 本文主要研究工作及结构

#### #### 主要工作

本论文针对大规模语言模型在应用开发方面展开研究，深度剖析传统大规模语言预测模型的开发流程，掌握相关核心开发技术如提示词汇构造、智能体构建以及数据检索生成等关键环节；对langchain框架在大规模语言模型应用开发领域的独特优势有深刻认识，以解决传统大规模模型应用程序开发面临的诸多难题。此外，本论文还对聊天机器人领域进行了深入探讨，旨在揭示其运作机制和潜在功能。最后，本论文将运用langchain开发框架，设计并构建一个功能完备的聊天机器人程序。该聊天机器人不仅具备卓越的文本解析能力，而且能够与数据库进行高效互动，甚至可以执行网络搜索等复杂任务，为用户提供丰富且便捷的交互体验。

#### #### 主要主题

第一章作为本书的序言部分，主要从课题的角度出发，深入探讨了聊天机器人系统的理论基础、发展背景及其在现代社会中的实际价值与意义，同时也参照国际视野，对这一领域的研究进展状况作了详尽综述。在此之后，我们简要概述了论文中所涉及到的主要科研方向及研究内容，及文章的整体结构布局。

第二章以丰富的实例为依托，对大语言模型应用开发过程中所涉及的关键性知识进行了详细阐述，包括大模型开发过程中使用的各类核心技术组件，以及具体的大语言模型应用开发流程等方面的内容。

第三章则是对langchain框架进行全方位的剖析，详细解读了该框架中各个重要组成部分的功能特性，并对其在整个langchain生态系统中的作用进行了简要介绍。

第四章的重点在于如何利用langchain框架来构建个性化的聊天机器人，以及如何对聊天机器人的性能表现进行科学评估。

## 大模型应用开发

### 大模型应用开发简介

我们将采用大语言模型作为核心功能，利用大语言模型的卓越理解和生成能力，并结合特定的数据或业务逻辑，打造具有独特功能的应用程序，称之为大模型开发。尽管开发大模型相关应用的技术核心在于大语言模型，但通常会通过调用API或开源模型来实现核心理解和生成，利用提示工程来实现对大语言模型的操控。因此，尽管大型模型是深度学习领域的集大成者，但大型模型开发更多地被视为一个工程问题。

在大型模型开发过程中，我们通常不会对模型进行大幅修改，而是将大型模型视为一种调用工具，通过提示工程、数据工程、业务逻辑分解等手段来充分发挥大型模型的潜力，以适应应用任务的需求，而非将精力集中于优化模型本身。因此，作为大型模型开发的初学者，我们无需深入研究大型模型的内部原理，而更应掌握运用大型模型的实践技巧。

此外，以调用和发挥大型模型为核心的大型模型开发与传统的AI开发在整体思路上存在显著差异。大语言模型的两大核心能力：指令理解和文本生成为复杂业务逻辑提供了简洁的替代方案。在传统的AI开发中，我们需要先将极其复杂的业务逻辑逐步拆分，针对每个子业务构建训练数据和验证数据，并对每个子业务进行模型训练和优化，最终形成完整的模型链路来解决整个业务逻辑。然而，在大型模型开发中，我们将尝试使用提示工程来取代子模型的训练和优化，通过提示链路的组合来实现业务逻辑，通过一个通用的大型模型和多个业务提示来完成任务，从而将传统的模型训练和优化转变为更为简单、轻松且成本较低的提示设计和优化。

同样，在评估思路方面，大型模型开发与传统的AI开发也存在本质区别。传统的AI开发需要事先构建训练集、测试集和验证集，通过在训练集上训练模型、在测试集上优化模型、以及在验证集上最终验证模型效果来实现性能评估。然而，大型模型开发更加敏捷、灵活，我们通常不会在初期明确地定义训练集和验证集，由于不再需要训练子模型，我们不再构建训练集，而是直接从实际业务需求出发构建小批量验证集，设计合理的提示来满足验证集的效果。然后，我们将持续从业务逻辑中收集当前提示的不良案例，并将这些案例纳入验证集中，针对性地优化提示，最终实现良好的泛化效果。

下图为传统AI开发和传统大模型开发图片（重新做一个）



#### 提示工程

提示工程（Prompt Engineering），亦可称为上下文提示（In-Context Prompting），即通过构建结构化文本等手段，改良提示词以导向我们预期的语言模型输出结果。借助提示词工程技术，在无需更新模型权重的前提下，语言模型得以完成各类任务。此项技术主要依赖于实践经验，且提示词工程方法的效果在不同的模型中可能存在显著差异，因此需要进行大量的实验与探索。

研究关注提示词开发及优化，以协助用户将大规模语言模型（Large Language Model, LLM）应用于各类应用场景及研究领域。掌握提示工程相关技术将助力用户更深入地理解大规模语言模型的潜力及其局限性。

