E-Commerce Letters 电子商务评论, 2025, 14(5), 1346-1359
Published Online May 2025 in Hans. https://www.hanspub.org/journal/ecl
https://doi.org/10.12677/ecl.2025.1451414

Hans汉斯

基于RAG本地电商知识库的DeepSeek电商模型构建与优化研究

冯　源，钱松荣*，陆宇亮

贵州大学公共大数据国家重点实验室，贵州　贵阳

收稿日期：2025年3月29日；录用日期：2025年4月16日；发布日期：2025年5月21日

摘　要

本文面向电商业客户对话及文案写作需求，提出基于RAG增强型检索技术的电商本地知识库与轻量化本地部署大语言模型结合，建立一种电商大语言模型智能助理。采用了Nomic-embed模型和数据处理方法，建立本地电商知识库的向量数据，通过设置采样和检索参数，结合DeepSeek-R1大语言模型实现对话与文案内容的推理。通过Dify框架和Ollama参数的试验，提出了基于大语言模型参数的模型部署调优方法。新搭建的本地电商大语言模型助理可有效解决在线大语言模型工具响应速度慢、信息传输不安全及推理内容不特定、不精确等问题，可为电商业提升工作效率提供新的途径。

关键词

电商文案，智慧电商，RAG检索技术，大语言模型

Research on Construction and Optimization of DeepSeek E-Commerce Model Based on RAG Local E-Commerce Knowledge Base

Yuan Feng, Songrong Qian*, Yuliang Lu

State Key Laboratory of Public Big Data, Guizhou University, Guiyang Guizhou

Received: Mar. 29th, 2025; accepted: Apr. 16th, 2025; published: May 21st, 2025

Abstract

This paper addresses the needs of e-commerce businesses in customer dialogue and copywriting by proposing an intelligent e-commerce assistant that integrates a RAG-enhanced retrieval technology-

*通讯作者。

文章引用: 冯源, 钱松荣, 陆宇亮. 基于 RAG 本地电商知识库的 DeepSeek 电商模型构建与优化研究[J]. 电子商务评论, 2025, 14(5): 1346-1359. DOI: 10.12677/ecl.2025.1451414

based local knowledge base with a lightweight, locally deployed large language model. The Nomic-embed model and data processing methods are employed to construct vector data for the local e-commerce knowledge base. By configuring sampling and retrieval parameters and combining them with the DeepSeek-R1 large language model, the system enables reasoning for dialogue and content generation. Through experiments with the Dify framework and Ollama parameters, a method for optimizing model deployment based on large language model parameters is proposed. The newly developed local e-commerce large language model assistant effectively addresses issues such as slow response times, insecure data transmission, and non-specific or imprecise reasoning content commonly found in online large language model tools. This approach offers a new pathway for improving operational efficiency in the e-commerce industry.

Keywords

E-Commerce Copywriting, Smart E-Commerce, RAG Retrieval Technology, Large Language Model

1. 引言

随着电子商务市场的持续扩张与消费者需求的多元化演进，平台运营正面临双重挑战：一方面，海量商品供给带来的选择过载显著提升了用户决策成本，个性化服务需求激增；另一方面，电商企业在客户服务响应、精准营销和内容创作等环节遭遇效率瓶颈。在电商行业快速发展的背景下，客户咨询响应效率与营销文案质量已成为影响企业竞争力的关键因素。这种效率与质量的双重压力，催生了对智能化解决方案的迫切需求。目前，许多行业开始应用基于大语言模型的智能助理系统，用于提升行业的工作效率和服务水平，但是，以在线大模型系统为基础的智能助理仍然存在一些问题，如网络高峰期在线响应缓慢，可能出现网络超时而难以有效回应用户；网络传输任务内容可能会出现商业机密安全问题；网络通用大模型未对特定电商知识优化，问答内容不能很好匹配电商特定需求。因此，建立本地专业知识库和知识图谱成为可能的解决思路。随着大语言模型技术的发展，文献[1]-[3]研究了建立专业知识库和知识图谱用于专业助理，进一步证明了技术路线的可行性。

2025 年 1 月，国产开源生成式大语言模型 DeepSeek-R1 发布，拥有 6710 亿的参数数量[4]。其在大语言模型推理方面的技术突破性进展显著，与传统依赖监督微调的模型不同，该模型通过强化学习实现自我演化，在无人工标注数据的情况下，模型能够自主优化推理路径并发展出优先级排序、复杂问题分步求解等高级认知能力。基于此，本文本地知识库与轻量化本地部署 DeepSeek-R1 模型相结合，探讨借助人工智能与商业服务领域的深度融合，为构建智能化电子商务工作助理提供了新的技术路径。

2. RAG 技术的电商知识库

在实现基于电商知识库的大模型智能助理之前，需要利用 RAG 技术建立相应的专业领域知识库，将对应的知识内容转化为语言向量，本研究将从 RAG 技术介绍、数据的模型选择和处理、知识库建立等方面阐述这部分内容。

2.1. RAG 技术的关键流程

检索增强生成(Retrieval Augmented Generation, RAG)是一种先进的自然语言处理技术，旨在通过结合信息检索系统和大规模语言模型(Large Language Model, LLM)的能力，提升文本生成的质量和准确性。具

体而言，RAG 框架首先利用高效的搜索算法从外部数据源中检索与用户查询相关的文档或信息片段，随后将这些检索到的上下文信息作为附加输入，整合到语言模型的提示(prompt)中。通过这种方式，RAG 不仅能够利用语言模型强大的生成能力，还能借助外部知识库的动态信息，增强模型对特定问题的理解和回答能力。这种技术特别适用于需要实时或领域特定知识的任务，例如开放域问答、知识密集型对话系统以及复杂的信息提取场景。RAG 的核心优势在于其能够将检索到的上下文与生成过程无缝结合，从而在保持生成文本流畅性的同时，显著提高其信息准确性和相关性。

