



本科生毕业设计（论文）

论文题目 大模型增强知识图谱的医疗问答研究与应用

作者姓名 刘泽广

专业 计算机科学与技术

指导教师 巢进波 讲师

2024 年 6 月

燕山大学本科生毕业设计（论文）

大模型增强知识图谱的医疗问答研究与应用

学 院：信息科学与工程学院
专 业：计算机科学与技术
姓 名：刘泽广
学 号：202011040104
指 导 教 师：巢进波
答 辩 日 期：2024 年 6 月

学位论文原创性声明

郑重声明：所呈交的学位论文《大模型增强知识图谱的医疗问答研究与应用》，是本人在导师的指导下，独立进行研究取得的成果。除文中已经注明引用的内容外，本论文不包括他人或集体已经发表或撰写过的作品成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果，并承诺因本声明而产生的法律结果由本人承担。

学位论文作者签名：刘泽广

日期：2024年6月3日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权燕山大学将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密□，在__年解密后适用本授权书。

本学位论文属于

不保密

(请在以上相应方框内打“√”)

学位论文作者签名：刘泽广

日期：2024年6月3日

指导教师签名：梁进波

日期：2024年6月3日

燕山大学毕业设计(论文)任务书

学院: 信息科学与工程学院

专业(方向): 计算机科学与技术

学号	202011040104	学生姓名	刘泽广	专业班级	计算机科学与技术 20-4 班
题目	题目名称	大模型增强知识图谱的医疗问答研究与应用			
	题目性质	1. 工理类: 工程设计 () ; 工程技术实验研究型 () ; 理论研究型 () ; 计算机软件型 (√) ; 综合型 () 。 2. 文管类 () ; 3. 外语类 () ; 4. 艺术类 () 。			
	题目类型	1. 毕业设计 (√) 2. 论文 ()			
	题目来源	科研课题 (√) 生产实际 () 自选题目 ()			
主要内容	调研大语言模型和知识图谱的国内外研究现状, 结合两者在知识表示和语义理解、自然语言生成方面的互补优势, 设计一种大语言模型增强知识图谱算法, 并基于该算法完成一个医疗问答系统的设计与实现。				
基本要求	1. 调研充分, 系统设计符合社会规范, 算法设计合理; 2. 编码规范, 可读性好, 用户界面友好; 3. 对系统进行充分测试, 测试数据丰富且符合实际意义, 达到成功、稳定运行的效果; 4. 论文内容逻辑结构清晰、采用软件工程工具表达设计思想, 符合燕山大学本课毕业设计论文的撰写规范; 5. 合理分配时间, 及时和老师沟通, 保证各环节顺利完成。				
参考资料	<p>[1] 张鹤译, 王鑫, 韩立帆等. 大语言模型融合知识图谱的问答系统研究 [J]. 计算机科学与探索, 2023, 17 (10) :2377-2388.</p> <p>[2] 王鑫雷, 李帅驰, 杨志豪等. 基于预训练语言模型的中文知识图谱问答系统 [J]. 山西大学学报(自然科学版):1-8[2023-12-15].</p> <p>[3] 韩毅, 乔林波, 李东升等. 知识增强型预训练语言模型综述 [J]. 计算机科学与探索:1-29[2023-12-15].</p> <p>[4] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2023). Unifying Large Language Models and Knowledge Graphs: A Roadmap. <i>ArXiv</i>, <i>abs/2306.08302</i>.</p>				
周次	12月12日—寒假	1—3周	4—9周	10—13周	14—15周
应完成的内容	查阅文献, 学习LLM和KG技术, 打通难关; 探索各种方案的可行性; 完成文献综述和外文翻译	需求分析 开题报告 总体设计 核心算法	编码实现 问答系统 中期答辩	系统测试 系统优化 撰写论文 论文查重	制作PPT 答辩
指导教师: 巢进波					
职称: 讲师 2023年12月16日					

摘要

问答系统是一种能理解自然语言问题并给出答案的计算机系统。随着互联网的普及，人们越来越习惯于在线查找信息。在医疗健康领域，医疗问答系统能够向公众提供准确、权威的医疗健康信息，帮助人们了解疾病预防、治疗和康复的相关知识，提高公众的健康素养，因此具有广泛的应用价值。基于知识图谱的问答系统能利用图谱结构提供更精准的答案，但面临知识更新慢、处理复杂问题能力有限等挑战。近年来大语言模型的兴起带来了强大的自然语言理解和生成能力，能够处理更复杂的问题，并通过上下文学习来更新知识。因此，利用大语言模型与知识图谱优势互补的特性，将两者结合起来构建医疗问答系统，有望为用户提供更智能、适应性更强的医药咨询服务。

本研究对医疗问答系统的功能及技术进行了调研与分析，确定了所用的技术、设计了系统的功能模块。随后按照良好的程序编码风格编写代码实现全部功能。首先，本文利用大模型结合知识图谱实现了问答功能。其次，实现了疾病名称提示功能，增加了用户问题在知识图谱中的查询命中率。随后，实现了知识图谱管理功能，利用大模型的自身知识结合用户反馈，对知识图谱进行补缺与修改。此外，系统围绕问答相关的需求，实现了历史记录、登陆注册、知识图谱可视化、用户反馈与审核、统计与分析等功能。实现了所有功能后，进行测试，验证系统的性能。

本系统使用 Pycharm 工具进行开发。后端使用 Flask 框架、Python 语言实现，前端采用 HTML+CSS+jQuery 实现，使用了 eCharts 绘制图标。使用 Neo4j 图数据库构建医疗知识图谱，使用了 ChatGLM、ChatGPT 等大语言模型。经测试，验证了问答功能的可用性和准确性，并对系统功能进行了测试，保证了系统运行的稳定性，具有一定的应用价值。

关键词：医疗问答；大语言模型；知识图谱；neo4j 图数据库

Abstract

Question-answering systems are computer systems capable of understanding natural language queries and providing answers. With the proliferation of the internet, people are increasingly accustomed to searching for information online. In the medical health field, medical question-answering systems can provide the public with accurate and authoritative medical health information, assisting individuals in understanding disease prevention, treatment, and rehabilitation knowledge, thereby enhancing public health literacy and offering broad application value. Question-answering systems based on knowledge graphs can utilize the graph structure to provide more precise answers but face challenges such as slow knowledge updates and limited ability to handle complex questions. The recent rise of large language models has brought about powerful natural language understanding and generation capabilities, capable of handling more complex issues and updating knowledge through contextual learning. Therefore, leveraging the complementary strengths of large language models and knowledge graphs to construct medical question-answering systems promises to offer users more intelligent and adaptable pharmaceutical consultation services.

This study investigates and analyzes the technology and functionality of medical question-answering systems, determining the technologies used and designing the system's functional modules. Subsequently, the code was written in a good programming style to implement all functionalities. Firstly, the paper utilizes large models combined with knowledge graphs to achieve the question-answering function. Secondly, it implements a disease name suggestion feature, increasing the hit rate of user queries in the knowledge graph. Then, it realizes knowledge graph management functionality, using the large model's inherent knowledge combined with user feedback to supplement and modify the knowledge graph. Additionally, the system implements features related to question-answering needs, such as history tracking, login registration, knowledge graph visualization, user feedback and review, and statistical analysis. After implementing all functionalities, the system was tested to verify its performance.

The system was developed using the Pycharm tool. The backend was implemented using the Flask framework and Python language, while the frontend was created with

Abstract

HTML+CSS+jQuery, utilizing eCharts for chart drawing. The medical knowledge graph was built using the Neo4j graph database, employing large language models such as ChatGLM and ChatGPT. Upon testing, the usability and accuracy of the question-answering function were verified, and the system's functionalities were tested to ensure the stability of its operation, demonstrating certain application value.

Keywords: Medical Q&A; Large Language Models; Knowledge Graphs; Neo4j Graph Database

目 录

目 录

摘 要	I
Abstract	II
目 录	V
第 1 章 绪 论	1
1.1 课题背景	1
1.2 目的和意义	2
1.3 国内外文献综述	3
1.4 主要研究内容	4
1.4.1 问答功能	4
1.4.2 疾病名称补全提示功能	4
1.4.3 知识图谱空缺补全	4
第 2 章 开发工具和技术	5
2.1 开发工具	5
2.1.1 PyCharm 介绍	5
2.1.2 Flask 框架介绍	5
2.1.3 Neo4j 图数据库介绍	5
2.2 开发技术	5
2.2.1 问答系统	6
2.2.2 知识图谱	6
2.2.3 大语言模型	6
2.2.4 词嵌入技术	7
2.3 本课题方案设计及对比	7
2.4 本章小结	9
第 3 章 系统分析	11
3.1 需求分析	11
3.1.1 功能需求	11
3.1.2 性能需求	12
3.2 可行性分析	13