研究者可运用提示工程提升大规模语言模型处理复杂任务场景的能力，例如问答及算术推理能力。开发者则可通过提示工程设计、研发高效的工程技术，实现与大规模语言模型或其他生态工具的无缝衔接。

提示工程并非仅限于设计及研发提示词。其涵盖了与大规模语言模型交互及研发的多种技能及技术。提示工程在实现与大规模语言模型的交互、对接，以及理解大规模语言模型能力方面均发挥着关键作用。用户可通过提示工程提升大规模语言模型的安全性，并赋予其更多功能，例如借助专业领域知识及外部工具增强大规模语言模型的能力。

鉴于对大规模语言模型的浓厚兴趣，我们编撰了这份全新指南，涵盖了大规模语言模型相关的论文研究、学习指南、模型、讲座、参考资料、大规模语言模型能力以及与提示工程相关的其他工具。

时至今日，改进提示词无疑有助于在各类任务中取得更优成果。这正是提示工程背后的核心理念。

尽管基础示例颇具趣味性，但在此章节中，我们将介绍更为高级的提示工程技术，使我们能够完成更为复杂且富有挑战性的任务。

以下将列举几个常用的提示词技术

##### 链式思考（CoT）提示

在 [Wei等人（2022）(opens in a new tab)](https://arxiv.org/abs/2201.11903) 中引入的链式思考（CoT）提示通过中间推理步骤实现了复杂的推理能力。您可以将其与少样本提示相结合，以获得更好的结果，以便在回答之前进行推理的更复杂的任务。

##### 检索增强生成 (RAG)

通用语言模型经微调便能胜任多类常见任务，如解析情感及辨识命名实体等。此类任务无需额外的背景知识支持即可顺利完成。

以语言模型为基础，构建一套系统，接入外部知识源，便可完成更为复杂且知识密集型的任务。此种实现方式与事实更为一致，生成的答案更为可靠，同时也有助于缓解“幻觉”问题。

Meta AI的研究人员引入了一种名为检索增强生成（Retrieval Augmented Generation，RAG）的方法来应对此类知识密集型的任务。RAG将信息检索组件与文本生成模型相结合。RAG可进行微调，其内部知识的修改方式高效，无需对整个模型进行重新训练。

RAG接收输入，检索出一组相关/支撑的文档，并提供文档来源（如维基百科）。这些文档作为上下文和输入的原始提示词组合，送至文本生成器获得最终输出。如此，RAG更能适应事实随时间变化的情况。这一点尤为重要，因为LLM的参数化知识是静态的。RAG使得语言模型无需重新训练即可获取最新信息，基于检索生成可靠的输出。

Lewis等人（2021）提出了一种通用的RAG微调方法。该方法利用预训练的seq2seq作为参数记忆，使用维基百科的密集向量索引作为非参数记忆（通过神经网络预训练的检索器访问）。

##### ReAct 框架

Yao等人于2022年（新标签中打开）首次引入了一种架构，在此架构中，LLMs可以交互地生成推理轨迹和任务特异性操作。

通过生成推理轨迹，模型能够引导、追踪并更新操作计划，甚至可以处理异常情况。操作步骤允许与外部资源（如知识库或者环境）进行交互，并收集相关信息。

ReAct架构能使LLMs与外部工具进行互动，从而获取进一步的信息，以做出更精确和实用的响应。

研究结果表明，ReAct在语言和决策任务中的性能要优于当前若干最先进水平所规定的基准。ReAct还提升了LLMs的人类可解释性和可信度。总体而言，作者发现将ReAct与链式思考（CoT）相结合的最佳策略是在推理过程中同时运用内部知识和获取到的外部信息。

#### 数据工程

何谓大规模模型的数据工程？当前，研发诸如GPT模型或BERT等模型时，存在两种策略。第一种策略以模型为主导，对数据的关注程度较低，不断优化模型的架构；第二种策略则以数据为导向（Data-Driven），这已成为当前网络算法领域的共识，算法的本质在于处理数据，其核心思想是保持模型不变，通过提高数据质量来提升模型性能，从而持续提升训练数据的质量。

以数据为导向的AI核心在于训练数据开发、推理数据开发以及数据维护。

训练数据开发涵盖了诸多环节，包括数据的收集、数据源的确定、高质量数据标注的实施、数据预处理或数据压缩或增强的操作。例如，在进行领域微调数据时，若行业数据仅有数万条，则需将其增强至数十万条或数百万条，方能真正将领域或行业的数据纳入其中。

推理数据开发则涉及如何评估先前的训练样本，以及如何更有效地评估测试集以外的数据。

数据维护则要求将与数据相关的工作形成一个闭环，包括对数据的深入理解。在使用训练数据进行训练时，若发现数据存在问题，应能够定位到问题源于哪个数据集，并进行针对性的追踪和优化。大模型的数据工程包括很多环节，比如收集数据、数据标注、数据预处理、数据增强、微调数据等等。

