利用检索增强生成技术开发本地知识库应用

朱俊仪[1]，朱尚明[2]
（1.上海市大数据中心，上海 200435；2.华东政法大学信息化办公室，上海 201620）

摘　要：检索增强生成（RAG）技术通过引入外部文档，让大模型具有访问外部知识库的能力，从而生成更真实可靠的回答，有效解决数据过时、语料不足问题。在介绍了大模型的基本架构和微调技术基础上，探讨了利用检索增强生成技术搭建本地知识库系统的应用框架，该应用框架由加载本地文档、文档拆分、拆分片段向量化、根据提问匹配文本、构造提示词和生成回答六部分构成。最后基于ERNIE-4.0大模型和AppBuilder开发平台，设计开发了一个面向校园信息服务的智能问答系统，并给出了具体实现。

关键词：大语言模型；检索增强生成；提示词；本地知识库；智能问答系统
中图分类号：TP391.1
文献标志码：A
DOI：10.11959/j.issn.1000−436x.2024227

Development of local knowledge base application using retrieval augmented generation technology

ZHU Junyi[1], ZHU Shangming[2]
1. Shanghai Muniplitical Data Center, Shanghai 200435, China
2. Informatization Office, East China University of Political Science and Law, Shanghai 201620, China

Abstract: Retrieval Augmented Generation (RAG) technology can enable large language models to access external knowledge bases by introducing external documents, thereby large language models can generate more authentic and reliable answers, and effectively solve the problems of outdated data and insufficient corpus. On the basis of introducing the basic architecture and fine-tuning techniques of large language models, the application framework of using retrieval enhanced generation technology to build a local knowledge base system was discussed. The application framework consisted of six parts: loading local documents, splitting documents, embedding splitting fragments, matching text based on questions, constructing prompts, and generating responses. Finally, based on the ERNIE-4.0 model and the AppBuilder development platform, an intelligent question answering system for campus information services was designed and developed, and a specific implementation was provided.

Keywords: large language model, retrieval augmented generation, prompt, local knowledge base, intelligent question answering system

0　引言

近年来，人工智能（AI）技术得到了飞速发展，尤其是ChatGPT的推出[1]，人工智能进入以大语言模型（LLM, large language model），简称大模型，为主的2.0时代，大模型已逐渐为人们所熟知，基于大模型的应用研究也成为人们日益关注的一个热点[2-4]。虽然国内外涌现出了很多通用大模型和行业大模型，但这些模型都存在一定程度的数据过时、语料不足

收稿日期：2024−10−20
通信作者：朱尚明，zhusm@ecupl.edu.cn
基金项目：中国高校产学研创新基金资助项目（No.2022MU061）
Foundation Item：The Industry-University-Research Innovation Fund of Chinese Universities (No.2022MU061)

等问题[5-6]，而检索增强生成（RAG, retrieval augmented generation）技术通过引入外部文档（如单位内部数据、个人隐私数据、小众数据等），让大模型能够访问外部知识库，从而生成更真实可靠的回答，能有效解决数据过时、语料不足问题[7]。因此，如何利用现有的大模型和检索增强生成技术，快速构建一个高可用的本地知识库应用系统，已成为一个AI领域值得深入研究的课题。本文以构建一个面向校园信息服务的智能问答系统为例，探讨利用大模型检索增强生成技术开发本地知识库的应用实践。

1　大模型发展概况

大模型是一种具有大规模参数和复杂计算结构的机器学习（ML, machine learning）模型，是一种使用海量数据和利用巨大算力训练而成的深度学习（DL, deep learning）神经网络，能够处理复杂的任务和数据，通过巨大的数据和参数实现生成式人工智能（Generative AI）。大模型和各种AI技术之间的关系如图1所示。