3.3 系统开发环境	13
3.4 本章小结	13
第4章 系统设计	15
4.1 总体设计	15
4.1.1 总体模块设计	15
4.1.2 知识图谱结构设计	16
4.1.3 数据库设计	17
4.2 详细设计	19
4.2.1 登陆注册模块	19
4.2.2 问答模块	20
4.2.3 疾病名称提示模块	21
4.2.4 知识图谱可视化模块	22
4.2.5 对话历史记录模块	23
4.2.6 用户反馈模块	23
4.2.7 知识图谱管理模块	24
4.2.8 系统设置模块	25
4.2.9 统计与分析可视化模块	26
4.3 本章小结	26
第5章 系统实现	27
5.1 系统配置	27
5.2 系统主页	29
5.3 登陆注册模块的实现	30
5.4 问答模块的实现	30
5.5 疾病名称提示模块的实现	33
5.6 知识图谱可视化模块的实现	35
5.7 对话历史记录模块的实现	37
5.8 用户反馈模块的实现	39
5.9 知识图谱管理模块的实现	43
5.10 系统设置模块的实现	46
5.11 统计与分析可视化模块的实现	46

目 录

5.12 本章小结	48
第 6 章 系统测试	49
6.1 问答效果测试	49
6.1.1 正常生成回答测试	49
6.1.2 回答准确性测试	50
6.2 系统功能测试	53
6.3 本章小结	56
结 论	57
参考文献	59
致 谢	61
附录 1 开题报告	63
附录 2 文献综述	69
附录 3 中期报告	73
附录 4 外文原文	83
附录 5 外文翻译	97

第1章 绪 论

1.1 课题背景

医疗是民生中的重要一环。目前中国有 14 亿人口，而我国的医疗资源配置情况难以完美覆盖如此庞大的人口基数，所以“求医难，问病难”这种情况普遍存在^[1]。面对医院线下问诊的时间成本，轻微症状的患者或仅有咨询需求的人群若亦前往医院，将不可避免地增加医生的工作负担，并延长急需面诊患者的等待时间。随着网络的发展，人们开始利用网络问诊。用户利用搜索引擎，即可找到含有相关信息的网页。但是搜索引擎搜索出的信息较为繁杂，其中包含广告和不真实的信息，对于用户来说，从搜索引擎返回的网页信息中找到正确答案有一定的难度。近年来，像“寻医问药”、“春雨医生”这类的在线求医问药的网站兴起，使用户可以便捷地在网络上描述自己的病情并得到医生的指导，但是这种方式仍然需要人力的维护，并且医生通常推荐面诊来确定病情^[2]。因此，人们对于医疗问答系统有较大的需求。

Google 公司于 2012 年提出知识图谱概念，知识图谱存储结构化的数据，可用三元组表示知识图谱中存储的数据：(e1,s,e2)，其中 e1, e2 是实体，s 是它们之间的关系。知识图谱能够存储和组织领域专业知识，并通过知识推理和推断回答问题^[3]。可利用知识图谱作为知识库构建基于知识图谱的问答系统，一般来说，基于知识图谱的问答系统利用深度学习技术进行命名实体识别和问题模板匹配，以在知识图谱中检索并返回答案。然而，自然语言的多样性和模糊性使得准确理解用户意图成为一项挑战。此外，当用户提出的问题在知识图谱中无法找到匹配答案时，系统往往只能返回无效信息。

2017 年，Google 提出了一种基于自注意力机制的特征提取器 Transformer，其作为基础单元出现于几乎所有的大语言模型上^[4]。大语言模型的发展为自然语言处理任务，包括问答系统，提供了新的可能性。当前的大语言模型，如 GPT-3.5 和 GLM，在问答任务中展现出巨大潜力，它们利用在预训练期间学习到的知识来提供答案。然而，当前的大语言模型在答案准确性和答案幻觉方面仍存在局限。

因此可以使用大语言模型结合知识图谱实现问答系统。通过大语言模型强大的自然语言理解能力，系统能够从用户问题中提取意图，匹配查询语句模板，或直接生

成查询语句。结合知识图谱检索到的结果，大语言模型可以生成更准确的回答，减少答案幻觉的发生。在知识图谱存在空缺的情况下，大语言模型可以利用其自身知识，结合专家审核，来填补这些空缺。

1.2 目的和意义

为了缓解“求医难，问病难”的现象，将一部分轻症患者和仅有咨询需求的人群的问诊需求交给问答系统，可以减轻医院线下问诊的时间成本。此外，问答系统还可用于科普教育目的，为民众解答医疗疑问，加强民众的医疗常识。本课题还可探索大模型增强知识图谱问答的实现方法，对自然语言到 CQL (Cypher Query Language) 语句的转换 (Text-to-CQL) 任务的几种实现方式做了测试，并利用大模型根据给定内容生成回答。

本研究主要意义有以下几点。

(1)为用户提供一个医疗问答的系统。本研究构建的医疗问答系统，可以回答疾病的常见问题，满足一些用户对于医疗问答的需求，帮助用户解答问题。具有应用价值和社会意义。

(2)减少线下问诊的时间成本。本研究通过构建问答系统，能够减少患者在医院的问诊时间。这不仅提高了医疗服务的效率，也减轻了医生的工作压力。通过对智能问答系统的咨询，患者可以更快地获得必要的医疗信息，而无需前往医院，或提前了解相关知识从而更好地与医生交流。这对于轻微症状或仅有咨询需求的患者尤其有益。

(3)实现几种 Text-to-CQL 的方法，并对他们进行测试，选出最适合本研究的一种方法应用到问答系统。通过这些方法，系统能够理解用户的自然语言问题，并转换为精确的数据库查询语句，从而在知识图谱中检索到准确的答案。

(4)对用户输入的疾病名称做补全提示，这不仅提高了用户体验，也减少了输入错误的可能性。同时开发了美观、易于操作的界面，系统的设计处处体现方便用户的理念。

(5)尝试利用大模型+专家审核补充知识图谱。本研究探索了如何将大语言模型的内在知识和专家的审核结合，来填补知识图谱中的空缺。

1.3 国内外文献综述

随着计算机技术的发展,为了解决用户对于问答的需求,各类问答系统被提出,问答系统也在不断地发展。公认最早问答系统理论是二十世纪六十年代由艾伦·麦席森·图灵提出的著名的图灵测试,在之后数十年中,涌现出一批具有代表性的问答系统,如 Baseball 和 Lunar 等^[5]。2012 年来,随着知识图谱的出现,其作为一种强大的知识表示和组织方式,在包括问答领域的很多领域得到应用。同时深度学习的发展,使得自然语言处理变得相当便捷,提高了问答的灵活性^[5]。基于知识图谱的问答系统(knowledge base question answering, KBQA)由此诞生,其利用深度学习技术和知识图谱技术实现对用户问题的理解与回答。2017 年来,随着 Transformer 技术的出现,大语言模型得到发展,其在各种自然语言处理任务上表现卓越。部分研究者利用大语言模型结合知识图谱,实现问答系统。实现思路如下:首先,根据用户问题,分析问题中的实体名称和语义信息,得到知识图谱查询语言,在知识图谱中查询答案,再根据查询结果,结合用户问题,使用大模型生成回答。

首先要解决的是查询语句的获取。本研究使用了 Neo4j 数据库构建的知识图谱使用了名为 CQL 的查询语句。国防科技大学的 Aibo Guo 等人提出了 Text-to-CQL 任务^[6],即将自然语言文本转换为 CQL 语句的任务。实现该任务有几种方式:

(1)由于大模型在预训练时涉及到 CQL 语句的语料,对于较为简单,实体、关系种类不多的知识图谱,可将知识图谱的结构作为先验知识,在 prompt 中提供给大模型,大模型根据 prompt 中的内容,将输入的自然语言文本转换为输出的 CQL 语句。

(2)利用 few-shot prompts,通过在提示中加入案例来提供给模型进行学习,来引导模型生成所期望的结果。

(3)利用大模型微调技术,将自然语言-Cypher 的文本对作为微调数据,对大模型进行微调,微调后的模型可以完成 Text-to-CQL 任务。

其次要根据用户问题和查询结果,生成答案。Chen Wang 等人在研究中将知识图谱中的三元组数据和用户问题一起加入到大模型的 prompt 中^[7]。大模型根据这些信息,可以生成基于查询结果的答案。

总而言之,大模型结合知识图谱问答是近几年的新兴领域,但国内外的一些学者已经做了不少研究,本研究将在他们的基础上,选择合适的实现方式,设计并实现大模型结合知识图谱的问答系统。

1.4 主要研究内容

本研究的主要研究内容有以下三点。此外，需要开发历史记录、知识图谱查询结果可视化、登录注册、系统设置等功能，构建一个功能完备、对用户友好的问答系统。

1.4.1 问答功能

本功能主要结合大语言模型和知识图谱技术，实现问答功能。问答功能主要需要实现 Text-to-CQL 任务，随后使用 CQL 在知识图谱中查询结果，最后将查询结果与用户问题一起作为 prompt 提交给大模型，令其生成回答。该功能将大语言模型擅长的自然语言处理任务与知识图谱擅长的知识存储与表示任务相结合，优势互补，实现对医疗问题的问答。

1.4.2 疾病名称补全提示功能

该功能为了方便用户的输入并减少用户的错误输入。面对复杂的疾病名称，用户很难正确输入疾病的全称，此时系统可根据用户输入的疾病名称的几个汉字，做模糊查询，为用户提示可能的疾病全称。利用 Word2Vec 词向量技术，实现近义词查询。近义词查询结果结合对知识图谱的模糊查询的结果，在用户输入时为用户提供可能的疾病名称补全提示。