大模型的模型开发和数据工程，哪个更重要，是一个经常被问到的话题，其实它没有一个明确的答案，数据和模型都是很重要的。更进一步说，算法结果的好坏则主要依赖于数据质量和模型结构的选择。

在实际应用中，数据工程的重要性往往被低估，因为它需要大量的时间和精力，而且往往没有直接的成果。但是，如果没有高质量的数据，模型的效果就会大打折扣。因此，数据工程是大模型开发中不可或缺的一部分。

### 大模型应用开发流程

目标设定阶段。在正式展开研发行动之前，我们首要的任务就是为开发定下明确的目标，即定义即将开发之应用的适用场景、目标受众群体以及核心价值所在。尤其对于独立开发者或者规模较小的开发团队而言，更应优先设定易于实现的最小化目标，并可从创建一个最小可行产品（mvp）入手，逐步完善和优化产品功能。

功能设计环节。在确立了开发目标之后，我们便需着手设计应用所需提供的各项功能，同时还需明确每个功能的大致实现逻辑。尽管我们借助大型模型来简化了业务逻辑的分解过程，但对业务逻辑的深度理解往往能带来更为出色的提示效果。同样地，对于独立开发者或者规模较小的开发团队来说，首先应当明确应用的核心功能，随后扩展设计核心功能的上下游功能；例如，如果我们计划打造一款个人知识库辅助工具，那么核心功能便是利用个人知识库中的内容来解答用户提出的问题，而上游功能则涉及用户上传知识库，下游功能则包括用户手动校正模型回答等子功能。

整体架构搭建阶段。当前，绝大多数大型模型应用均采用特定数据库+提示器+通用大型模型的架构模式。因此，我们需要根据已设计好的功能，搭建项目的整体架构，以实现从用户输入至应用输出的全流程顺畅运行。通常情况下，我们建议基于LangChain框架进行开发。LangChain提供了Chain、Tool等架构的实现方式，我们可以在此基础上进行个性化定制，以实现从用户输入至数据库、再到大型模型最终输出的整体架构连接。

数据库搭建阶段。个性化大型模型应用需要有个性化数据库作为支持。鉴于大型模型应用需要进行向量语义检索，因此一般会选择使用如chroma这样的向量数据库。在此步骤中，我们需要收集相关数据并进行预处理，然后将其向量化并存储到数据库之中。数据预处理通常涵盖从各种格式向纯文本的转换，如pdf、markdown、html、音视频等，以及对错误数据、异常数据、脏数据的清理工作。预处理完毕后，需要进行切片、向量化操作，以建立起个性化数据库。

提示工程建设阶段。优质的提示器对大型模型的性能有着显著的影响，因此我们需要逐步迭代构建高质量的提示工程，以提高应用的运行效率。在此阶段，我们首先应明确提示器设计的一般原则及技巧，构建一个源自实际业务的小型验证集，并基于此验证集设计出符合基本需求、具备基本能力的提示器。

验证迭代阶段。验证迭代在大型模型开发过程中占据着至关重要的地位，它通常是通过持续发现不良案例并针对性地改进提示工程来提升系统性能、应对各种边界情况。在完成上一步的初始提示器设计后，我们应进行实际业务测试，探讨可能出现的边界情况，找出不良案例，并针对性地分析提示器存在的问题，以便不断迭代优化，直至达到一个相对稳定、能够基本实现预期目标的提示器版本。

前后端搭建阶段。在完成提示工程的建设及其迭代优化之后，我们已经成功实现了应用的核心功能，此时便可充分发挥大型语言模型的强大实力。接下来，我们需要搭建前后端，设计产品界面，使我们的应用得以顺利上线并成为真正的产品。前后端开发是一个非常经典且成熟的领域，在此不再赘述，我们将重点介绍两款有助于个体开发者迅速搭建可视化页面并实现演示上线的框架：Gradio和Streamlit。

体验优化阶段。在完成前后端的搭建之后，应用便可正式上线供用户体验。接下来，我们需要进行长期的用户体验跟踪，记录下可能出现的不良案例与用户的负面反馈，然后针对性地进行优化调整。

### 大模型下的聊天机器人

#### 聊天机器人介绍

聊天机器人（ChatBot）是一种能够通过对话或文字方式进行交流的计算机程序，其功能不仅能模拟人类对话，且需通过图灵测试，重要的是它还需要具备实际应用价值，如客户服务或信息获取等。

有些聊天机器人配备了自然语言处理系统，然而，大多数简单的系统只能捕获输入的关键词，然后从语料库中查找最合适的应答句。当前，聊天机器人已成为虚拟助手的一部分，可以与众多组织的应用程序、网站和通讯平台相连接。非助手应用程序包括娱乐性质的聊天室，研究和特定产品推广，以及社交机器人等。