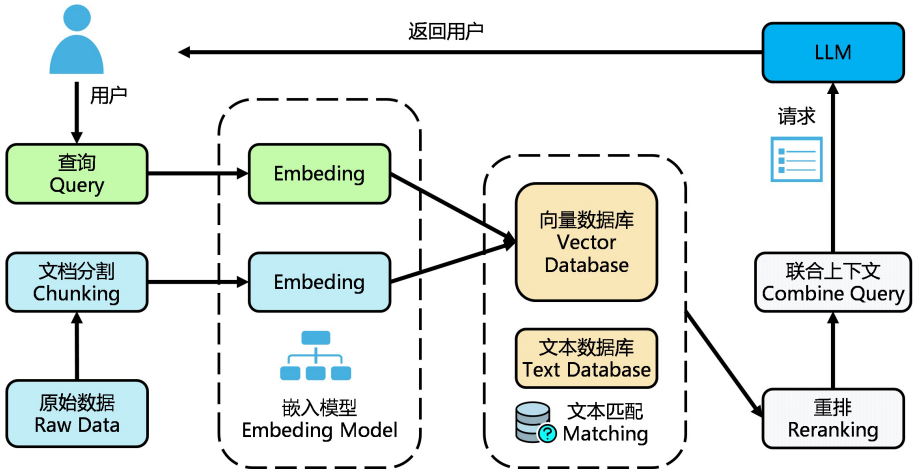


Figure 1. Text processing mode of RAG-enhanced retrieval technology
图 1. RAG 增强检索技术文本处理模式

在使用 RAG 技术处理文本时，一般需要经历数据清洗与格式化、向量化与嵌入、建立向量数据库、查询处理与检索优化、重排、联合查询与返回用户答案等步骤，图 1 为 RAG 文本推理过程。

- 数据清洗与格式化(Chunking)：清洗原始数据(如 HTML 标签去除、PDF 文本提取)，将原始数据按照一定的代码块大小进行分割，确保检索的粒度保持相同。
- 向量化与嵌入(Embedding)：利用嵌入模型对分割后的文本向量表示，提升匹配精度。
- 建立向量数据库(Vector Database)：用向量数据库存储向量数据。
- 查询处理与检索优化(Query Processing & Retrieval)：用户输入查询后，嵌入模型对用户输入进行重写，构建查询向量，这使用户输入的查询构建为更合适检索的向量格式，提交用户的查询到向量库中检索，找到最相关的文档内容。
- 重排(Reranking)：对检索到的文档内容进行重排，转换为大模型的输入文本，文档重排可使文档逻辑更准确，这一步可提高大模型文本查询的准确度。
- 联合查询(Combine Query)：将重排的文档内容和用户查询联合，作为大语言模型的输入，形成大语言模型丰富版本的 Prompt。
- 返回用户答案(Response)：大语言模型对输入的 Prompt 进行推理，输出用户需求的答案。

至 2024 年，基于大语言模型(LLM)的系统架构中，检索增强生成已成为某些行业实践的核心范式。其通过动态整合外部知识检索与生成能力，显著提升了模型的事实准确性、可解释性及适应性，被广泛应用于从通用问答服务到垂直领域智能应用的构建中。例如，微软的 CoRAG 框架[5]支持迭代查询重构，允许模型根据中间推理状态优化检索策略，显著提升多跳问答任务的性能，而 RAG-Gym 框架[6]则通过过程监督奖励模型，将搜索代理的决策过程形式化为嵌套马尔可夫决策过程(MDP)，实现了生成行为与高质量检索的深度协同。

2.2. 嵌入式模型选择与系统架构设计

RAG 管道处理知识库文本的重要工作之一是选择适当的嵌入式模型(Embedding Model)，其决定了知识库文本向量化的特征，图 2 为几种嵌入式模型的精度比较，其中 Nomic-embed-text 在较小参数的情况下，拥有比较高的精度。

