

Deepin wiki QAbot

摘要

本文介绍了一种基于中文大型语言模型（LLM）的文档问答机器人的开发过程，旨在通过深度学习技术，使机器人能够根据Deepin Wiki的内容准确回答用户提出的问题。通过采用预筛选和生成式问答（QA）技术，我们的系统能够在保证答案质量的同时，提高回答的自然度和流畅性。本文详细介绍了项目内容、开发流程与思路，之后介绍相比前人的创新与核心贡献，以及对未来的展望。

介绍

Deepin Wiki是一个涵盖了900多条与Deepin系统相关的中文教程和词条的资源库。为了提高用户体验，我们提出了开发一个能够根据Deepin Wiki内容回答问题的聊天机器人。该机器人不仅需要理解用户的问题，还要能够在庞大的文档集合中找到最相关的答案，并以自然语言的形式呈现给用户。我们可以将该模型扩展至其他文档资源，如玲珑使用手册，提供全自动化的部署流程。

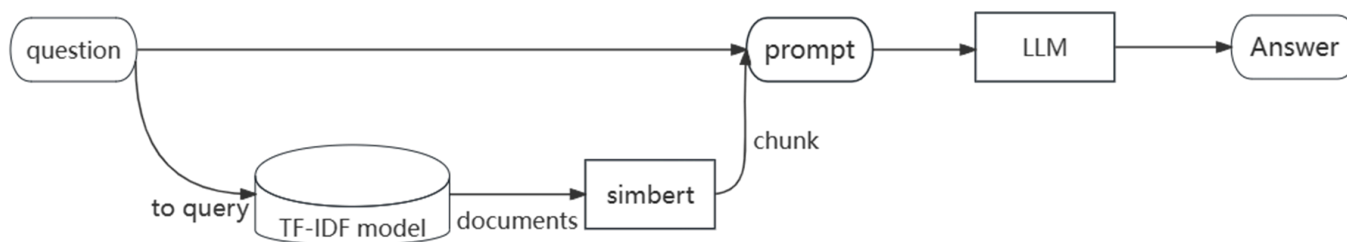
开发流程

问答系统首先需要一个基础的语言模型，由于要求中文模型，因此我们选取了目前比较先进的中文语言模型ChatGLM3。

一种能想到的方法是把所有的文档微调到LLM中，之后输入问题。但因为所有的文档一共有20MB，在我们的算力资源下，一般的模型并不能支持这么多的文本量输入。

因此我们借鉴前人的思路，通过预筛选的方式减少上下文的文本量，之后将筛选出来的文本作为LLM的上下文，就能对用户的问题进行问答。

为了筛选出最相关的文档段落，我们开创性地提出了二步的筛选方法，先选出和问题最为相关的若干篇文章，之后从这些文章中进一步筛选出相关性最高的若干个段落。



我们首先使用TF-IDF方法评估问题与各个文章的相似度。

TF-IDF (Term Frequency-Inverse Document Frequency) 是一种用于信息检索与文本挖掘的常用加权技术。这种方法用于评估一个字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

TF-IDF方法是一种统计方法，用以评估字词对于一个文件集中的其中一份文件的重要性。TF 表示词条（关键字）在文档中出现的频率。这个数字是对词条权重评估的一种直观感受。但是，由于每个文档的长度不同，它可能会对词条的重要性产生影响。因此，词频（TF）通常会被归一化（一般是词条在文档中的出现次数除以文档中的总词条数目）。IDF 的主要思想是如果包含词条 T 的文档越少，也就是说 T 越能够代表这个文档。逆文档频率是一个词条重要性增长的度量。计算某一特定词条的 IDF，可以将语料库中的文档总数除以包含该词条之文档的数目，然后将得到的商取对数得到。

