

旅行代理 LLM：整合 LangChain、知识图谱和检索增强生成，实现酒店和景点推荐

石秋

乔治华盛顿大学

萨达尔·哈米迪安教授和阿明·梅赫拉比安教授

摘要—本项目介绍了一种会话式旅游助手，它将大语言模型（LLM）、LangChain、自定义知识图谱（KG）和检索增强生成（RAG）结合在一起。该助手将来自 Kaggle 酒店数据集的结构化数据与来自知识图谱的非结构化语义知识相结合，根据用户查询推荐酒店、景点和交通。评估结果表明，KG 和 RAG 集成后，非酒店查询有了明显改善，突出了它们在增强助手能力方面的作用。

关键词—**旅游** 助手、LLM、LangChain、知识图谱、检索增强生成、酒店推荐

1. 问题定义和理解

1.1. 问题定义

该项目旨在开发一款旅行助手，帮助用户找到通过自然语言查询酒店、景点和交通选择。系统：

- 将查询分类为酒店相关或非酒店相关。
- 在大型 CSV 酒店数据集上使用基于规则的酒店过滤功能询问。
- 在自定义知识图谱上为以下内容执行语义搜索非酒店查询。

- 通过 LLM 生成响应，整合检索到的数据和背景。
- 保持对话记忆，以便后续查询。

这可确保有效利用结构化和非结构化数据，以提高用户满意度和查询相关性。

1.2. 了解领域和数据

旅行决策涉及酒店特色、地点偏好等因素。ences，以及当地景点。**数据来源**

- **酒店数据集**：来自 Kaggle 的包含酒店元数据的大型 CSV 数据集（例如，*HotelName*、*HotelRating*、*HotelFacilities*）。用于筛选与酒店有关的查询。
- **知识图谱**：定制，编码关系，如 *HAS_ATTRACTION*、*HAS_TRANSPORT* 和 *HAS_HOTEL*。最初以纽约市为基础，可以扩展到其他目的地。

2.2. 特征选择和特征工程

该系统没有采用大量的功能工程设计，而是采用了

- 根据用户查询进行简单筛选（如城市、星级、设施）。
- 通过 KG 进行语义搜索的文本嵌入。

我们**没有**使用二进制特征（如 NEAR）来表示邻近性。取而代之的是相关性是在检索过程中通过语义相似性推断出来的。

3. 模型选择与开发

3.1. 选择适当的模式

该系统的功能来自于

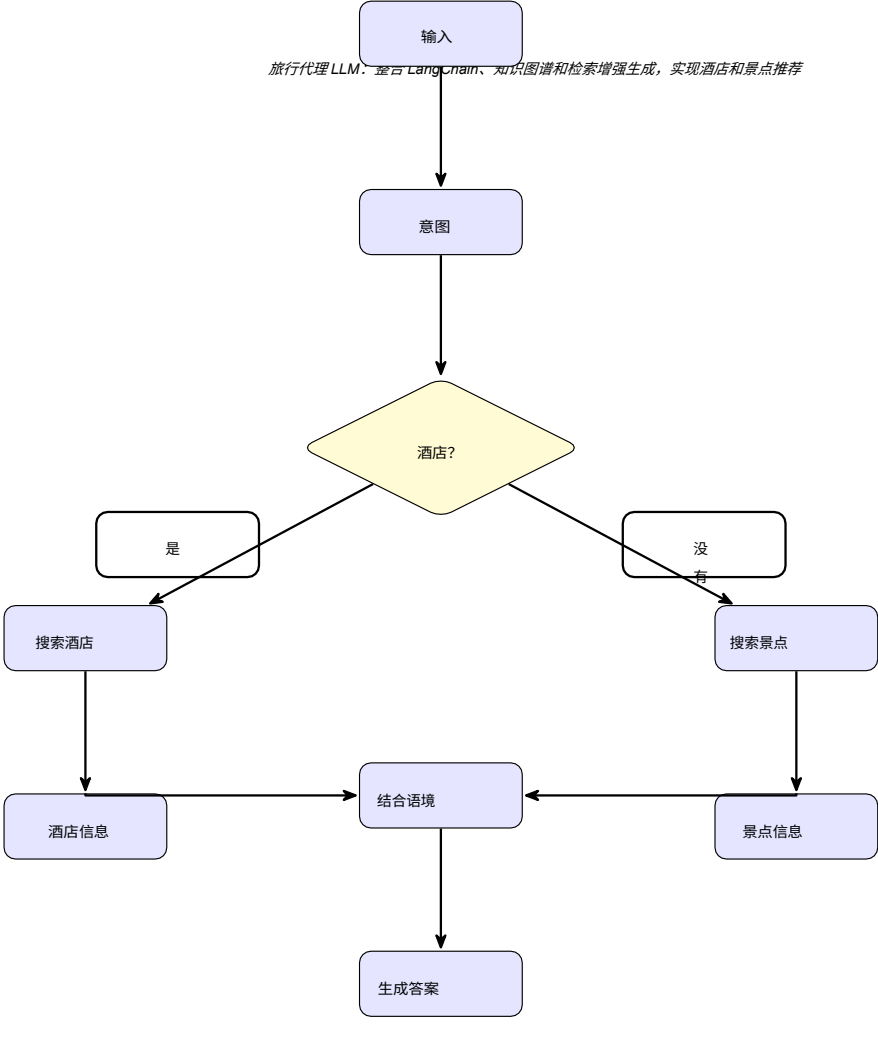
1. **基于 LLM 的意图分类**：将查询分类为酒店或使用 LangChain 的非酒店类别。
2. **酒店查询过滤**：在 CSV 上应用筛选器（如.....）、城市、）。
3. **基于向量的非酒店查询检索**：使用 OpenAI 嵌入和 FAISS，以便在 49 个网络地址上进行语义搜索。KG.