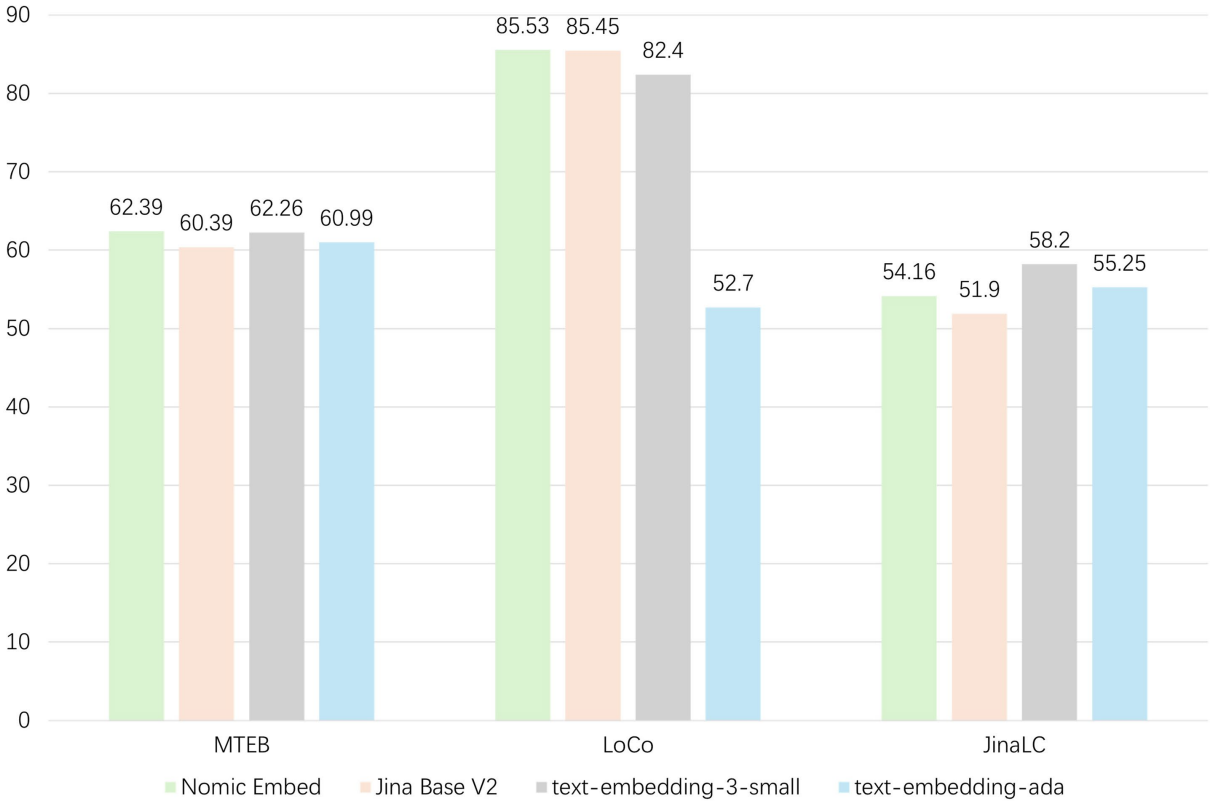


Figure 2. Accuracy comparison of Nomic-Embed-text Model, Jina Base V2 Model, text-embedding-3-small Model, and text-embedding-ada Model [7]
图 2. Nomic-Embed-text 模型、Jina Base V2 模型、text-embedding-3-small 模型、text-embedding-ada 模型精度比较[7]

目前，比较流行的嵌入式模型包括 OpenAI 的 Ada-002、ext-embedding-3-small 和 Nomic-embed-text 等。由于本研究为本地部署模型，考虑到个人电商用户的计算资源有限，应该优先选择轻量化和高效率的嵌入式模型。Nomic-embed-text 模型是一个基于 Sentence Transformers 库的句子嵌入模型，作为新型嵌入式模型，其在处理短文和长文本任务方面都超越了多数现有模型。Nomic-embed-text 模型拥有 137M 个参数，在多数模型中，属于较为轻量化的模型。同时，在 Ollama 和 Huggingface 等大模型网站上具有较高下载量，本研究选用 Nomic-embed-text 模型作为电商知识库文本向量化处理的嵌入式模型。

根据本地部署知识库的构建流程，如图 3，使用 Nomic-embed-text 模型的电商知识库大语言模型助手的基本架构应分为三个主要模块，即电商知识库向量模块、大语言模型查询模块和人机交互模块。

电商知识库向量模块主要由知识库文本和嵌入式 Nomic-embed-text 模型组成。该模块是系统的知识中枢，负责将电商领域的结构化与非结构化数据(如商品描述、用户评价、售后政策)转化为机器可理解的语义表征。通过预训练的嵌入模型(如 BGE、Cohere Multilingual)对文本、图像属性(如商品图标签)进行向量化，并构建分层索引(如 HNSW + IVF-PQ)，支持多模态混合检索。其核心作用在于将分散的商品知识、促销规则等动态信息编码为高密度向量空间，实现毫秒级语义匹配，例如精准关联“夏季连衣裙”查询

与库存中具有“透气”“雪纺”标签的商品详情，同时支持实时索引更新以适应价格变动或新品上架。

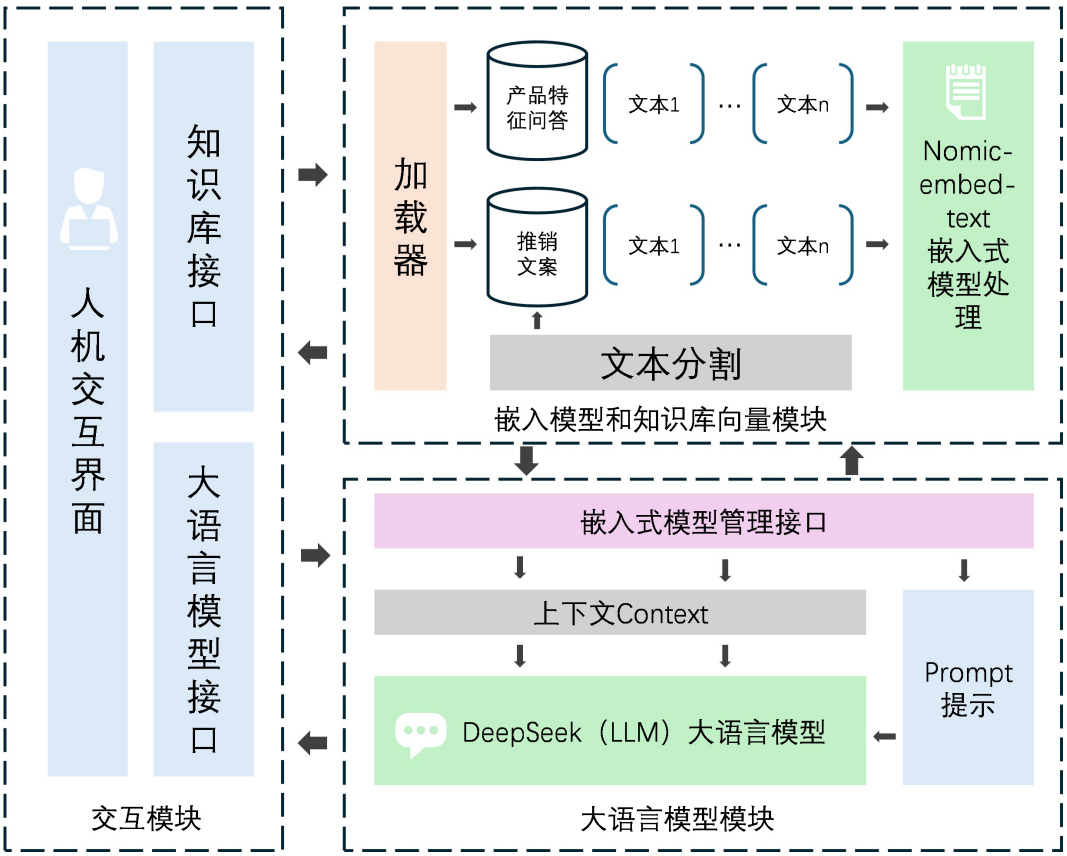


Figure 3. The basic framework of the E-commerce Knowledge Base Large Language Model Assistant consists of three components: the local vector-only module, the large language model module, and the interaction module
图 3. 电商知识库大语言模型助理的基本框架包括 3 个部分：本地知识向量模块、大语言模型模块和交互模块