1.4.3 知识图谱空缺补全

当知识图谱中查询不到某一内容时，系统需要利用大模型生成回答。用户可通过反馈功能对当前回答进行反馈，反馈后，系统将会从回答中提取三元组，经专家审核后，补充到知识图谱中。以实现对知识图谱内容的完善。

第 2 章 开发工具和技术

2.1 开发工具

本系统使用 Python 语言进行开发，使用 PyCharm 编写代码，使用 Miniconda 创建 Python 虚拟环境，使用 Flask 框架开发 web 应用，使用 Neo4j 图数据库搭建知识图谱，使用 MySQL 数据库存储用户数据。

2.1.1 PyCharm 介绍

PyCharm 是一款由 JetBrains 公司开发的强大的集成开发环境（IDE），专门用于 Python 开发，它提供了丰富的功能和工具，帮助开发者提高开发效率、改善代码质量，并支持各种 Python 相关的技术和框架^[8]。本研究的开发基于该软件进行。

2.1.2 Flask 框架介绍

Flask 是一个使用 Python 编写的轻量级 Web 应用框架，它的显著特点是小，仅需几行代码便可以运行一个完整的 Web 应用程序^[9]。本研究的开发的问答系统建立于由 Flask 框架开发的 Web 应用之上。

2.1.3 Neo4j 图数据库介绍

Neo4j 是一个基于 JVM 的 NOSQL 数据库，它可以存储和查询数据以及数据之间的连接，其将数据信息以节点和边的形式储存在图中，并通过连接关系来表示节点之间的关联^[10]。对于高度连接的数据，Neo4j 的处理速度比关系型数据库快很多，使其成为管理跨金融、社交、电信到地理空间等多个领域的复杂数据的理想选择。

2.2 开发技术

本课题主要用到的技术有：问答系统、知识图谱技术、大语言模型、分词技术、词嵌入技术等。针对界面的开发，用到了 jQuery、layui、eCharts 等技术。下面对本研究用到的主要技术进行介绍。

2.2.1 问答系统

问答系统（Question Answering System, QAS）是自然语言处理领域的一个重要分支，它的目标是理解并回答人类提出的问题。问答系统的主要特点是其能够理解自然语言问题，并从大量的知识源中找到最相关的答案。这些知识源可以是结构化的数据库，也可以是非结构化的文本数据。问答系统的发展历程可以追溯到早期的基于规则的系统，如 ELIZA 和 BASEBALL，这些系统主要依赖于预定义的模式匹配和人工编写的规则^[10]。随着深度学习的发展，问答系统开始采用神经网络模型，如 Seq2Seq 模型和注意力机制，这些模型能够在大规模数据集上进行训练，从而提高了问答系统的性能。

2.2.2 知识图谱

知识图谱（Knowledge Graph, KG）是一种结构化的知识表示方法，它以图的形式表示实体及其之间的关系，从而能够对复杂的现实世界进行建模。其可以建立在图数据库之上，提供了一种新的知识储存和利用的途径，旨在解决具有复杂关系的数据存储和查询问题^[11]。知识图谱可以存储结构化的三元组，即知识实体以及实体之间的关系，基于它们构建一个知识网络。进而可以在该网络上进行查询、知识推理等工作。知识图谱可作为问答系统的知识库，为问答系统提供知识支撑。

2.2.3 大语言模型

大语言模型（Large Language Model, LLM）是自然语言处理领域的里程碑，它是一种能够通过处理大量文本数据来理解和生成人类语言的 AI 系统，它展现了强大的自然语言理解和解决复杂任务的能力，许多传统的自然语言处理任务在大语言模型的帮助下正在变得更加简化，且取得了更好的效果^[12]。它的两个主要特点是其使用的 Transformer 架构的自注意力机制和巨大的参数量^[13]，大语言模型在自然语言任务中展现了强大的能力。使用大模型的方式主要有两种：对于 GPT3.5、GLM4 等参数量大、闭源的模型，可以通过调用各大厂商提供的 API 来使用，对于 GLM3-6B^[14]、Llama2-7B^[15]等参数量较小、开源的模型，可将模型部署到本地使用。此外，还可以使用自己的数据对模型进行微调，以更好地适应特定场景的任务。本研究主要采用了 API 调用的方式使用大模型，也将其部署到本地进行了微调实验。

(1) 大模型的提示词工程

提示词是指一段描述或指令，用于引导模型生成特定的内容^[16]。在调用大模型时设定好提示词，大模型将会按照提示词中指定的方式完成任务。调用大模型时不仅可以要求大模型按照特定的方式完成任务，还可以将一些信息写入提示词中作为先验知识，大模型可以根据这些知识生成回答。提示词工程中提示词的书写需要遵循一定的技巧和原则。首先，书写要目的性明确、描述清晰、避免歧义和误导，这样大模型才能更好地理解提示词中的要求。此外，还有使用分隔符、设定输出格式、指定步骤、给定少量样本等技巧。这些都有可以帮助大模型更好地理解用户的要求。

(2) 大模型的微调技术

大语言模型完成自然语言处理任务的能力，基于在大量的数据上训练。但是将原始的通用大模型应用于特定领域的问答，可能没有较好的效果。原因是通用大模型的训练数据可能对于特定领域的数据并未包含或包含较少。为了使通用的大模型良好地应用到特定领域问答上，需要构建高质量的问-答对数据，对其进行微调训练，将新的知识融合到大模型的大规模参数中。目前，主流的参数高效微调策略技术有 P-Tuning、Lora 等^[17]。相较于全参数微调，它们所需的资源较少，更适合学习和研究。

2.2.4 词嵌入技术

词嵌入技术（Word Embedding）是自然语言处理领域的一种重要技术，它能够将词语转化为实数向量，从而使得计算机能够理解和处理人类语言。词嵌入技术的发展历程可以追溯到早期的 one-hot 编码，这种方法虽然简单，但是无法表示词语之间的语义关系。随着深度学习的发展，开始采用神经网络模型，如 Word2Vec、GloVe 和 FastText^[18]。本研究使用 Word2Vec 技术的主要目的是进行近义词查询。Word2Vec 是由 Google 的 Mikolov 等人提出的一个词向量计算模型^[19]。它的基本思路是输入大量经过分词的文本，输出多维度的向量，用于代指某一词语。词向量的训练主要有 CBOW 和 Skip-gram 两种方式。通过对某两个词的词向量做距离计算得知两个词的相似性。通过查找词表中的某一词语的词向量距离最近的 n 个词向量，即可找到该词语的 n 个近义词。

2.3 本课题方案设计及对比

(1) Text-to-CQL 的实现方案及对比

在 1.3 节中，介绍了几种 Text-to-CQL 的实现方法，可以主要分为基于 Prompt 工程和基于微调两种方式。基于 Prompt 的实现思路较为简单，只需将知识图谱的结构在 Prompt 中进行说明，并描述清楚从自然语言转换为 CQL 的任务，大模型即可根据 Prompt 中的描述，实现 Text-to-CQL 的功能。这种方式无需微调，资源消耗少、易于实现，且结果较为可控，但调用大模型可能会产生一些费用。

基于微调的实现方式需要构造 Text-to-CQL 数据集，选取合适的模型和微调方法，即可进行微调。微调可以使用开源模型的源码或 Llama-Factory^[20]等工具进行，较为方便，但需要一定的 GPU 资源，一般的家用电脑可能无法胜任。可以借助云计算资源进行，这也会产生一定的费用。

对于知识图谱的实体、关系种类较少，可以在 Prompt 中描述清楚的情况，可以考虑采用 Prompt 的方式实现 Text-to-CQL 任务。对于类似 OwnThink 这类的大规模的知识图谱，较难在 Prompt 中描述清楚其中的实体和关系，则可使用微调的方式。对于本研究使用的知识图谱，Prompt 方式较为适宜。两种方式的对比见表 2-1 所示。

表 2-1 Text-to-CQL 的两种实现方式对比

对比项\实现方法	基于 Prompt	基于微调
实现难易程度	较易	较难
资源消耗	较少	较多
花费	一定花费	一定花费

(2)构建知识图谱的数据库类型对比

目前最常用的数据库类型是关系型数据库，将知识图谱建立在关系型数据库上是可行的，但是本研究采用了 Neo4j 图数据库，下面将对构建知识图谱的数据库类型作对比，以阐述采用 Neo4j 图数据库构建知识图谱的原因。

相较于图形数据库来说，关系数据库的理论体系更加坚实，而且关系数据库也有相当成熟的实现，且在安全性上，关系型数据库优于 Neo4j 图数据库^[21]。但是，如果将知识图谱构建在关系型数据库上，需要创建各类的实体表和关系表，面对知识图谱中实体间复杂的关系，查询极为不便。而 Neo4j 图数据库将数据存储在结点和结点间的关系中。图的查找与搜索通过特定算法完成，图数据库针对搜索进行了优化，通过建立索引，可以更快地查询。