3.2. 正确执行程序

工作流程如图 1 所示，包括以下步骤：

1. **用户查询**：用户用自然语言提问。

通过整合结构化的酒店数据和景点与酒店的 KG 运输，系统可以智能地应对不同的旅行需求。查询。



法学硕士

LangChain
KG

RAG

图 1.旅行助理系统的综合工作流程

68 4. 评估和绩效指标

4.2.模型性能分析

70

69 4.1.分类指标比较

表 1.整合和未整合 RAG 系统的分类指标比较

公制	有 RAG	无 RAG 提供的酒
店	3,994	数量（精确 号码未提供）
4-星级酒店	490	众多
5-星级酒店	89	众多
正确的联系信息	是	不，不正确、
交通信息	是	化妆
响应成本	高	是
		中

表 2.模型性能统计（最近 7 天）

统计	数值
运行次数	267
代币总数	80,001 / \$1.02
代币中位数	221
误差率	2%
% 流媒体	0%
延迟	P50: 0.68 秒, P99: 10.13 秒

在过去的一周里，系统执行了 267 次运行，使用了 80 001 到 71
代币的成本为 1.02 美元。每次运行的令牌数量中位数为 221 个。它 72
错误率保持在 2%，没有出现任何流媒体问题，表明高 73
可靠性。50% 的查询延迟为 0.68 秒，10.13 74
秒，确保迅速做出反应。这些指标表明 75
通过持续监测优化高效绩效 76
异常值的延迟。 77

5. 结论

我们利用 LLMs 开发了一个系统，这是一个综合性的 78
酒店数据集和自定义知识图谱。通过采用意图 79
响应时间 中 低 80

分类、基于规则的酒店过滤和语义嵌入-- 81

81

基于检索的非酒店查询，该系统避免了复杂的查询过程。—————

82

83 特征工程。大幅整合知识图谱
84 提高了回复的相关性和用户满意度。
85 在非酒店查询方面的性能比使用了.....的系统高。
86 out KG integration。这凸显了结构化
87 在提供高质量旅游推荐时的语义上下文。

6. 参考资料

- LangChain 文档：<https://python.langchain.com/>
- OpenAI 嵌入：<https://platform.openai.com/docs/guides/embeddings>
- FAISS 矢量商店：<https://faiss.ai/>
- NetworkX：<https://networkx.org/>
- Kaggle 酒店数据集：<https://www.kaggle.com/datasets>
- 检索-增强生成：<https://arxiv.org/abs/2005.11401>
- 知识图谱：https://en.wikipedia.org/wiki/Knowledge_graph
- Bilibili 视频：<https://www.bilibili.com/video/BV1FtytYeEQM>
- 旅游景点预测论文：<http://bos.itdks.com/6b6277a4f30b41aeb7a56bd5524dbe49.pdf>
- 利用增强的位置知识预测旅游景点 -- 边缘图：<https://ieeexplore.ieee.org/document/10466836>
- TravelRAG：基于多层知识图谱的旅游景点检索框架：<https://www.mdpi.com/2220-9964/13/11/414#>
- 代理框架概述，作者 Lilian Weng：<https://lilianweng.github.io/posts/2023-06-23-agent/>
- LangChain RAG 教程：<https://python.langchain.com/docs/tutorials/rag/>