大语言模型查询模块是处理用户查询的核心模块，本研究采用 DeepSeek-R1 模型作为大语言模型引擎。作为问答推理引擎，该模块深度融合检索结果与生成能力。首先基于用户查询从向量库召回相关上下文(如商品参数、退换货条款)，再通过指令微调的大语言模型(如优化的 DeepSeek-R1 模型)进行多轮意图解析与答案合成。关键技术包括动态上下文压缩(过滤无关促销信息)、事实一致性校验(如对比商品页 SPU 编号防止幻觉)，以及领域自适应生成(输出结构化答案如比价表格、促销倒计时)。例如，当用户询问“双十一如何叠加优惠券”，模型将自动关联会员等级规则、活动页条款，生成分步骤的个性化操作指南。

人机交互模块主要是提供人机交互界面，通过界面，用户可实现电商知识库的建立，参数的微调及查询。该模块承担自然语言理解、多模态输入输出及体验优化职能。同时内置容错机制：当检测到模糊查询(如“那个红色的包”)时，触发澄清追问(“您指的是 2024 新款马鞍包还是托特包？”)，并基于用户行为数据(询问特点及用户习惯)优化交互流程。

2.3. 向量化电商知识库的建立

建立知识库需要对知识库的内容进行收集，设计知识库包括个人电商中的哪些领域，如服装领域电商、电子领域电商等，服务内容包括客户问答及产品展示页文案等。接着需要对文本进行数据清洗，然

后采用上述 Nomic-embed-text 模型对文本内容选择适当参数向量化处理。

2.3.1. 知识库数据收集

本研究的电商数据包括客服对话数据及产品介绍数据。为研究不同电商场景，研究选择服装电商和旅游电商两个不同领域的电商内容，其中，对话数据选自互联网服装领域的用户和客服的对话数据，产品介绍数据涵盖某旅游电商旅游景区的景点数据，用于旅游产品的营销介绍，旅游景区的介绍数据从相关景区网站的公开资料收集。

2.3.2. 数据向量化原理

Nomic-embed-text 模型采用掩码语言模型的轻监督训练，在训练过程中，每次从一个数据源中单个抽样，整批次采用单一数据源样本，这个训练过程可以让模型避免对个别数据源的过拟合训练(减避免重复最短向量路径，减少过拟合)。而弱监督对比预训练旨在教一个模型来区分最相似的文档和其他不相关的文档。如公式(1)，其选用了 InfoNCE 对比损失函数作为损失函数模型。

$$\mathcal{L}_C = -\frac{1}{n}\sum_i \log \frac{e^{\frac{s(q_i,d_i)}{\tau}}}{e^{\frac{s(q_i,d_i)}{\tau}} + \sum_{j\neq i}^{n} e^{\frac{s(q_i,d_j)}{\tau}}} \tag{1}$$

其中，

n 是样本的数量；

q_i 是查询样本的编码向量；

d_i 是文档样本的编码向量；

τ 是温度系数，用于调节相似度得分的分布，后面会详细讨论；

$s(q_i,d_i)$ 是查询 q_i 与文档 d_i 之间余弦相似度。

nomic-embed-text 与其他模型如 text-embedding-ada-002 和 jina-embeddings-v2-base-en 的比较。nomic-embed-text-v1 在 MTEB 上的表现超过了 text-embedding-ada-002 和 jina-embeddings-v2-base-en。它在长上下文基准测试(LoCo 和 Jina 长上下文基准测试)上一致优于 jina-embeddings-v2-base-en，展现出对长序列更优越的性能。

2.3.3. 知识库数据处理

在构建大语言模型知识库向量时，由于模型本身的上下文窗口限制及检索效率需求，科学的内容分段处理成为关键环节。当前主流大语言模型虽然部分支持完整文档上传，但实验数据显示直接处理未分割的长文本会导致检索效能显著下降，其核心原因在于：模型精准回答的能力本质上依赖于知识库对内容块的精准检索与召回效率，这如同在搭建数据库时为数据库建立检索提高数据读取率。本研究采用 Dify 作为搭建数据知识库的软件平台。Dify 支持 RAG 技术的数据集成处理和管理。

Dify 提供两种结构化分段策略：其一为“通用分段模式”，采用智能滑动窗口技术，通过动态重叠机制确保上下文连贯性，特别适用于结构平铺的文档类型(如对话内容、技术报告)，在保证语义完整性的同时实现内容块的精准切分；其二为“父子分段模式”，通过建立多层级索引结构(如将产品劫煞牌按照名称和内容构建树状体系)，既能在父级节点保留宏观语义框架，又能在子级节点存储细粒度信息，这种架构设计尤其适配合同文书、法典条例和产品介绍等具有明确层次结构的文档类型。两种模式均内置自适应算法，可根据文档特征自动优化分段颗粒度，使每个内容块既能承载完整语义单元，又避免信息冗余，最终实现检索环节的查准率与查全率平衡。

现介绍主要分段参数如下。

分段标识符：即按照如何的方式分段知识库文本，默认为“\n”，按文章段落分段，其表达式可按照正则表达式，用户可按照句子的内容自己规定。图 4 为不同分段参数时的分段效果。