综上，对于 Text-to-CQL 的实现，本研究使用了基于提示词的方法，对于存储由节点和关系连接而成的知识图谱，使用 Neo4j 数据库更加合适。

2.4 本章小结

本章主要介绍了本研究所使用的开发工具和技术。首先，详细介绍了使用的开发工具，包括 Python 语言、PyCharm 集成开发环境、Miniconda 虚拟环境、Flask Web 应用框架、Neo4j 图数据库和 MySQL 数据库。其次，深入讨论了本课题主要用到的技术，包括问答系统、知识图谱技术、大语言模型、分词技术和词嵌入技术，以及界面开发所用到的 jQuery、layui、eCharts 等技术。

第3章 系统分析

3.1 需求分析

随着科技的发展，医疗问答系统的需求日益增长。本系统旨在通过医疗问答系统减轻医院线下问诊的压力，并增强公众的医疗知识。由于当前时代人们的生活节奏较快，很多人在有问诊需求时为了节省时间而选择通过网络问诊，本研究开发的医疗问答系统可以帮助用户获取医疗知识，满足他们的一部分问诊需求。此外，围绕医疗问答功能，又设计了一些辅助功能，以增强问答的效果，方便用户。

3.1.1 功能需求

通过对用户的需求分析以及对知识图谱和大语言模型结合的现状，将系统的需要设置为以下几个方面：

(1) 登陆注册功能。用户需要创建账户并登录才能使用问答系统进行问答、反馈、查看历史记录等操作。

(2) 问答功能。问答功能是系统的核心功能。系统应该对于用户输入的医疗相关的问题进行回答，并对非医疗相关的问题进行提示。系统应该可以回答用户关于疾病的症状、用药、挂号、预防、检查、病因等方面的问题。此外，问答功能还应考虑到知识图谱中查不到结果的情况，此时系统应调用大模型进行尝试回答，并对用户进行提示。

(3) 疾病名称提示功能。为了方便用户的输入并减少用户的错误输入，系统应为此开发疾病名称补全提示功能。面对复杂的疾病名称，用户很难正确输入疾病的全称，此时系统可根据用户输入的疾病名称的几个汉字，做模糊查询，为用户提示可能的疾病全称。例如，通过用户输入的“嗜酸性细胞”，系统可为用户提示“嗜酸性粒细胞白血病、嗜酸性细胞肠胃炎”等疾病。另外，当用户输入内容偏口语化，与知识图谱中存储的内容不一致时，可利用词向量技术查找近义词，为用户提供可能的疾病全称提示。例如，通过用户输入的“头疼”，系统可为用户提示“头痛、眩晕”等疾病。

(4) 用户反馈与知识图谱补全功能。为了对知识图谱中的内容进行补充和修改，系统将包含一个用户反馈机制。当用户所问问题在知识图谱中查不到时，系统会利用

大模型自身知识生成回答，此时用户应可进行点赞反馈。当用户所问问题可在知识图谱中查询到时，用户应可对基于知识图谱生成的结果提出点踩反馈。系统将记录这些反馈并提交给后台管理员。管理员将审核用户反馈，并根据需要更新或修改知识图谱，以填补信息空缺或纠正错误信息。

(5)知识图谱查询结果可视化功能。系统应提供一个知识图谱查询结果的可视化界面，使用户能够直观地理解问答系统从知识图谱中检索的信息。可视化界面将展示知识图谱中的节点、关系和属性，帮助用户更好地理解复杂的医疗信息和关系。

(6)对话历史记录功能。为了方便用户回顾过往的咨询内容，系统应提供一个对话历史记录功能。用户可以查看和搜索他们之前的问答记录。用户还可对记录进行删除、查看图谱等操作。

(7)系统设置功能。系统应包含一个设置功能，允许管理员根据需要，对系统的一些功能进行配置。

(8)统计与分析可视化功能。系统应包含一个可以展示用户询问疾病的统计分析结果的功能。可帮助管理员了解某段时间内查询的 TOPn 疾病以及疾病查询数量随时间的变化趋势。

3.1.2 性能需求

(1)稳定性需求。

本系统应具备强大的异常处理能力，以应对可能出现的错误，确保系统的稳定运行。同时，对于用户可能触发的错误操作，系统也需要进行有效的处理和预防。

(2)响应速度需求。

在用户进行对话或使用系统其他功能时，系统需要能够实时处理并快速响应用户的请求，以保证系统的性能优势。在网络环境良好的前提下，游客应能够在任何时间、任何地点无障碍地访问系统。

(3)安全性需求

在应用设计过程中，我们必须充分考虑到用户数据的安全性。例如，我们需要对用户的密码进行加密存储，同时确保不同用户之间无法查看对方的对话数据等。这些措施都是为了保护用户的隐私和数据安全。

3.2 可行性分析

(1)经济可行性。在满足用户需求的同时，开发一个基于大模型结合知识图谱的医疗问答系统所需的成本比较低，只需要一台电脑就可以进行编译和调试。通过 API 调用大模型，将会产生一定的费用。若本地部署模型或进行模型微调，可利用腾讯云、阿里云等云计算资源，也会产生一定的费用。但费用总体可以接受并可控。所以，本系统在经济上是可行的。

(2)技术可行性。完成本研究的医疗问答系统的设计与实现，主要是掌握 Python 语言、Flask 框架、以及对大模型和知识图谱技术的利用。虽然本研究的方向较新，但也可以查询到中外学者做的一些研究。经过学习后，本研究可以顺利地设计并开发该系统。所以本研究在技术上是可行的。

(3)社会可行性。医疗问答系统的设计与实现可以满足人们对疾病相关问题的咨询需求。并且这种系统能够提供快速、较为准确的信息，有助于提高公众的健康意识和疾病预防知识。在当前信息化时代，人们越来越依赖网络获取健康信息，因此，一个这样的系统将会受到欢迎。所以本研究是具有社会可行性的。

3.3 系统开发环境

操作系统：Windows 11；

系统类型：64 位系统；

CPU：Intel(R) Core(TM) i5-10210U 8 核

硬盘：512G；

内存：16G；

开发语言：Python；

开发工具：Pycharm；

数据库：Neo4j 数据库、MySQL 数据库；

3.4 本章小结

本章主要对本研究的系统进行了详细的分析。首先，进行了需求分析，明确了系统的主要功能。其次，进行了可行性分析，从经济、技术和社会三个方面对系统的可行性进行了评估。最后，我们介绍了系统的开发环境。

第4章 系统设计

4.1 总体设计

4.1.1 总体模块设计

本研究开发的医疗问答系统，旨在利用先进的大模型和知识图谱技术，为用户提供准确、及时的医疗疾病咨询服务。在该系统中，用户可以通过简单的询问获得关于疾病、症状、科室、治疗方法、药物使用等方面的回答。开发本系统的目的是为了提升公众健康知识，减轻医院线下问诊的压力。

根据第三章的需求分析，本系统主要分为以下几个模块：问答模块、疾病名称提示模块、用户反馈模块、知识图谱补全模块、知识图谱可视化模块、对话历史记录模块、系统设置模块和登录注册模块。以上模块共同构成了本医疗问答系统的总体架构，确保系统能够高效、稳定地运行。系统的总体功能模块图如图 4-1 所示。

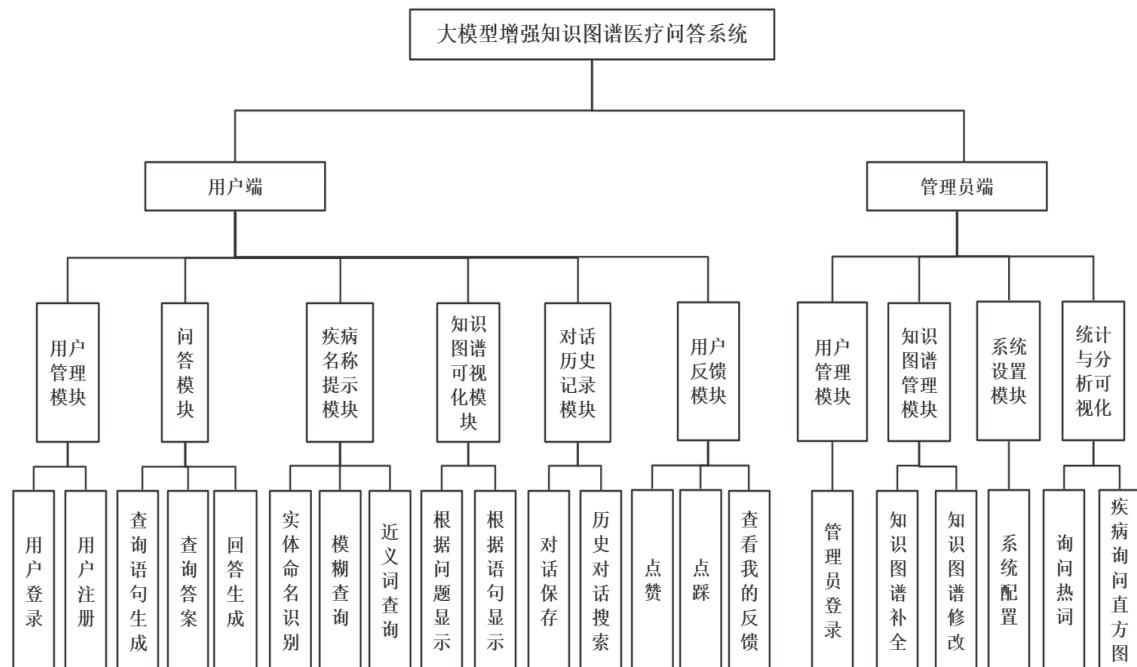


图 4-1 系统的总体功能模块图

下面将对系统的各个模块的设计做简要说明，系统分为用户端和管理员端。用户端有以下模块：

(1) 用户管理模块。用户注册、登录后才可进行问答、反馈、历史记录等操作。

(2)问答模块。作为系统的核心，负责处理用户的问题，经过 Text-to-CQL 任务得到 CQL 语句，利用 CQL 语句查询知识图谱得到答案，最后利用查询结果为用户提供回答。