随着OpenAI的ChatGPT（使用GPT-3或GPT-3.5)在2022年12月的发布和普及，该领域受到了广泛关注。人工智能可以从理解概念出发，给出专业甚至具有创新性的答案。随后，许多公司和机构纷纷推出了大型语言模型和聊天机器人，这类产品能够胜任复杂的指令并提供稳定的切题反馈。不久之后，微软推出了Bing chat（使用OpenAI的GPT-4），谷歌推出了Bard，进一步与生产力平台相融合。

通用聊天机器人（对话代理）结合了信息检索、多轮交互和文本生成（包括代码）的任务。

**机器人聊天组件的功能及作用**

**检索功能（Retrieval, Optional）：此项为可选组件之一，其主要职责是从大规模的知识库或数据集中检索与用户问题相关联的信息。倘若系统判断需要借助更多相关信息以提供更为精确的答复时，便可能会启动该检索功能。**

**存储功能（Storage）：此组件承担着系统内所有数据的存储与管理工作，其中涵盖了历史问题、解答结果、知识库等各类数据。通过该功能的实现，确保了系统在必要时刻能迅速获取所需信息。**

**提取块功能（Retrieved Chunks）：若启用了检索功能，那么此部分将会展示从检索过程中所获取到的相关信息的片段或区块。这些片段或区块有望提升语言模型（LLM）的生成能力。**

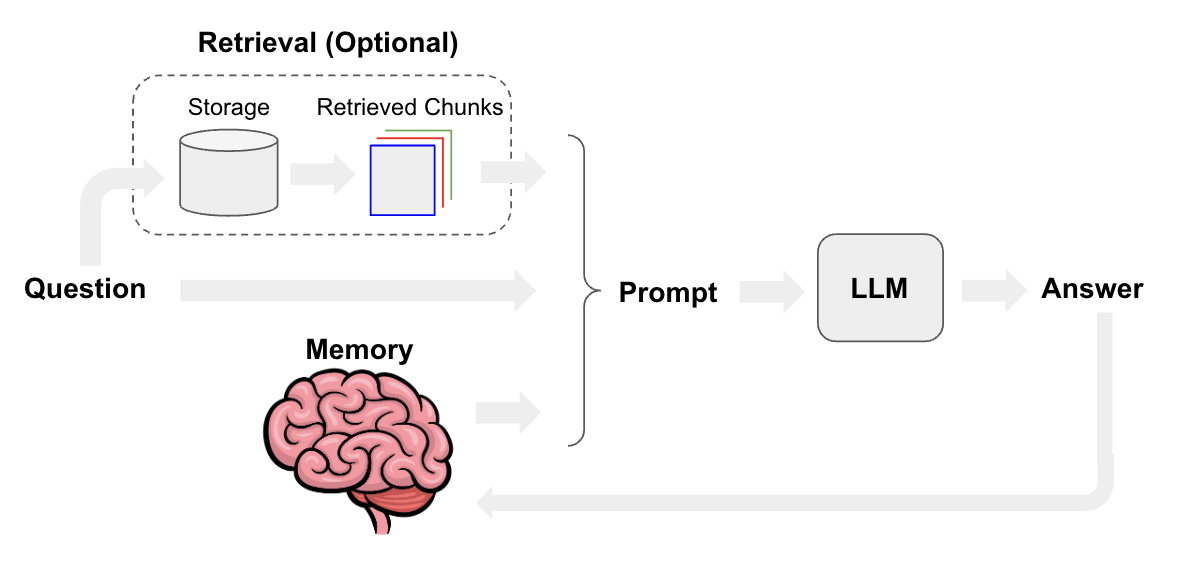
**问题功能（Question）：此部分为用户向系统输入的文本内容，代表了用户希望了解的信息或寻求的答案。**

**提示功能（Prompt）：提示功能则是向语言模型（LLM）发送的文本信息，旨在引导模型更好地理解并回复用户提出的问题。该功能通常会依据用户所提问题进行个性化定制，从而保证模型能够生成相应的答案。**

**语言模型（LLM）：作为系统的核心组件，语言模型（LLM）是一种基于神经网络技术构建的模型，用以解析用户提出的问题并生成相应的答案。LLM在生成答案时，会利用存储功能中提取出的信息（如有）以及提示功能提供的信息。**

**答案功能（Answer）：此部分为系统根据用户所提问题及提示信息生成的文本应答，是对用户查询的直接反馈。**

#### 聊天机器人开发流程



这副图片的retrieval 需要改一下

聊天机器人的开发需要结合上文中所说，使用提示词工程，并加入记忆力机制通过大语言模型完成

**构建聊天机器人的流程**：

1. **接收问题**：首先，系统接收用户输入的问题。
2. **生成提示**：系统根据问题生成一个或多个提示，这些提示被设计用来指导语言模型（LLM）如何理解和回答用户的问题。
3. **检索（可选）**：如果系统认为需要额外的信息来生成更准确的答案，它可能会使用检索组件从大型知识库或数据集中检索相关信息。
4. **处理**：系统使用语言模型（LLM）来处理问题、提示和（如果有的话）检索到的信息。LLM使用这些信息来生成一个答案。
5. **返回答案**：最后，系统将生成的答案返回给用户。