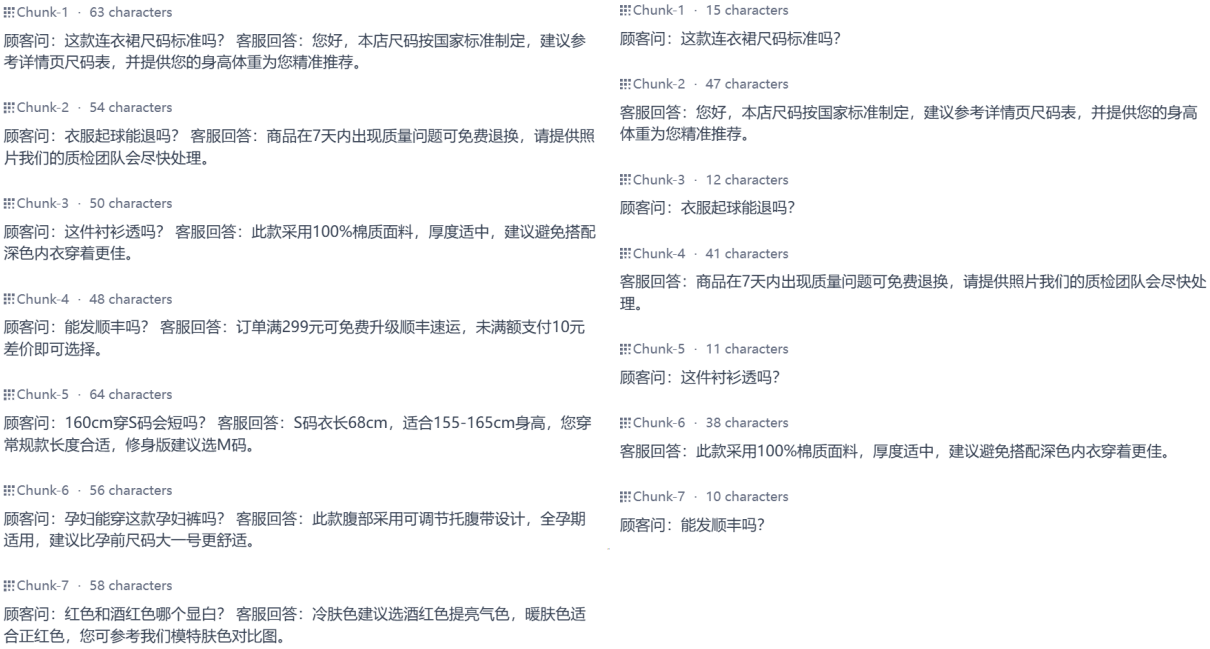


Figure 4. Segmentation effects of the knowledge base under different segmentation parameters
图 4. 对知识库采用不同分段参数时的分段效果

分段最大长度：这个参数调整分段内最大文本字符的数量，默认为 500 Tokens，超过这个数值时，系统将自动分段，分段时取分段标识符与最大分段长度较小者。由于不同机器的运算性能不同，个人机器部署时资源较少，而分段越大，处理时所消耗的资源就越多，因此，在较短文本时，可保持较小的默认分段长度，在较大文本时，建议按照公式(2)进行分段：

$$n = \frac{\sum_1^n T_i}{\mu} \tag{2}$$

其中，

n 为分段数量；

T_i 为文本库中第 i 个文本的容量，单位为字节；

μ 为分段大小，取决于机器的运算资源，单位为字节。

上述公式中，分段大小取决于机器的运算资源配置，建议在迷你主机级别部署时，取 $\mu = 2k$ 字节，在中等显卡如 4070 Ti 配置下，取 $\mu = 4k$ 字节，在较高资源配置，如 4090 Ti 配置下，取 $\mu = 6k$ 字节。然后调整分段最大长度，使分段数量接近 n 的数值。

分段重叠长度：这个数值为分段之间重叠的部分。重叠可以帮助在构建知识库向量时，提升准确性、响应召回。建议设置重叠长度在分段 Tokens 数量的 15%~25%。

文本预处理规则：文本预处理规则可以清理知识库内主要内容关联性较小的部分。

本研究中，旅游电商知识库分段采用“父子分段”模式，服装电商场景的对话分段则采用“通用分段”模式。在旅游电商场景知识库中，分段大小采用 1024 Tokens,，分段数量为 11 段，分段情况如图 5，分段成果如表 1。

Figure 5. Data segmentation in the tourism e-commerce knowledge base
图 5. 旅游电商知识库的数据分段情况

Figure 6. Data segmentation in the apparel e-commerce knowledge base
图 6. 服装电商知识库的数据分段情况

服装电商场景中，采用最大分段 1024 Tokens，重叠率按照 15%设置，由于文本为对话内容，分段数量较多，分段情况如图 6，分段成果如表 2。

Table 1. Segmentation results table of the travel e-commerce knowledge base
表 1. 旅游电商知识库分段成果表

分段模式	父子分段
最大分段长度	父：1024；子：200
文本预处理规则	替换掉连续的空格、换行符和制表符
索引方式	高质量
索引设置	向量检索

Table 2. Segmentation results table of the apparel e-commerce knowledge base
表 2. 服装电商知识库分段成果表

分段模式	通用
最大分段长度	1024；分段重叠：150
文本预处理规则	替换掉连续的空格、换行符和制表符
索引方式	高质量
索引设置	向量检索

经过上述步骤，建立了较为高效的电商场景的本地知识文本库。

3. 轻量化大语言模型本地部署与调优

为保证知识库由较高的表达和检索水准，需要选择和部署相适应的大语言模型。本研究采用 DeepSeek-R1 模型为大语言模型。

3.1. 大语言模型量化的选择。

为保持较低成本和轻量化部署，本研究分别从两组不同的低成本配置角度考虑选择模型，分别对应低成本独立显卡主机与轻量化核心显卡迷你主机。其中一组配置为利用 2080 Ti 22GB 显存版本显卡、6 核心 i5 处理器、16 GB 内存、Nvme 通道 SSD 硬盘；一组配置为具有 AMD780M 核显、锐龙 R7 处理器、64GB 内存、Nvme 通道 SSD 硬盘，系统为 Windows11，22H2 版本。在轻量化主机上部署大语言模型时，需综合考虑模型参数量、量化精度、硬件资源占用及生成速度的平衡。以下从资源占用、响应速率、量化精度三个核心维度，分析 DeepSeek-R1 系列中 7b-Q4、14b-Q4、32b-Q4/Q6 模型的适配性，并提供具体选型建议。