(3)疾病名称提示模块。首先利用实体命名识别技术提取问题中关于疾病的名称的文字。再利用模糊查询和近义词查询，查询出与用户输入相关的疾病名称，将着色写名称反馈给用户做提示，以辅助用户在提问时准确输入疾病名称，减少错误和提高查询效率。

(4)知识图谱可视化模块。通过可视化界面展示知识图谱查询结果，帮助用户更好地理解医疗信息在知识图谱中的存储。

(5)对话历史记录模块。保存用户的咨询历史，方便用户随时查阅过往的问答。

(6)用户反馈模块。允许用户对问题的回答进行反馈，对于基于知识图谱生成的回答，用户可进行点踩；对于基于大模型生成的回答，用户可进行点赞。反馈信息将提供给管理员，以帮助完善知识图谱。此外用户查看对自己的反馈历史，获取反馈进度。

用户端有以下模块：

(1)用户管理模块。管理员登录后才可进行管理员相关的操作。

(2)知识图谱管理模块。管理员审核待补全或待修改的内容，对知识图谱进行补全或修改。

(3)系统设置模块。供管理员使用，用于配置系统参数和功能。

(4)统计与分析可视化模块。对用户问答的疾病数据进行可视化展示，以帮助管理员进行查看和分析。

4.1.2 知识图谱结构设计

本研究采用的知识图谱源自 GitHub 上的一个开源项目^[22]。使用 neo4j 图数据库搭建知识图谱。该知识图谱包含疾病、症状、科室、检查项目、药品及食物六类实体，各实体间的关系见表 4-1。

表 4-1 实体之间的关系

关系名称	关系连接的节点类型	描述
accompany_with	疾病 疾病	并发症
belongs_to	疾病 科室	疾病所属科室

表 4-1 实体之间的关系（续表）

common_drug	疾病 药品	疾病常用药物
need_check	疾病 检查项目	疾病诊断检查
has_symptom	疾病 症状	疾病症状
do_eat	疾病 食物	疾病宜吃
no_eat	疾病 食物	疾病忌吃

此外，疾病实体还有简介、病因、治疗方法、治疗周期、易感人群、治愈率等属性。以疾病“胰腺炎”为例，在知识图谱中相关的实体与关系如图 4-2 所示。

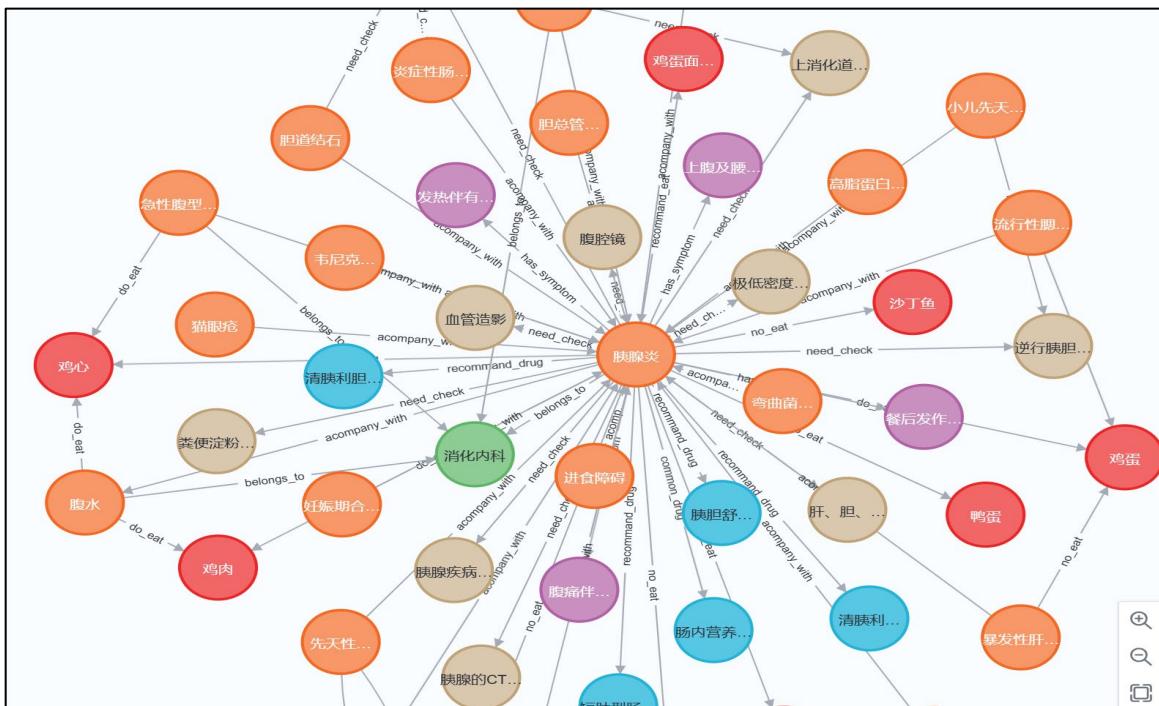


图 4-2 疾病胰腺炎的相关实体与关系

4.1.3 数据库设计

本项目中用户登录、对话历史记录、管理员审核 和用户反馈功能使用了 MySQL 数据库存储数据，项目中共创建了四张表，下面将对各张表的结构加以说明。

首先是用户表，用户表用于登录、注册功能。记录了用户名、密码等信息，用户表的设计如表 4-2 所示。

表 4-2 用户表(Users)

字段名	描述	数据类型	长度	是否主键
id	编号	int	8	是
username	用户名	varchar	10	否
password	密码	varchar	20	否
identity	身份	bool	1	否

对话历史记录表记录了历史对话中的问题和回答，并记录该问答所属的用户 id、问题对应的 cql 语句和时间信息，该表的设计如表 4-3 所示。

表 4-3 历史记录表(QAHistroy)

字段名	描述	数据类型	长度	是否主键
id	编号	int	8	是
question	用户问题	varchar	200	否
answer	回答	varchar	400	否
cql	cql 语句	varchar	200	否
userId	用户 id	int	8	否
time	时间	varchar	20	否

待审核内容表记录了待审核的用户反馈，包含 cql 语句、系统回答、三元组、点赞次数和点踩次数，该表的设计如表 4-4 所示。

表 4-4 待审核内容表(AuditContent)

字段名	描述	数据类型	长度	是否主键
cql	cql 语句	varchar	200	是
answer	系统回答	varchar	400	否
time	时间	varchar	20	否
triples	提取的三元组	varchar	200	否
likes	点赞次数	int	8	否
dislikes	点踩次数	int	8	否

用户反馈表记录了用户的反馈，包含用户 id、cql 语句、反馈内容是否点赞、是否已审核，该表的设计如表 4-5 所示。

表 4-5 用户反馈表(UserReview)

字段名	描述	数据类型	长度	是否主键
userId	用户 id	int	8	是
cqlMatch	cql 语句	varchar	200	是
reviewContent	反馈内容	varchar	200	否
isLike	是否点赞	bool	1	否
valid	是否已审核	bool	1	否

4.2 详细设计

本节将根据总体设计的内容，进一步对各个模块的功能和实现进行详细设计。下 4.2.1 至 4.2.6 为用户端的详细设计，4.2.7 至 4.2.8 为管理员端的详细设计。

4.2.1 登陆注册模块

登陆注册模块的主要功能为登陆注册，为系统增加身份验证。登录、注册的流程见图 4-3、4-4 所示。

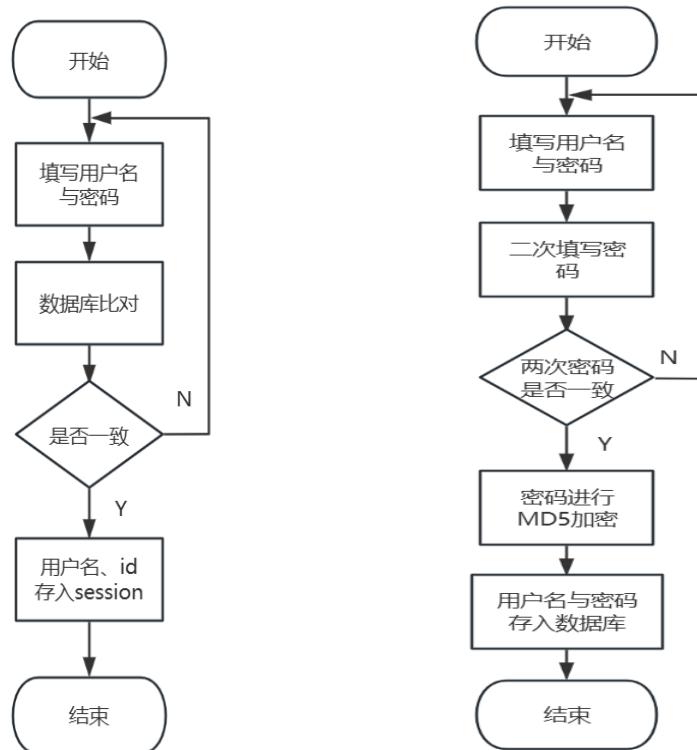


图 4-3 登录的流程图

图 4-4 注册的流程图

在用户注册过程中，系统要求提供用户名、密码以及密码的二次确认，以验证用户信息的完整性和一致性。系统对于空白输入、过长的输入以及不匹配的密码等情况进行了严格的校验。为了进一步加强数据安全，密码在传输和存储过程中均采用 MD5 加密算法进行加密处理。用户登录时，系统通过比对输入的用户名和密码与数据库中存储的信息，来验证用户身份。验证成功后，系统会将用户名和用户 id 存储在会话(session)中，以便在用户的后续操作中快速检索和使用。后续的对话、历史记录、反馈、查看我的反馈等操作中都会使用到用户的 id。