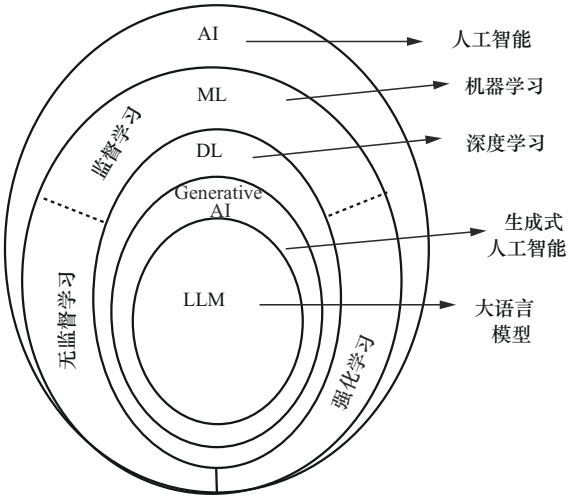

图1　大模型和各种AI技术之间的关系

大模型以大数据、大参数、大算力为典型特征，从使用方式上可分为开源大模型和闭源大模型，从部署方式上可分为在线大模型和本地大模型，从应用场景上又可分为通用大模型和行业大模型。2019年以来，大模型技术得到了飞速发展，各种模型层出不穷，国内外多家机构也都加大了对大模型的研发投入。表1列举了当前国内外常见的大模型，并对其模型参数量、最大序列长度和使用方式进行了对比。

表1　常见大模型对比

模型	发布机构	模型参数量	最大序列长度	使用方式
GPT-3	OpenAI	1750亿	2048	API
GPT-4	OpenAI	未知	32000	API
J1-Jumbo	AI21 Labs	未知	32000	受限访问
J1-Grande	AI21 Labs	1780亿	2048	受限访问
CodeGen	Salesforce	160亿	2048	开源
OPT	Meta	1750亿	2048	开源
LLaMA	Meta	650亿	2048	开源
T5	Google	110亿	512	开源
PaLM	Google	5400亿	2048	API
GLM-130B	清华大学、智谱	1300亿	2048	开源
MOSS	复旦大学	160亿	2048	开源
ERNIE 3.0 Titan	百度	2600亿	512	受限访问
源1.0	浪潮	2450亿	2048	受限访问
Baichuan	百川智能	70亿	4096	开源

2　大模型的架构和微调技术

2.1　Transformer架构

现在的大模型多数都是基于Transformer架构的，Transformer是一种使用自注意力机制的深度学习神经网络[8]，具有优秀的规模化能力和并行化计算能力，现已演变成自然语言处理的基础模型，广泛用于机器学习和人工智能。Transformer模型架构主要由输入部分、编码器、解码器和输出部分构成，其内部结构如图2所示。

2.1.1　输入（Input）部分

输入部分主要完成输入文本的Token化、向量嵌入和位置编码等功能。

1）Token化。token是文本处理的基本单元，token可能是一个中文词语（或英文单词）或者一个汉字，也可能是半个或三分之一个汉字。输入的文本被Token化，拆分成各个token，并用token ID表示。

2）向量嵌入（Input Embedding）。token ID通过嵌入层转化为向量表示，向量能包含更多的语法语义信息，并能反映词语之间的复杂关系。以GPT-3模型为例[9]，其向量长度是12288。

3）位置编码（Positional Encoding）。把词在文本中的顺序向量和词向量进行相加，结果传给编码器。

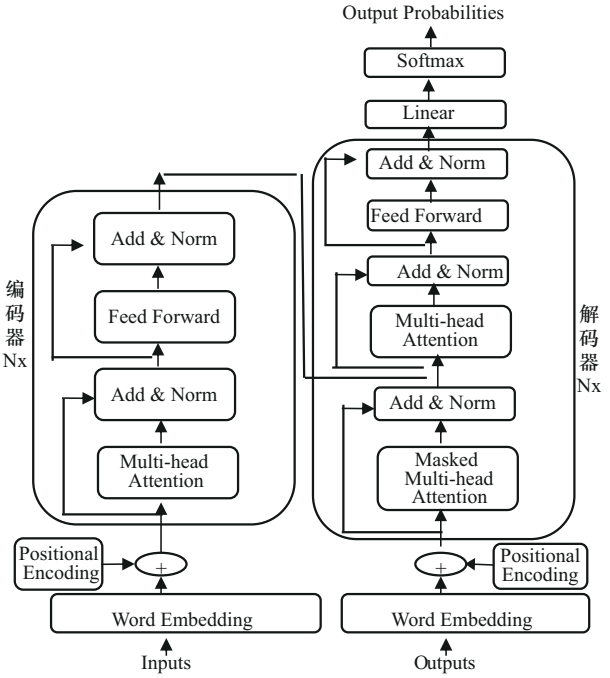

图2　Transformer内部结构

2.1.2　编码器(Encoder)部分

编码器在Transformer里由多个(Nx)堆叠在一起，其内部结构一样，但不共享权重，以便更深入地理解数据和处理更复杂的内容。

1) 把输入转化为更抽象的表示，也是向量表示，保留了词汇信息和顺序关系，并补充了语法语义的关键特征：自注意力机制。

2) 自注意力机制。关注词本身、附近的词，还关注输入序列中所有其他词。通过计算每对词之间的相关性来决定注意力权重。两个词之间的相关性强，则其注意力权重会高。编码器融合了词本身和上下文关系。