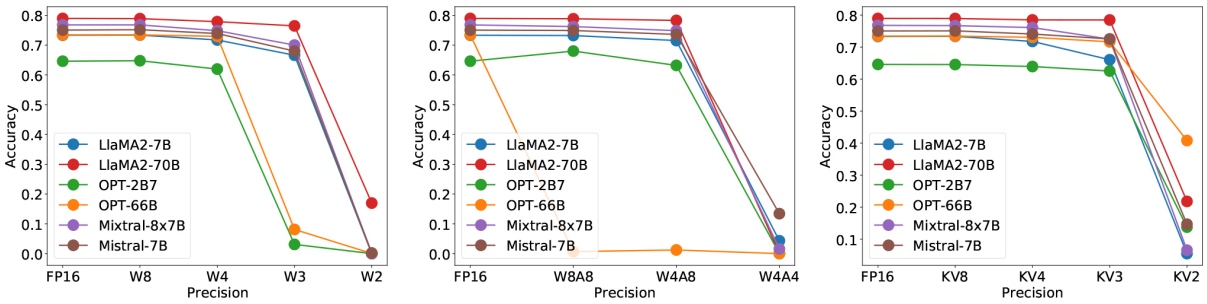
显存需求：7B-Q4 模型：经 4-bit 量化后显存占用约 4.2GB (原模型 13 GB)，核显需共享系统内存(需 16GB 以上物理内存)；14B-Q4 模型：显存需求约 8.5 GB，核显主机需 32 GB 内存以支持共享显存分配；32B-Q4 模型：在独立显卡上面，显存需求约 20 GB，核显主机显存占用约 36 GB，Q6 量化核心显卡显存占用约 40 GB，在 32B-Q6 量化时，模型无法在不占用外部显存的情况下，部署在 22 GB 显存的独立显卡上，本研究在核心显卡部署 32B-Q6 量化模型时，系统总内存占用约 57 GB，不同系统占用资源略有不同。由于不同 Modelfile 文件配置时，模型占用资源情况会有不同，因此，表 3 数据为默认 Modelfile 文件配置下占用资源。

Table 3. Comparison table of resource usage by models with different quantization
表 3. 采用不同量化的模型所占用的资源对照表

模型名称	量化水平	独显显存占用	核显内存占用	核显系统内存占用
DeepSeek-R1:7b	Q4	4.2 GB	6 GB	15 GB
DeepSeek-R1:14b	Q4	8.5 GB	15 GB	26 GB
DeepSeek-R1:32b	Q4	20 GB	36 GB	49 GB
DeepSeek-R1:32b	Q6	大于 22 GB	44 GB	57 GB

量化精度：量化精度影响任务类型与误差容忍度和模型输出的准确性与逻辑连贯性，需根据任务需求选择。4-bit (Q4)量化优势：显存占用最小，生成速度最快；Q4 劣势：精度损失约 8%~10%，复杂推理(如代码生成)易出错；适用场景：简单问答、文本摘要、基础翻译 58。6-bit (Q6)量化

优势：精度损失降至 3%~5%，逻辑推理能力更接近原模型；Q6 劣势：显存与计算需求增加，核显主机需更高内存带宽。根据图 7 研究，Q4 量化精度最接近模型量化的效能边界，其与 Q6 精度的差别较小，但占用资源较低，低于 Q4 精度的量化模型，精度损失较大。



(a) W-only Quant. (b) WA Quant. (c) KV Cache Quant.

Figure 7. The effect of quantization on different tensor types on LAMBADA (Natural language understanding task) [8]
图 7. 不同量化情况的模型在 LAMBADA 的表现(自然语言理解任务) [8]

根据上述内容，本研究选择 DeepSeek-R1:32b-Q4 量化模型作为电商大语言模型助理本地部署的模型。

3.2. 模型部署与调优

3.2.1. 模型的部署

目前，支持部署 DeepSeek-R1 模型的软件框架包括 LLMStdio 和 Ollama 等。为便于个人电商用户部署与调试模型，本研究采用部署较为方便的 Ollama 框架作为大模型部署基础。Ollama 具有官方网站，拥有较为完整的官方支持与 GitHub 社区资源。部署时，用户需要从 Ollama 官方网站下载 Ollama 软件，安装在计算机上。这里需要注意的是，由于 Ollama 官方软件对 AMD 核心显卡支持较少，在使用 AMD780M 核心显卡部署 DeepSeek-R1 时，用户需要从 GitHub 上找到“likelovewant/ollama-for-amd”项目，从“ollama-for-amd”项目中下载为 AMD 核显设计的 Ollama 软件，然后从 AMD 网站下载对应版本的 ROCm 软件驱动。

Ollama 软件安装完成之后，即可通过“ollama run deepseek-r1:32b”命令部署模型。

3.2.2. Modelfile 文件及参数调优

Modelfile 是用于定义和配置大语言模型的核心文件，通过编写该文件，用户可以根据指令自定义模型的加载方式、生成参数、系统提示等。其中的主要指令有“FROM”“PARAMETER”“SYSTEM”

“TEMPLATE”“ADAPTER”等，现就核心参数解释如表 4。

Table 4. Table of main configuration parameters in the Modelfile
表 4. Modelfile 文件的主要配置参数表

指令	描述
FROM	定义要使用的基础模型。
PARAMETER	设置 Ollama 如何运行模型的参数。
SYSTEM	指定将在模板中设置的系统消息。
TEMPLATE	发送到模型的完整提示模板。
ADAPTER	应用于模型的(Q)LoRA 适配器。