这个流程展示了如何结合检索、存储和语言模型来构建一个功能强大的聊天机器人，它能够理解用户的问题并生成相关的答案。

### 本章小结

本章介绍了。。。

## langchain大语言模型开发框架

### langchain框架介绍

**LangChain 框架是一个开源工具，充分利用了大语言模型的强大能力，以便开发各种下游应用。它的目标是为各种大语言模型应用提供通用接口，从而简化应用程序的开发流程**。具体来说，LangChain 框架可以实现数据感知和环境互动，也就是说，它能够让语言模型与其他数据来源连接，并且允许语言模型与其所处的环境进行互动。

利用 LangChain 框架，我们可以轻松地构建如下所示的 RAG 应用（[图片来源](https://github.com/chatchat-space/Langchain-Chatchat/blob/master/img/langchain+chatglm.png)）。在下图中，每个椭圆形代表了 LangChain 的一个模块，例如数据收集模块或预处理模块。每个矩形代表了一个数据状态，例如原始数据或预处理后的数据。箭头表示数据流的方向，从一个模块流向另一个模块。在每一步中，LangChain 都可以提供对应的解决方案，帮助我们处理各种任务。

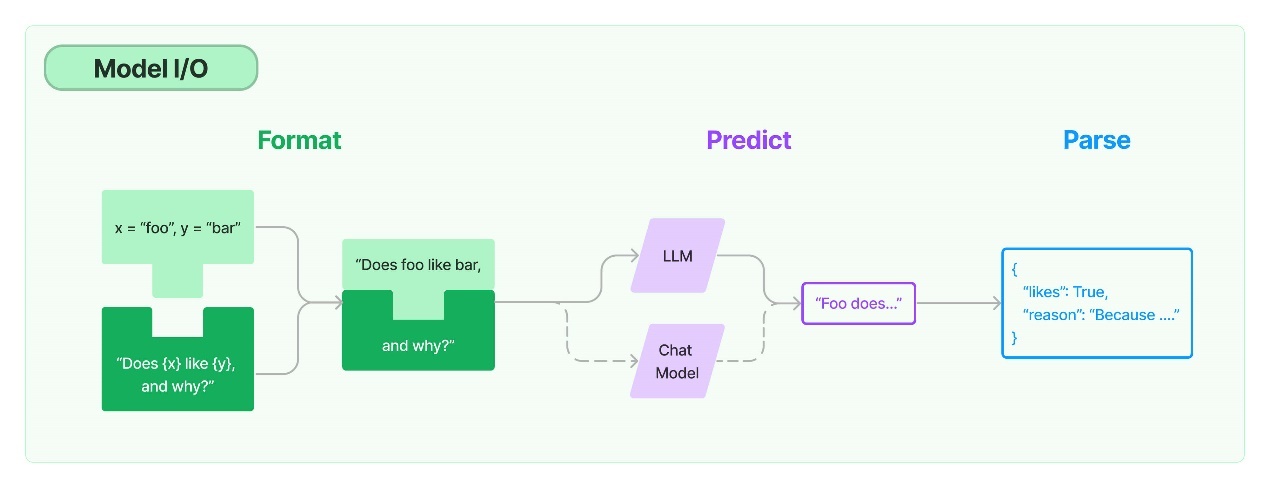
在 LLM 技术领域的迅猛发展浪潮中，LangChain 作为一个不断进化的创新平台，持续推动着技术边界的拓展。2024 年 1 月 9 日，LangChain 正式发布了其稳定版本 **v0.1.0**，这一里程碑式的更新，为开发者带来了全面而强大的功能支持。其涵盖了模型的输入与输出处理、数据连接、链式操作、记忆机制、代理服务以及回调处理等关键组件，为 LLM 应用的开发和部署提供了坚实的基础。 同时，LangChain 的持续优化和功能迭代，未来将带来更多创新特性和性能提升。

### LangChain表达式语言

LangChain Expression Language (LCEL) 是LangChain 工具包的重要补充，旨在提高文本处理任务的效率和灵活性。LCEL 允许用户采用声明式方法来组合链，便于进行流处理、批处理和异步任务。其模块化架构还允许轻松定制和修改链组件。LCEL 的优势之一是它使用户更容易个性化链的不同部分。链的声明性和模块化特性允许轻松地交换组件。此外，现在的提示更加明显，可以轻松地修改以适应特定的用例。在 LangChain 中，提示只是默认值，但是可以为生产应用程序进行更改。