4.2.2 问答模块

问答模块包含查询语句生成、答案查询和回答生成三个主要功能。问答模块的流程如图 4-5 所示。

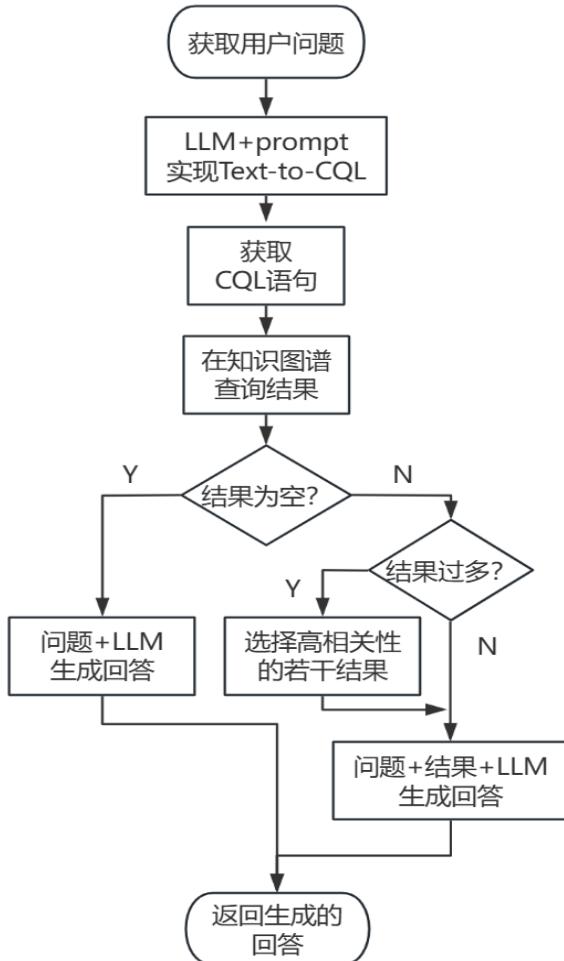


图 4-5 问答的流程图

查询语句生成功能在第 2.3 节中经过详细的方案比较后，采用了基于 Prompt 的

方法。为了优化 Text-to-CQL 任务的执行，本研究对 Prompt 进行了设计调整，以确定最适合的 Prompt 内容。可能的一个例子为：你是一位知识图谱专家，请根据给定的知识图谱图结构和问题，写出 Cypher 查询语句，图的结构如下：----{schema}---- 问题如下：----{question}----^[20]。本研究的提示词告知大模型知识图谱的实体和关系类型，令其从问题中提取出实体、关系类型和实体名称。之后匹配 CQL 语句。

此外，还需告知大模型只返回语句，不用给出解释，不用道歉。并强调不要使用 schema 以外的实体和关系的类型等内容。

答案查询利用了 py2neo 库，实现在 Python 语言中对 Neo4j 数据库的查询。获取到查询结果后，进行格式转换，将其转换为 string 类型。在答案生成时，将查询结果与用户问题一并交给大模型，令其根据查询结果生成针对问题的回答。生成答案时，若查询不到结果时，代表知识图谱中不存在相关内容，此时利用大模型生成回答，并提示用户注意甄别，用户可对该回答进行反馈操作。若查到的结果过多时，需要利用大模型选取最优的几个结果。

4.2.3 疾病名称提示模块

疾病名称提示模块包含实体命名识别、模糊查询和近义词查询三个主要功能。疾病名称提示的流程图见图 4-6 所示。

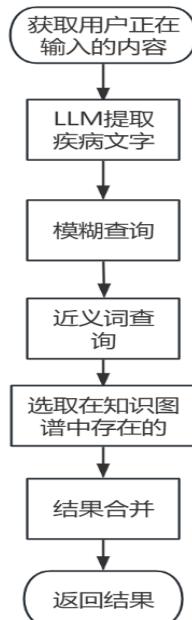


图 4-6 疾病名称提示的流程图

首先，实体命名识别功能负责从用户输入中提取与疾病相关的文本信息。其次，

模糊查询功能通过在知识图谱中对提取的文本进行模糊匹配，以识别相关疾病实体。最后，近义词查询功能会使用 Word2Vec 技术训练后的词向量，通过计算词向量相似度，检索与查询词汇语义相近的词项。

在实体命名识别过程中，我们利用了大型语言模型来实现。模糊查询则通过 Cypher 查询语言的 CONTAINS 关键字实现，在知识图谱中检索包含特定文本的疾病名称。对于近义词查询，我们使用训练好的词向量模型来发现给定词汇的语义邻近词，并选取在知识图谱存在的词语作为结果。最终，将模糊查询和近义词查询的结果综合起来，为用户提供综合性的疾病名称提示。

4.2.4 知识图谱可视化模块

知识图谱可视化模块的核心功能是以图形化界面呈现知识图谱查询结果的结构和内容。该模块的流程图见图 4-7 所示。

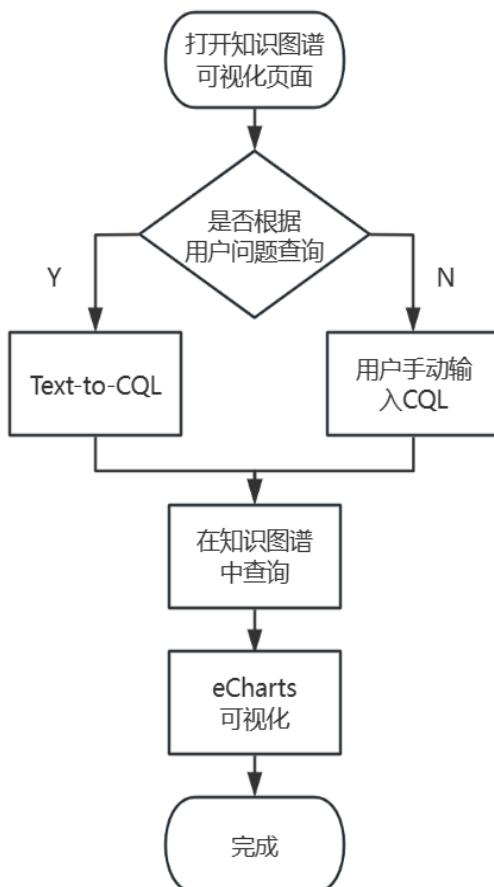


图 4-7 知识图谱可视化的流程图

本模块支持两种模式：一种是将基于用户问题的查询结果可视化展现，另一种是

将基于用户输入的 CQL 语句的查询结果可视化展现。在基于用户问题的模式中，系统会利用 Text-to-CQL 任务生成的 CQL 语句在知识图谱数据库中检索相关的实体和关系。检索结果以 JSON 格式返回，经过格式转换后，通过 eCharts 工具进行可视化展现，以便将查询结果以直观的图形方式展示给用户。在基于用户输入的 CQL 语句的模式中，用户可以直接输入 CQL 语句，系统利用输入的 CQL 语句在知识图谱数据库中检索相关的实体和关系。通过 eCharts 工具进行可视化展现。

4.2.5 对话历史记录模块

对话历史记录模块的主要功能是保存用户对话的历史记录，并可以对其进行查询、删除操作。为了增强用户对话内容的可视化理解，本系统采用词云生成技术，根据用户的对话内容构建词云。此外，系统还允许用户查看每条对话对应的知识图谱查询结果，以加深对话内容的理解。历史记录以表格形式呈现，其中除了用户问题、系统回答、对话时间以及相应的 CQL 语句。表格每列末端配备两个功能按钮，分别用于删除特定对话记录和查看相关知识图谱节点。

用户打开历史记录页面后，系统会自动查询该用户的最近 200 条历史数据，以表格的形式呈现，之后用户可以进行关键词查询、查看词云、删除历史记录等操作。

4.2.6 用户反馈模块

用户反馈模块包含点赞、点踩和查看我的反馈功能。点赞功能在知识图谱无法提供答案时起作用，此时大型语言模型介入生成回答。用户点赞后，系统会利用大模型，将回答中的信息以三元组形式提取出来，暂存于待审核数据库中，等待管理员的进一步审核。