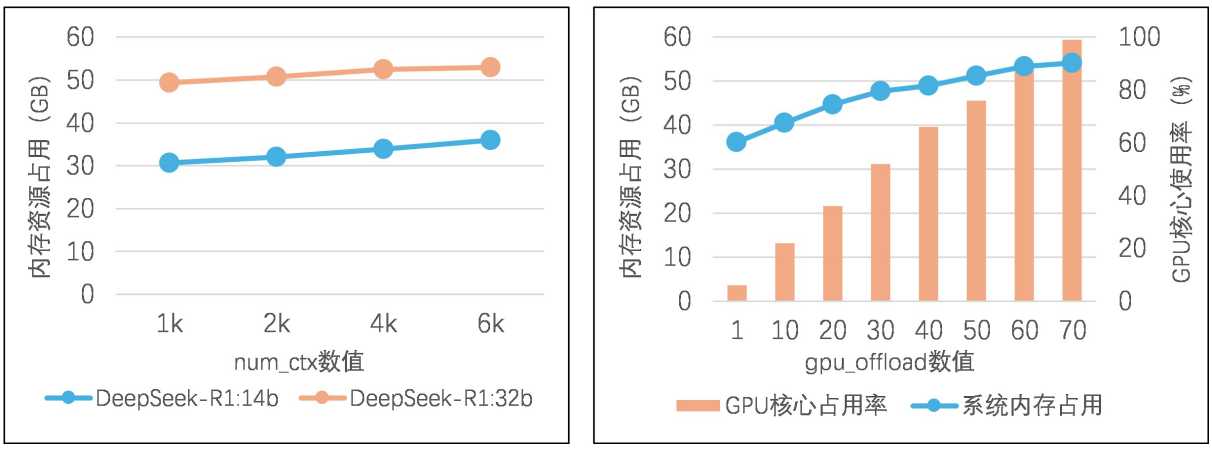
表 5 为“PARAMETER”中部分参数，在实际部署中，“PARAMETER”中的部分参数会直接影响模型的推理效率，因此我们需要关注和试验“PARAMETER”中的部分参数。

Table 5. Table of PARAMETER configuration parameters
表 5. PARAMETER 参数配置表

PARAMETER 参数	描述	值类型	数值案例
num_ctx	设置用于生成下一个标记的上下文窗口的大小。(默认值：2 K)	整型	1024
repeat_penalty	设置对重复项的惩罚强度。(默认值：1.1)	浮点	1.0
temperature	模型的温度。提高温度会使模型更有创意地回答。(默认值：0.8)	浮点	0.7
num_predict	生成文本时要预测的最大令牌数。(默认值：−1，无限生成)	整型	256
top_k	降低产生无意义的可能性。(默认值：40)	整型	40
top_p	这个值与“top_k”联动，对文本的多样化起到影响作用。(默认值：0.9)	浮点	0.07

现在就其中几个参数解释如下。

“num_ctx”参数会影响大语言模型同时理解的上下文范围。经过实验，“num_ctx”的数值会较为明



(a) (b)

Figure 8. (a) Resource usage (GB) across different num_ctx parameters; (b) Resource usage (GB) and GPU utilization (%) of the DeepSeek-R1:32b model under various gpu_offload parameters
图 8. (a) 不同 num_ctx 参数占用资源(GB)；(b) DeepSeek-R1:32b 模型在不同 gpu_offload 参数时占用资源(GB)和 GPU 使用率(%)

显的影响模型部署时的内存占用，尽管“num_ctx”的数值越大，模型能收集的上下文信息越多，但同时，会占用更多的内存及显存。因此，建议将其调整为适合的数值，以达到最优的模型推理效率。在问答型知识库时，模型往往需要查询用户之前的对话，因此，建议将其调整为 4 k~6 k 左右，而在生成文案时，则可调整为 2 k~4 k。图 8(a)为不同“num_ctx”对系统资源的占用。

“temperature”是非常重要的参数，其数值会直接影响大语言模型生成推理内容时候的宽容度。如果希望推理内容尽量贴合给定材料，较少自由创作，则应该使用较低的“temperature”数值；如果希望推理内容尽可能具有创造性，则应设置较高的“temperature”数值。一般来说，数值在 1 附近时，模型会有比较大的自由创作空间，而数值小于 0.4 时，模型会更贴合指定的材料。

“top_k”会对重复性内容造成影响，其决定了模型在选择某个字时，从 k 个机率最高的字中选择，从而避免某些低频率的字出现。较小的 k 值使输出更集中和确定，较大的 k 值则增加多样性和随机性。“top_k”与“top_p”联合使用，可以控制采样的范围。

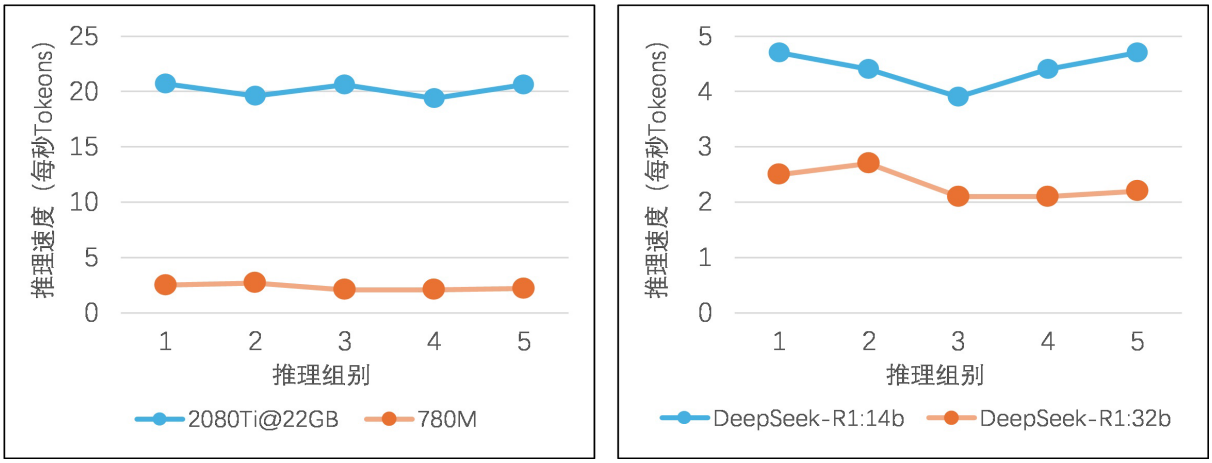
“gpu_offload”是一种显存优化策略，通过动态分配模型的组件到 GPU 和 CPU，实现减少单卡显存占用。在使用 Dify 框架部署模型时，可以通过调整这个参数实现较少的单卡显存占用率，需要注意的是，由于在独立显卡时，模型超过专用显存，加载到共享显存时，会降低模型推理速度，因此在使用独立显卡推理时，需要调整参数使模型全部加载到专用显存。根据试验不同“gpu_offload”时系统内存占用和 GPU 使用率如图 8(b)，随着“gpu_offload”值增加，系统内存占用和 GPU 使用率增加。gpu_offload 显存占用可通过公式(3)计算。