### 核心组件

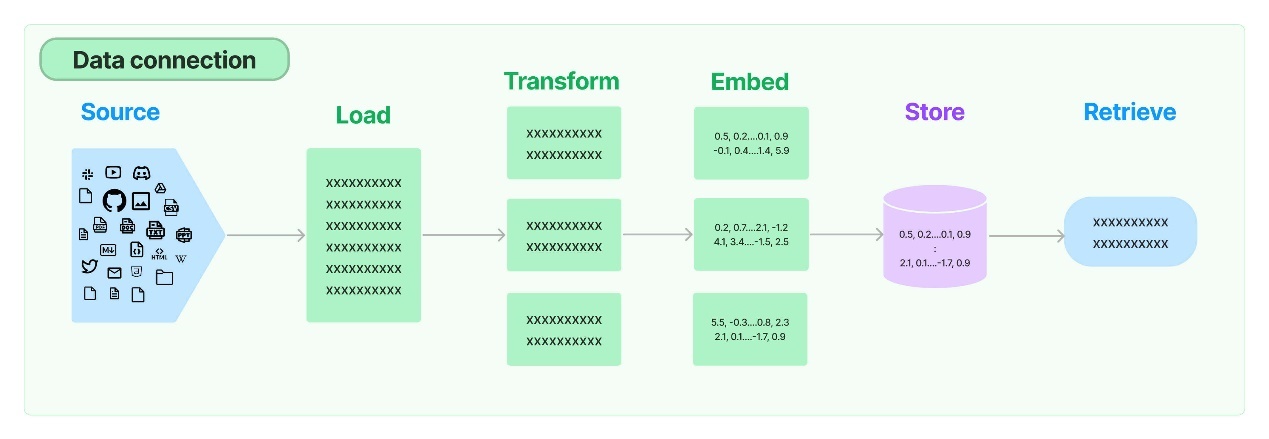
#### Model I/O



任何语言模型应用程序的核心元素都是......模型。 LangChain 为您提供了与任何语言模型交互的构建块。

Langchain在该部分提供了一些大模型的抽象类，可以使用他们更快捷方便的构建我们的应用程序。 同时还有方便在代码中使用的输入处理和输出处理功能。

#### #### 检索（Retrieval）



许多大模型需要特定于用户的数据，这些数据不属于模型训练集的一部分。 实现这一目标的主要方法是通过检索增强生成（RAG）。 在此过程中，将检索外部数据，然后在执行生成步骤时将其传递给 LLM。 LangChain 提供了 RAG 应用程序的所有构建模块 - 从简单到复杂。 文档的这一部分涵盖了与检索步骤相关的所有内容 - 例如 数据的获取。 虽然这听起来很简单，但实际上可能非常复杂。 这包含几个关键模块。

文件装载机 文档加载器加载来自许多不同来源的文档。 LangChain 提供 100 多种不同的文档加载器，并与该领域的其他主要提供商（例如 AirByte 和 Unstructed）集成。 LangChain 提供了从所有类型的位置（私有 S3 存储桶、公共网站）加载所有类型文档（HTML、PDF、代码）的集成。 文本分割 检索的关键部分是仅获取文档的相关部分。 这涉及几个转换步骤来准备文档以供检索。 这里的主要任务之一是将大文档分割（或分块）为更小的块。 LangChain 提供了多种转换算法来执行此操作，以及针对特定文档类型（代码、Markdown 等）优化的逻辑。

文本嵌入模型 检索的另一个关键部分是为文档创建嵌入。 嵌入捕获文本的语义，使您能够快速有效地找到文本的其他相似部分。 LangChain 提供与超过 25 种不同嵌入提供商和方法的集成，从开源到专有 API，让您可以选择最适合您需求的一种。 LangChain提供了标准的接口，让您可以轻松地在模型之间进行切换。

矢量商店 随着嵌入的兴起，需要数据库来支持这些嵌入的高效存储和搜索。 LangChain 提供与 50 多种不同矢量存储的集成，从开源本地矢量存储到云托管专有矢量存储，让您可以选择最适合您需求的一种。 LangChain公开了一个标准接口，让您可以轻松地在向量存储之间进行交换。 猎犬 一旦数据进入数据库，您仍然需要检索它。 LangChain支持多种不同的检索算法，是我们增加最大价值的地方之一。 LangChain支持易于上手的基本方法——即简单语义搜索。 然而，我们还在此基础上添加了一系列算法以提高性能。 这些包括： 父文档检索器：这允许您为每个父文档创建多个嵌入，从而允许您查找较小的块但返回更大的上下文。

自查询检索器：用户问题通常包含对某些内容的引用，这些内容不仅是语义的，而且表达了一些可以最好地表示为元数据过滤器的逻辑。 自查询允许您从查询中存在的其他元数据过滤器解析出查询的语义部分。 集成检索器：有时您可能希望从多个不同的源检索文档，或使用多种不同的算法。 集成检索器可以让您轻松地做到这一点。 和更多！