点踩功能则在知识图谱提供的回答未能满足用户需求时发挥作用，用户可以提交不满意的反馈并附上理由，这些信息同样会被记录在待审核数据库中，供管理员审核。

用户可以在“我的反馈”页面中跟踪其提交的反馈的处理进度。这一模块的设计旨在通过用户参与，增强系统的自我优化能力，提升整体的用户体验。

用户反馈模块的流程图见图 4-8 所示。

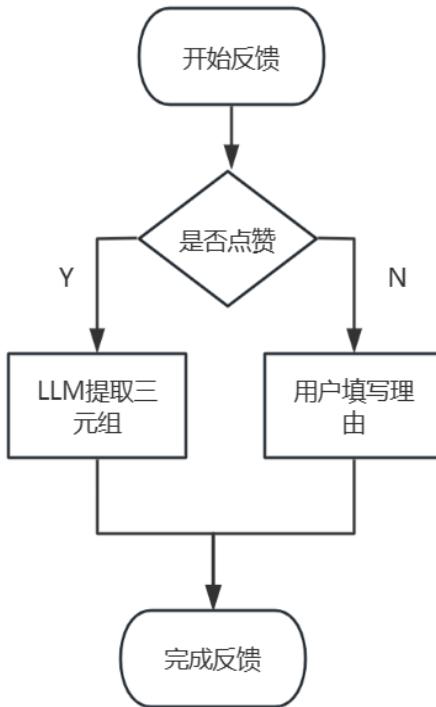


图 4-8 用户反馈的流程图

4.2.7 知识图谱管理模块

知识图谱管理模块的主要功能为知识图谱补全和知识图谱修改。具体来说，管理员负责审查用户反馈，并据此对知识图谱进行必要的补充和修正。当用户对于系统针对知识图谱中尚未包含的实体，利用大模型生成的回答表示认可时，系统会自动将相关的 CQL 查询语句、系统生成的回答以及从回答中提取的三元组从数据库中查询出来展示给管理员，管理员有权对这些信息进行编辑。基于审核后的三元组信息，系统会自动生成相应的 Cypher 语句，以便利用 Cypher 语句在知识图谱中添加新的实体和关系。管理员可以利用知识图谱可视化模块来检查新增内容，并决定是否撤销或确认这些变更。

当用户对于系统基于知识图谱中已有实体生成的回答表示不认可时，用户需提供具体的反馈理由。系统将这些反馈连同相关的 CQL 查询语句和系统回答一并呈现给管理员。管理员将根据用户提供的理由来判断是否需要对知识图谱进行调整。在此过程中，管理员可以通过可视化模块来审查相关信息，并通过输入相应的 Cypher 语句来更新知识图谱。

管理员使用该模块进行审核的流程图见图 4-9 所示。

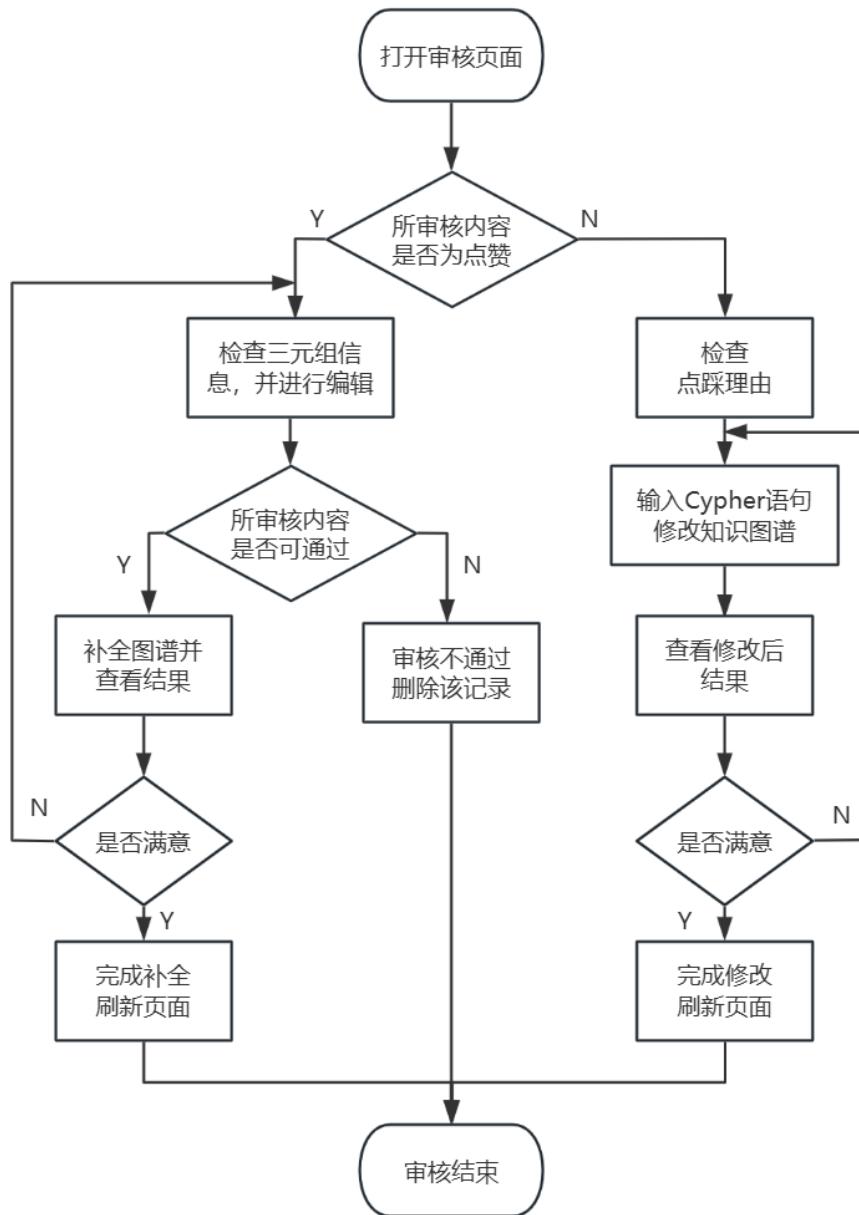


图 4-9 知识图谱管理审核的流程图

4.2.8 系统设置模块

系统设置模块的主要功能为管理员对系统进行配置。管理员可以通过此模块进行以下设置：

- (1)对话历史记录保存开关，管理员可以启用或禁用对话历史记录的保存功能。这一功能对于追踪用户咨询历史至关重要。
- (2)历史记录保存数量设置，此设置允许管理员定义系统应保留的对话历史记录的条数，以平衡存储资源和信息可用性。

- (3)近义词查询开关，通过开启近义词查询功能，系统能够在用户输入时对可能的相似的疾病名称做出提示。
- (4)大模型选择，管理员可以根据需求选择不同的大型语言模型，以适应不同的要求。
- (5)api-key 和 baseurl 设置，根据所选的大语言模型，配置与之相对应的 api-key 和 baseurl。

4.2.9 统计与分析可视化模块

统计与分析可视化模块包含查看询问热词和查看疾病询问直方图功能。

查看询问热词功能会对于某一时间段内用户历史询问的所有疾病中 TOPn 的疾病做词云可视化展现，以帮助管理人员查看、了解、分析用户查询疾病的数据。其中的时间段、TOPn 都为管理员可调参数。

查看疾病询问直方图功能会显示某一疾病或某几个在一年的各个月份中查询数量的情况。其中的疾病为管理员可调参数。

4.3 本章小结

在本章中，我们详细探讨了系统设计的各个方面。本章首先介绍了系统的总体设计，包括模块设计、知识图谱的结构以及数据库架构。介绍了系统的功能模块，阐述了本系统所用知识图谱的结构，说明了本系统数据库表的情况。随后，我们深入分析了系统的详细设计，涵盖了问答模块、疾病名称提示模块、用户反馈模块、知识图谱可视化模块、知识图谱补全模块、对话历史记录模块、系统设置模块以及登录注册模块。分别介绍了它们的主要内容和详细流程，为实现系统打下基础。

第 5 章 系统实现

本章节将在前文分析的基础上，介绍系统实现过程，并分模块介绍每个模块的实现过程。

5.1 系统配置

下面将从大模型调用、知识图谱构建、后端服务器配置三个方面介绍系统实现过程中的配置情况。

(1) 大模型调用。系统中使用 openai 库以及 zhipuai 库提供的函数来调用大模型厂商提供的大模型服务。以 openai 库为例，本研究开发了 get_response 函数用于整理需要向大模型 API 提交的请求数据，在该函数中，首先需要填写好 api-key，这是调用大模型时必填的内容，用于身份验证。其次将要向大模型发送的消息内容整合成以下格式：message: [{‘role’: ‘system’, ‘content’: prompt}, {‘role’: ‘user’, ‘content’: question}]。随后，调用 openai.ChatCompletion.create 函数，将模型名称、message 等参数填写好，即可接收到大模型的回复。一个简单的例子如图 5-1 所示。

```
def get_response(prompt, question, model='gpt-3.5-turbo'):
    messages = [{'role': 'system', 'content': prompt},
                {'role': 'user', 'content': question}]
    response = openai.ChatCompletion.create(
        model=model,
        messages=messages, # 提交的 message
        max_tokens=300, # 控制生成的最大标记数
        temperature=0.5, # 控制模型的随机性
    )
    return response['choices'][0]['message']['content']

text = f"Q: 中国的首都? "
prompt = f" 你的任务是仿照下面的风格回答问题: Q: 法国的首都? A: 巴黎。
res = get_response(prompt, text)
print(res)

A: 北京。
```