3) Transformer使用了多头自注意力。编码器有多个自注意力模块，每个头都有自己的注意力权重，用于关注文本里的不同特征。自注意力模块之间可以做并行计算。

4) 前馈神经网络对自注意力模块的输出进行进一步处理，增强模型的表达能力。

2.1.3　解码器(Decoder)部分

编码器输出的抽象表示会传给解码器，解码器在Transformer里也由多个(Nx)堆叠在一起。

1) 解码器会先接收一个特殊值，作为输出序列的开头，把已经生成的文本也作为输入。和编码器一样，也要经过文本嵌入层和位置编码，然后被输入多头自注意力层。

2) 针对已生成输出序列的带掩码的多头自注意力。解码器的多头自注意力层和编码器不一样，只关注词本身和它前面的词，不去关注后面的词，以遵循正确的时间次序。

3) 针对输入序列抽象表示的多头自注意力。注意力模块会捕获来自编码器的输出和解码器即将生成的输出之间的关系，将原始输入序列的信息融合到输出序列的生成过程中。

4) 解码器里的前馈神经网络和编码器类似。

2.1.4　输出(Output)部分

输出部分包含一个线性层和Softmax层，把解码器的输出转换为词汇表的概率分布。

1) 词汇表的概率分布代表下一个被生成token的概率，一般模型会选择概率最高的token作为下一个输出。

2) 解码器的输出流程会重复多次，新的token会持续生成，直到生成一个表示序列结束的特殊token。

2.2　基于Transformer的大模型类别

基于Transformer的大模型主要有以下三类。

1) 自回归语言模型。自回归语言模型仅使用解码器，用于文本生成。这类模型的文本生成能力强，如GPT-1、GPT-2、GPT-3、GPT-4等。

2) 掩码语言模型。掩码语言模型仅使用编码器（自编码模型），可用于情感分析。这类模型文本理解能力强，如BERT、RoBERT、DeBERTa等。

3) 编码器-解码器模型。编码器-解码器模型主要用于翻译、总结等场景，如T5、BART等模型。

2.3　大模型的微调技术

大模型的预训练过程主要包括无监督学习、监督微调和强化学习训练三步。具体如下。

Step1　无监督学习。通过海量的文本，进行无监督学习预训练，得到具有文本生成能力的基座模型。

Step2　监督微调。通过人工撰写的高质量对话数据，对基座模型进行监督微调，得到微调后的基座模型。

Step3　训练奖励模型+强化学习训练。通过问题和对应多个回答的数据，人工对回答进行质量排序，得到奖励模型，然后再用奖励模型对微调后的基座模型的问题回答进行评分，利用评分作为反馈

进行强化学习训练，得到最终对话模型。

微调是目前大模型采用的主流调优方法。目前大模型种类繁多，训练方法也很多，且具有模型越大、效果越好的特点，这就造成模型的微调越来越困难。常用的大模型微调机制主要有提示词学习和少量微调 2 种。其中，提示词学习能增强大模型的小样本学习能力，而少量微调则是一种用少量参数来优化大模型的方法。

3　基于 RAG 的本地知识库应用构建和开发

3.1　RAG 应用架构

信息检索、文本生成和机器问答，是大模型在自然语言处理方面 3 个代表性应用。但大模型在预训练和微调时，都会存在一定程度的数据过时、语料不足问题，而 RAG 技术通过引入外部文档（如 word、excel、PDF 等格式的文档），可以构建基于大模型的本地知识库应用，让大模型具有访问本地知识库的能力，从而生成更真实可靠的回答，有效地解决数据过时、语料不足等问题。RAG 是一种先进的自然语言处理方法，它结合了信息检索和文本生成技术，用于提高问答系统、聊天机器人等应用的性能。