$$\text{GPU\_Memory}_{\text{used}} = \left[\frac{\text{Params}_{\text{total}} \cdot b_{\text{dtype}} + M_{\text{activation}} + M_{\text{buffer}}}{1024^3}\right] \cdot (1-\alpha) \tag{3}$$

其中，$Params_{total}$ 为模型总参数量，单位为参数个数，32B 模型为 3.2×10^{10}；b_{dtype} 为参数数据类型字节数，$FP32$ 为 4，$FP16$ 为 2；$M_{activation}$ 为前向和反向传播中的中间激活值内存，单位字节；M_{buffer} 为临时缓冲区内存，单位字节，α 为 GPU 卸载比例，$0 \le \alpha \le 1$。

3.2.3. 模型部署试验

本研究对 DeepSeek-R1:14b 和 DeepSeek-R1:32b 模型在不同配置机器部署和调优，其部署成果如图 9。



(a) (b)

Figure 9. (a) Inference speed of DeepSeek-R1:32b on 2080Ti@22GB vs. 780 M GPUs; (b) Inference speed comparison: DeepSeek-R1:14b vs. DeepSeek-R1:32b on 780 M GPU

图 9. (a) DeepSeek-R1:32b 在 2080Ti@22GB 显卡和 780 M 显卡上的推理速度；(b) DeepSeek-R1:14b 和 DeepSeek-R1:32b 在 780 M 显卡上的推理速度

独立显卡专用显存加载模型的情况时，模型推理速度显著快于核心显卡；模型连续推理时保持基本相同推理速度。

4. 总结与电商大语言模型行业影响展望

4.1. 本文研究成果总结

本文围绕本地电商知识库搭建电商大语言模型助手的设计和实现，探讨了在大语言模型时代，新兴AI 技术如何应用和改变电商行业；讨论了本地电商知识库的搭建及部署，对影响大语言模型的参数进行了试验和调优，主要有以下成果：

1) 探索和验证了基于 RAG 技术的轻量化主机部署本地电商知识库的技术路径。

2) 设计了搭建电商本地知识库的基本方法，包括数据的整理与知识库的搭建。

3) 通过试验，提出了本地部署电商大语言模型的优化方法。

4.2. 研究的局限性

尽管本研究提出了一些成果，但仍然存在以下局限性：

1) 本地电商知识库的实时性不足。智能体技术的进一步研究有助于通过自动化的方法收集某些需要人工采集的数据。

2) 多模态交互能力不足。目前的交互仍然基于语言交互，需要进一步扩展多模态交互能力，扩展电商大语言模型助理的应用场景。

4.3. 电商行业影响展望

随着 AI 大语言模型技术的发展，自然语言推理能力的成长正在改变许多工作[9]，通过 RAG 技术构建本地电商知识库并开发大语言模型助理，将显著提升电商行业的智能化水平，优化用户体验，提高运营效率，并推动行业生态的变革。未来，随着技术的不断进步，基于 RAG 技术的本地部署电商大语言模型助理与电商的深度融合将为行业带来更多创新机遇，同时也需要行业共同努力应对技术挑战，实现可持续发展[10]。虽然这个过程也面临数据质量、模型“幻觉”和计算资源等挑战，需要行业共同努力推动技术创新和标准化解决方案，但总体而言，RAG 技术的应用将推动电商业从“以商品为中心”向“以更高效率的用户为中心”转变，为行业带来更多智能化机遇和长期增长潜力。

参考文献

[1] 张元鸣, 姬琦, 徐雪松, 程振波, 肖刚. 基于知识图谱关系路径的多跳智能问答模型研究[J]. 电子学报, 2023, 51(11): 3092-3099.

[2] 孟令鑫, 才华, 付强, 易亚希, 刘广文, 张晨洁. 基于关系记忆与路径信息的多跳知识图谱问答算法[J]. 吉林大学学报(理学版), 2024, 62(6): 1391-1400.

[3] 段雨希, 邱芹军, 田苗, 等. 面向地质图的知识图谱构建及智能问答应用[J]. 地质科学, 2024, 59(2): 588-602.

[4] DeepSeek-AI, Guo, D.Y., *et al*. (2025) DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv: 2501.12948.

[5] Wang, Z.T., Yuan, H.T., Dong, W., Cong, G. and Li, F.F. (2024) CORAG: A Cost-Constrained Retrieval Optimization System for Retrieval-Augmented Generation. arXiv: 2411.00744.

[6] Xiong, G.Z., Jin, Q., *et al*. (2025) RAG-Gym: Optimizing Reasoning and Search Agents with Process Supervision. arXiv: 2502.13957.

[7] Nussbaum, Z., Morris, J.X., Duderstadt, B. and Mulyar, A. (2024) Nomic Embed: Training a Reproducible Long Context Text Embedder. arXiv: 2402.01613.

[8] Li, S.Y., Ning, X.F., Wang, L.N., *et al*. (2024) Evaluating Quantized Large Language Models. arXiv: 2402.18158.
[9] 李国杰. DeepSeek 引发的 AI 发展路径思考[J]. 科技导报, 2025, 43(3): 14-19.
[10] 李斯伦. 大语言模型视域下电商客服人才培养思考[J]. 市场周刊, 2024(36): 151-154.