索引 LangChain Indexing API 将您的数据从任何来源同步到向量存储中，帮助您： 避免将重复的内容写入矢量存储 避免重写未更改的内容 避免在未更改的内容上重新计算嵌入 所有这些都可以节省您的时间和金钱，并改善您的矢量搜索结果。

#### #### 组成

本节包含将其他任意系统（例如外部 API 和服务）和/或 LangChain 原语组合在一起的更高级别组件。

本节的一个很好的入门知识是阅读 LangChain 表达式语言部分，并熟悉通过管道和提供的各种原语构建序列。

本节涵盖的组件是：

工具

工具为法学硕士和其他组件提供与其他系统交互的接口。 示例包括维基百科、计算器和 Python REPL。

代理商

代理使用语言模型来决定要采取的操作，通常由工具定义。 它们需要一个执行器，即代理的运行时。 执行器实际上调用代理，执行它选择的工具，将操作输出传递回代理，然后重复。 代理负责解析先前结果的输出并选择下一步。

链条

其他基元和组件的构建块式组合。

##### ##### 工具

具是代理、链或 LLM 可以用来与世界交互的接口。

它们结合了一些东西：

工具名称

该工具是什么的描述

工具输入内容的 JSON 架构

要调用的函数

工具的结果是否应直接返回给用户

拥有所有这些信息非常有用，因为这些信息可用于构建采取行动的系统！ 名称、描述和 JSON 模式可用于提示 LLM，以便它知道如何指定要执行的操作，然后调用的函数相当于执行该操作。

工具的输入越简单，法学硕士就越容易使用它。 许多代理只能使用具有单个字符串输入的工具。 有关代理类型列表以及哪些代理类型可处理更复杂的输入，请参阅此文档

重要的是，名称、描述和 JSON 模式（如果使用）都在提示中使用。 因此，清晰并准确地描述如何使用该工具非常重要。 如果 LLM 不了解如何使用该工具，您可能需要更改默认名称、描述或 JSON 架构。

##### ##### agents

代理的核心思想是使用语言模型来选择要采取的一系列操作。 在链中，一系列操作被硬编码（在代码中）。 在代理中，语言模型被用作推理引擎来确定要采取哪些操作以及按什么顺序。

##### ##### 链

链是指调用序列——无论是法学硕士、工具还是数据预处理步骤。 主要支持的方法是使用 LCEL。

LCEL 非常适合构建链条，但使用现成的链条也很好。 LangChain支持的现成链有两种：

使用 LCEL 构建的链条。 在这种情况下，LangChain提供了更高级的构造方法。 然而，幕后所做的一切都是用 LCEL 构建一条链。

[旧版] 通过继承旧版 Chain 类的子类而构造的链。 这些链在底层不使用 LCEL，而是独立的类。

我们正在致力于创建创建所有链的 LCEL 版本的方法。 我们这样做有几个原因。

以这种方式构造的链很好，因为如果你想修改链的内部结构，你可以简单地修改 LCEL。

这些链本身支持开箱即用的流式传输、异步和批处理。

这些链在每一步都会自动获得可观察性。

此页面包含两个列表。 首先是所有 LCEL 链构造函数的列表。 其次，所有遗留链的列表。

### ### langchain生态

#### #### langsmith

一个聊天机器人的性能并非一成不变。随着数据量的增加和用户需求的变化，模型需要不断地进行训练和优化。这就是LangSmith发挥作用的地方。作为框架的模型训练和优化工具，LangSmith为开发者提供了丰富的算法和技巧。它支持多种模型训练方式，包括有监督学习、无监督学习等，以帮助开发者不断提升模型性能。这意味着开发者可以利用LangSmith来优化他们的聊天机器人，使其更加准确、可靠，并更好地满足用户需求。

#### #### langserve

一个优秀的聊天机器人不仅需要强大的模型和高效的开发工具，还需要稳定、可靠的服务端支持。这就是LangServe的作用所在。作为框架的服务端组件，LangServe负责提供高效且稳定的在线服务。它支持多种部署方式，包括云服务器、容器化部署等，以满足不同场景下的需求。通过LangServe，开发者可以将聊天机器人快速部署到线上，为用户提供实时、准确的回答和交互。这不仅提高了用户体验，也增强了聊天机器人的可用性和可靠性。