基于大模型的 RAG 应用架构如图 3 所示，该应用架构由加载本地文档、文档拆分、拆分片段向量化、根据提问匹配文本、构造提示词和生成回答六部分构成。

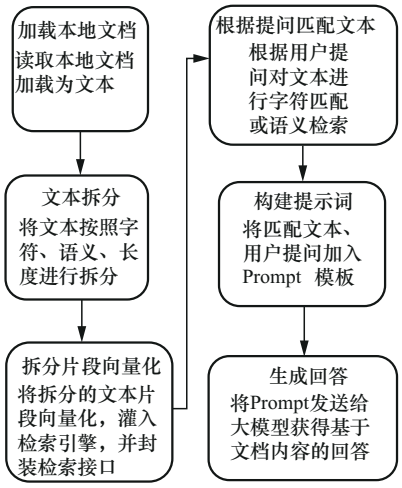

图 3　大模型的 RAG 应用架构

检索增强生成技术利用垂直领域数据扩充了大模型的能力，搭建一个完整的 RAG 系统的过程如下。首先，加载文档，并按一定条件切割成片段；然后，将切割的文档片段向量化，灌入检索引擎，并封装检索接口；最后，构建调用过程。其中，调用过程又可以分解为首先输入用户提问进行检索，然后构建生成提示词 Prompt，最后送入大模型得到回复回答。整个在线过程可以理解为获得用户问题，用户问题向量化，检索向量数据库，将检索结果和用户问题填入 Prompt 模板，用最终获得的 Prompt 调用 LLM 大模型，由 LLM 大模型生成回复。

3.2　基于本地知识库的应用构建及开发实现

从大模型的部署方式上，基于本地知识库的应用构建可以分为本地化部署实现和利用在线大模型来实现 2 种。本地化部署实现大多以 Langchain 为应用开发框架，受限于算力的约束，本文采用百度 ERNIE-4.0-8K 在线大模型来进行本地知识库的应用构建，并以 AppBuilder 为开发平台[10]，来设计开发一套面向校园信息服务的智能问答系统。

3.2.1　AppBuilder 开发流程

百度智能云千帆 AppBuilder 平台提供了 AppBuilder-SDK，AppBuilder-SDK 提供了完整的 AI 原生应用开发套件，包括丰富的开发组件和应用示例。开发组件包括大模型组件、AI 能力组件、基础云组件和软硬件一体组件，满足各类高灵活度定制开发需求；应用示例提供了丰富灵活的应用框架最佳实践，基于业内主流大模型应用框架搭建，包含如支持知识增强的应用框架、文本生成应用框架、具备思维链及工具使用能力的框架、生成式数据分析框架等。AppBuilder-SDK 目前提供了 Python 语言的 SDK，支持 Python 3.9 及以上版本。

AppBuilder 的开发流程为：创建应用—>发布应用—>获取应用 Token—>调用应用。AppBuilder 提供了 RAG、Agent、GBI 等应用框架，具有文档问答、表格问答、对话、创作等应用组件，以及文生图、语音等传统 AI 组件，降低了 AI 原生应用的开发门槛，赋能开发者快速实现应用搭建。

3.2.2　构建本地知识库

AppBuilder 构建本地知识库的过程实际上是创建应用和应用发布的过程。

1)创建应用

AppBuilder 支持用户通过应用配置界面完成应用设定、能力扩展等设定，并对应用进行在线测

试。通过输入指令、开场白和推荐提问，选择组件、知识库，设定模型配置、追问配置和知识库检索方式，即可完成应用的创建。具体包括如下几个步骤。

步骤1　配置应用的基本信息。基本信息包含应用的名称、描述等信息。

步骤2　配置角色指令。确定所创造应用的角色和任务，在角色指令中描述期望角色完成的任务和目标，指定回答的输出格式、结果内容、风格要求或字数限制等。

步骤3　上传知识库。大模型将基于上传的知识文档回答问题，可选择直接上传文件，或选择已有知识集合。上传或选择知识库后，还可以调整知识库检索的策略和参数，知识库检索包括全文检索和高级检索两种策略。上传文件支持docx、xlsx、jsonl、png、pdf格式的文件。

2)应用发布

完成应用配置和效果调试后，可以进行应用发布，实现多渠道发布并支持创建API调用密钥。

方式**1** 多渠道发布。具体发布渠道包括网页版、智能体平台、微信客服和微信公众号等。

方式**2** API调用。直接通过应用API调用接口，以API形式进行调用。

方式**3** 代码态开发。可参考AppBuilder-SDK使用说明进行代码态开发。AppBuilder-SDK提供了完整的AI原生应用开发套件，包括丰富的开发组件和应用示例代码。

3.2.3　校园智能问答系统开发与实现

校园信息服务领域有大量的文档类信息，人工对这些信息进行摘要和提取又是一件耗时耗力的事，而大模型对这类信息的处理得心应手，因此校园信息服务领域非常适合结合RAG的垂直类LLM应用部署。