通过将题目和每篇文章转换成TF-IDF权重向量，计算题目向量与每篇文章向量之间的余弦相似度，可以选择最相关的文章。

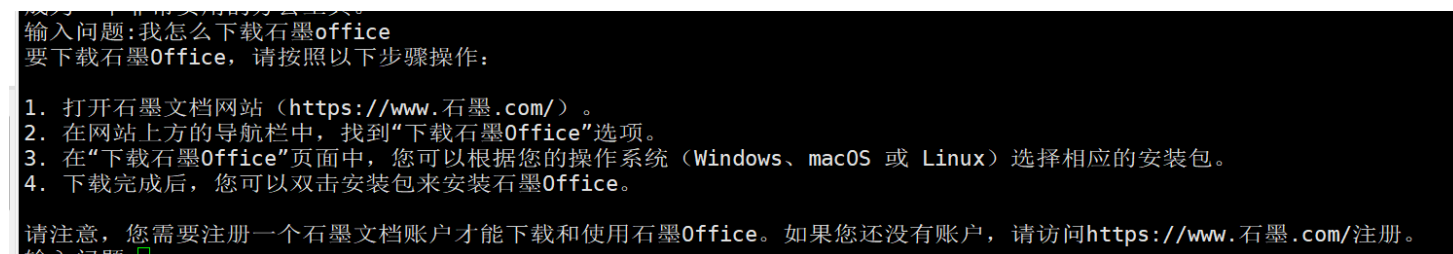
之后，我们再利用Simbert模型，找出与问题最为相关的n个段落。

为了提高效率，我们进行预处理，对于每篇文章依次分析，将各个段落映射成向量，保存到本地文件中。之后当我们向问答系统输入问题流时，我们可以直接对前一步找到的文章的段落向量与问题进行相似度计算，找出这些文章各自最相关的段落。

可以想象，假如不进行预处理，每次输入问题都要对所有文章的段落进行相似度计算，这将是非常耗时的。

最后，我们将这些段落与他们所在的链接一起作为上下文，结合原来的问题输入给预训练中文语言模型，得到问题的答案。

结果展示



输入问题:我怎么下载石墨office
要下载石墨Office，请按照以下步骤操作：

1. 打开石墨文档网站（<https://www.石墨.com/>）。
2. 在网站上方的导航栏中，找到“下载石墨Office”选项。
3. 在“下载石墨Office”页面中，您可以根据您的操作系统（Windows、macOS 或 Linux）选择相应的安装包。
4. 下载完成后，您可以双击安装包来安装石墨Office。

请注意，您需要注册一个石墨文档账户才能下载和使用石墨Office。如果您还没有账户，请访问<https://www.石墨.com/>注册。

通过采用ChatGLM3中文语言模型和预筛选技术，我们的机器人能够有效处理用户的问题，并从Deepin Wiki中找到最相关的内容作为回答。在测试阶段，我们的系统展示了80%以上的问题与答案的相关性，成功实现了项目的初步目标。此外，我们还实现了对玲珑使用手册内容的支持，并为机器人添加了可视化界面，使其能够在Deepin系统上运行。

总结

核心贡献

通过结合创新性的预筛选技术和中文大语言模型，我们的系统能够快速准确地理解用户的问题，并找到最相关的文档段落作为回答。

我们的二次筛选方法相比只筛选文章大大减少了模型的输入文本量，提高了交互效率。与此同时，我们相比之前一些项目提高了用户的使用效率，如无需要求用户给出自己问题的领域方向，允许用户提出涉及多个文档集合中子领域的问题。同时可以直观看到，二次筛选相比从所有文章段落组成的段落集合中筛选和提出的问题相关性最高的段落会大大提高效率。

在部署模型的过程中，我们实现了文档从文件集合到.json格式化过程，以及存储为数据库文件全过程的自动化，在预处理过程中我们实现了用文件存储simbert模型生成的词向量的自动化流程，还解决了simbert模型与chatglm模型所需transformer版本的不兼容问题，使得两者能够顺利结合。

同时，本项目不仅支持Deepin Wiki内容，还可扩展至其他文档资源，如玲珑使用手册，展现了良好的系统扩展性。

展望

未来的工作将集中在以下几个方面：

优化文档预筛选算法：进一步提高问题与文档匹配的准确性和效率。

增强模型的上下文理解能力：通过模型微调和优化，提高机器人在处理复杂问题时的准确性和可靠性。

提高系统的可扩展性和通用性：使系统能够更容易地适应不同领域的文档资源和用户需求。

通过持续的优化和改进，我们相信Deepin Wiki文档问答机器人将为用户提供更加丰富、准确和自然的信息检索体验。