## 聊天机器人程序设计

在第二章介绍了大模型和聊天机器人的相关知识，第三章介绍了LangChain框架。这一章将上述知识结合起来，完成构建聊天机器人的任务。

### 总体设计目标

本文将在实现基础的聊天机器人的功能上进一步实现PDF文档交互查询、微软Bing搜索、本地数据库查询交互的功能，帮助用户完成更多方便的工作。

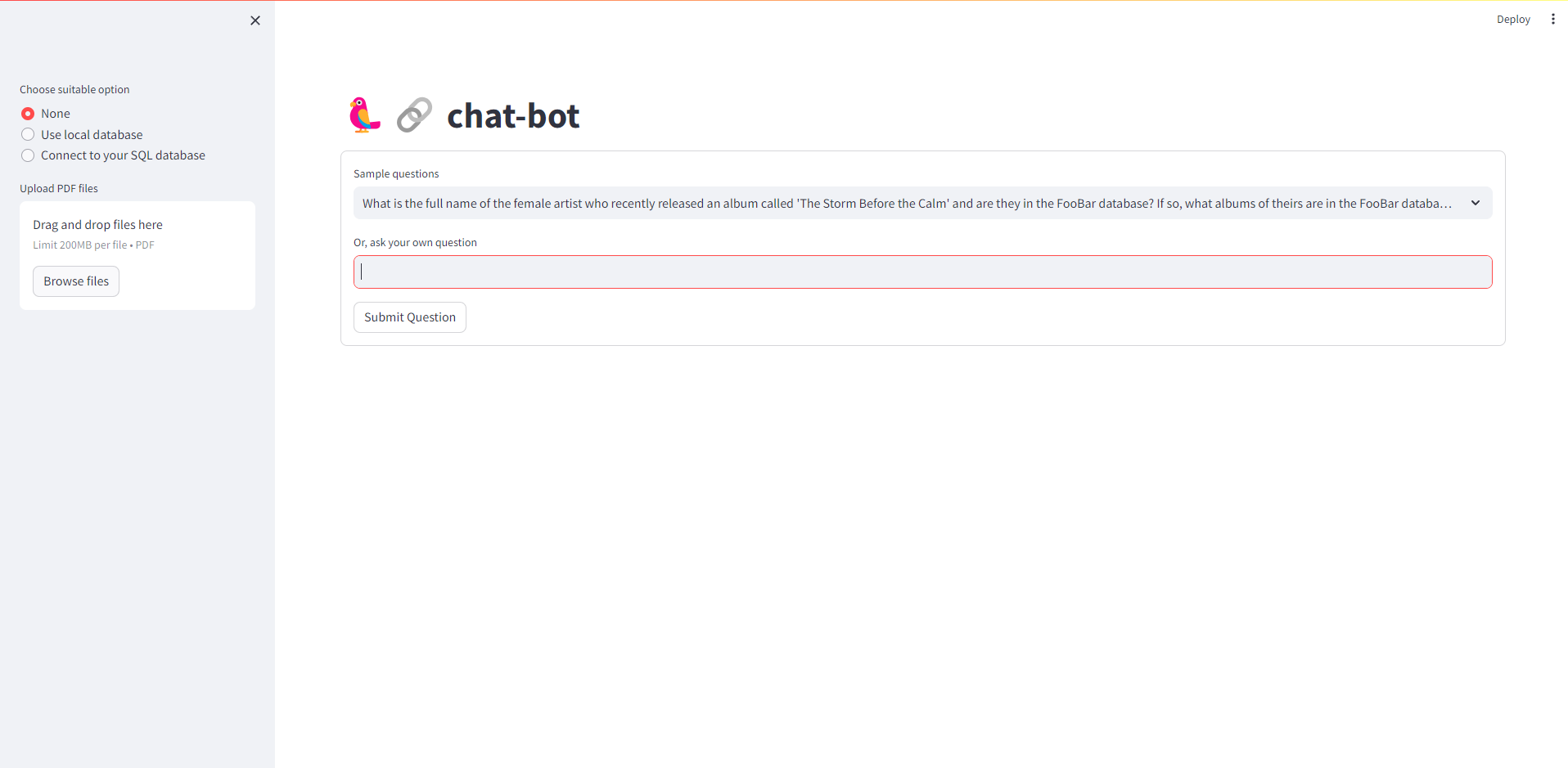
这里可以做一张图

### 前端streamlit交互界面程序设计

#### Streamlit框架简介

Streamlit是第一个专门针对机器学习和数据科学团队的应用开发框架，它是开发自定义机器学习工具的最快的方法，目的是取代Flask在机器学习项目中的地位，可以帮助机器学习工程师快速开发用户交互工具。

#### 交互界面构建与预期效果



一个聊天机器人一般只作一种用途使用，所以这里选择通过单项选择栏来切换运行不同任务的聊天机器人，在侧边栏中提供了三种不同的功能，当选择栏选择到“None”选项时候为普通聊天机器人，但具有网络搜索功能。

当选择栏目选择“使用本地数据库”或“连接到数据库”选项时候为数据库交互聊天机器人，它可以根据你的指令完成数据库查询等功能。

当在选择栏目中上传一个PDF文件后，聊天机器人将转变为文档交互机器人，它可以根据你的指令对文档的内容进行查询。

### 后端langchain服务程序设计

聊天机器人的核心业务逻辑将通过langchain框架进行构建。

将通过langchain api实现如下三种服务设计。

#### agent工具服务

#### 数据库工具服务

#### 向量数据库文本分析服务

### 聊天机器人效果分析