AppBuilder提供Python支持，可首先通过pip install AppBuilder-SDK命令安装其Python包。在百度智能云千帆平台获取账号所对应的AppBuilder_Token和App_ID后，即可创建AppBuilder实例并进行调用。其核心调用代码如下。

```
# 配置密钥与应用ID
os.environ["APPBUILDER_TOKEN"] ="xxxx"
app_id = "xxxx"
# 初始化Agent实例
builder = appbuilder.AppBuilderClient(app_id)
# 创建会话ID
conversation_id = builder.create_conversation()
msg = builder.run(conversation_id, input, )
return(msg.content.answer)
```

结合Python的Flask Web框架，最后可实现基于RAG技术的、面向校园信息服务的智能问答系统网页应用。部署后相关工作人员通过网页端即可实现文档摘要、自定义智能咨询等服务。其系统实现结果如图4所示。

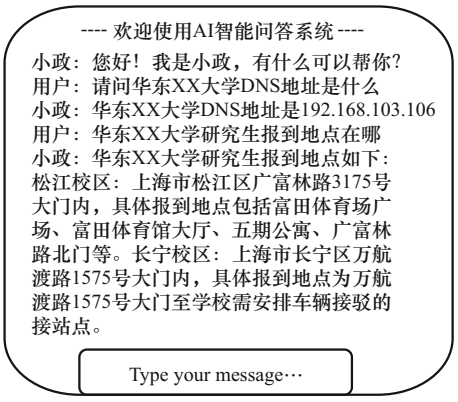

图4　系统实现结果

4　结束语

本文以校园信息服务为场景，基于百度ERNIE-4.0在线大模型和AppBuilder开发平台，探讨了利用检索增强生成技术实现本地知识库搭建和智能问答系统的开发，将来可以根据其他应用场景开发更多基于私有知识库的智能问答系统，例如“一网通办”智能问答、课程知识点智能问答、基于知识图谱的智能问答、根据对话历史的智能问答等。

参考文献：

[1] OpenAI .Introducing ChatGPT[EB/OL]. (2022) [2022-11-30].

[2] 舒文韬, 李睿满, 孙天祥, 等. 大型语言模型: 原理、实现与发展[J]. 计算机研究与发展, 2024, 61(2): 351-361.
SHU W T, LI R X, SUN T X, et al. Large language models: principles, implementation, and progress[J]. Journal of Computer Research and Development, 2024, 61(2): 351-361.

[3] 陶建华, 聂帅, 车飞虎. 语言大模型的演进与启示[J]. 中国科学基金, 2023, 37(5): 767-775.
TAO J H, NIE S, CHE F H. The evolution and inspiration of large language model technology[J]. Bulletin of National Natural Science Foundation of China, 2023, 37(5): 767-775.

[4] 徐月梅, 胡玲, 赵佳艺, 等. 大语言模型与多语言智能的研究进展与启

示[J]. 计算机应用, 2023, 43(S2): 1-8.
XU Y M, HU L, ZHAO J Y, et al. Research progress and enlightenment of large language models on multi-lingual intelligence[J]. Journal of Computer Applications, 2023, 43(S2): 1-8.

[5] 罗文, 王厚峰. 大语言模型评测综述[J]. 中文信息学报, 2024, 38(1): 1-23.
LUO W, WANG H F. Evaluating large language models: a survey of research progress[J]. Journal of Chinese Information Processing, 2024, 38 (1): 1-23.

[6] 赵月, 何锦雯, 朱申辰, 等. 大语言模型安全现状与挑战[J]. 计算机科学, 2024, 51(1): 68-71.
ZHAO Y, HE J W, ZHU S C, et al. Security of large language models: current status and challenges[J]. Computer Science, 2024, 51(1): 68-71.

[7] OMRANI P, HOSSEINI A, HOOSHANFAR K, et al. Hybrid retrieval-augmented generation approach for LLMs query response enhancement [C]//Proceedings of the 2024 10th International Conference on Web Research (ICWR). Piscataway: IEEE Press, 2024: 22-26.

[8] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 30th Annual Conference on Neural Information Processing Systems. New York: Curran Associates, 2017: 5990-6008.

[9] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877-1901.

[10] 百度公司. 百度云千帆 AppBuilder 产品简介 [EB/OL]. (2024) [2024-04-16].

[作者简介]

朱俊仪（1996– ），男，上海人， 上海市大数据中心助理工程师，主要研究方向为人工智能、大数据分析。

朱尚明（1969– ），男，河南虞城人，博士，华东政法大学研究员，主要研究方向为计算机应用、人工智能和网络安全等。