图 5-1 大模型 API 调用的代码案例

(2) 知识图谱构建。系统中使用 neo4j 图数据库构建了医疗知识图谱，为此本研究租用了云服务资源，在公网可访问的服务器上安装部署了 neo4j 图数据库。为了使数据库可以被公网访问，需要将其配置文件中的监听地址改为 0.0.0.0，并将其 http 服务开启，指定服务端口，在云服务器运营商平台上开放相关端口。配置文件的修改如图 5-2 所示。

```
87 dbms.default_listen_address=0.0.0.0
88 dbms.connector.http.enabled=true
89 dbms.connector.http.listen_address=:7474
90 dbms.connector.http.advertised_address=:7474
91 dbms.security.auth_enabled=false
```

图 5-2 对 neo4j 数据库配置文件的修改

然后，调用 py2neo 库，将数据集中的内容填充到 neo4j 图数据库中，构建本研究所需的医疗知识图谱。可以访问服务器所在公网 IP:7474 来登录数据库查看知识图谱，数据库登陆后的界面如图 5-3 所示。

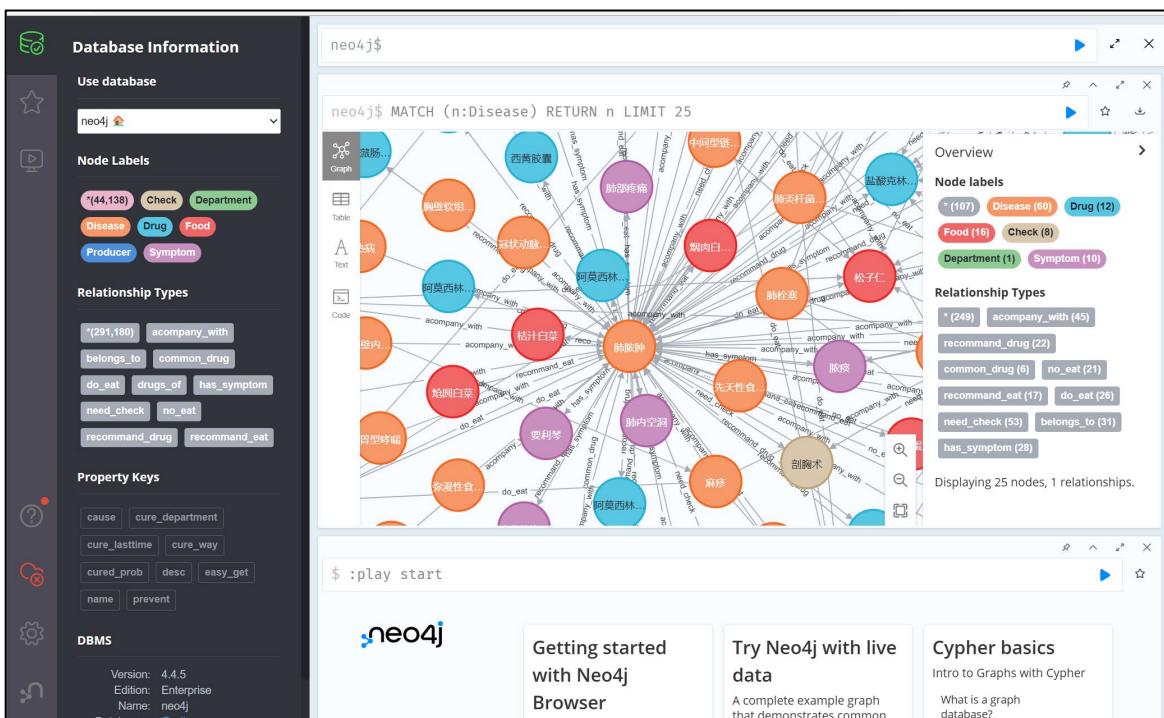


图 5-3 基于 neo4j 图数据库构建的知识图谱

在代码中，可通过 py2neo 库的 `Graph("IP:7474", auth=("用户名", "密码"))` 的方式初始化 `graph` 对象，最后调用该对象的 `run` 函数，传入要执行的 CQL 语句，即可实现对知识图谱的查询操作。

(3)后端服务器配置。本研究开发的问答系统采用了前后端分离的设计理念，采用 Flask 框架开发，利用该框架的路由概念，设计开发了不同的 url 地址，用以接收前端不同功能传来的请求，接收数据后在后端的相应函数中运行，向前端返回结果。部分路由的设置如图 5-4 所示，跟在路由后的函数负责处理前端通过该接口发来的请求。

```

35 @app.route('/chat', methods=['POST'])
36 > def chat(): ...
76
77 @app.route('/search1', methods=['POST'])
78 > def search1(): ...
89

```

图 5-4 部分路由的设置

此外，还需设置后端服务的监听 IP 和端口，为了公网可访问，监听 IP 设置为 0.0.0.0，端口设置为 8080。通过 IP:8080/xxx 即可向后端服务发送请求。

5.2 系统主页

本研究开发了一个系统主页用以对系统进行介绍，系统主页如图 5-5 所示。

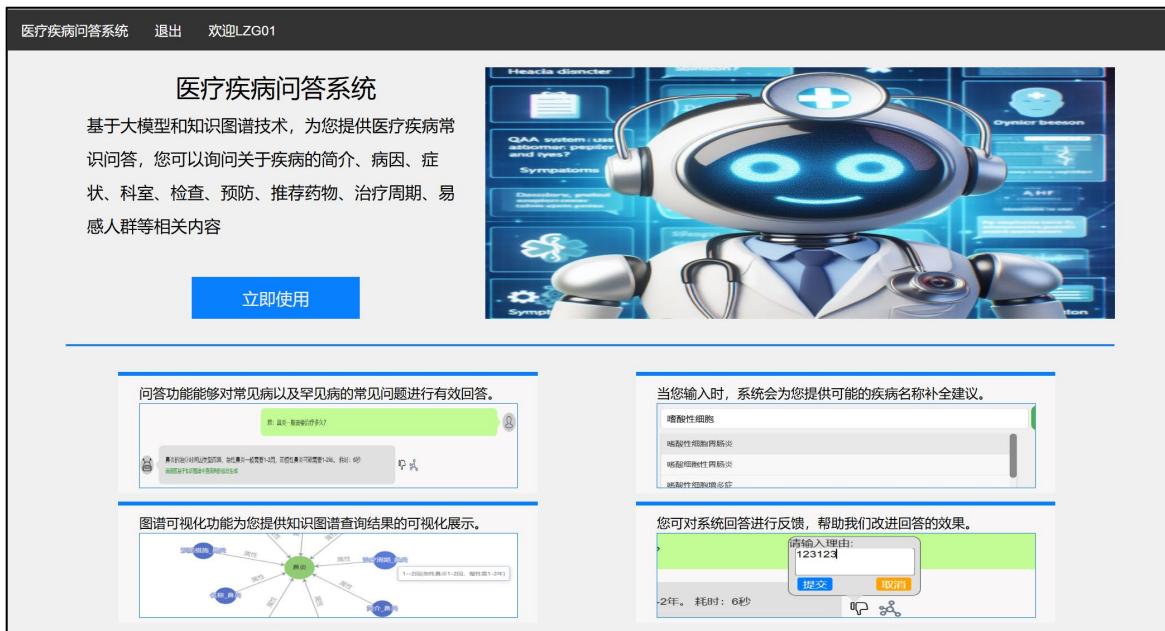


图 5-5 系统主页

主页中包含了系统主要功能的介绍以及系统的使用入口，点击“立即使用”即可进入系统。

5.3 登陆注册模块的实现

登陆注册模块的实现较为简单，用户必须先登录才能进入系统进行问答等操作，在系统主页中，用户点击导航栏中的“登录”项，即可打开登陆页面。用户进入问答页面时，若没有登陆，系统会自动跳转到登陆页面，如图 5-6 所示。

图 5-6 登陆页面

登陆后，会将用户的用户名、用户 ID 存入 session 中，以便后续使用。若用户尚未注册，可点击“注册”按钮进行注册，如图 5-7 所示。

图 5-7 注册页面

密码在前端经过 MD5 加密后传输给后端，存储在数据库中，这样即使数据被泄露，也不会直接暴露用户的原始密码。在后端编写了 register 函数，以接收前端传来的用户名和加密后的密码，在数据库中保存。此外，对于登录的请求，也在后端编写了 login 函数，以接收前端传来的用户名和加密后的密码，在数据库中进行查询，将用户 ID、用户名返回给前端。

5.4 问答模块的实现

问答模块的设计旨在实现三个核心功能：查询语句生成、答案查询和回答生成。首先，查询语句生成功能致力于实现文本到 CQL 语言的转换，即 Text-to-CQL 任务。本研究采用 API 调用与提示词（Prompt）相结合的方法，借助大语言模型来

完成此项任务。具体而言，通过调用 OpenAI 库与 Zhipuai 库，本研究能够调用 ChatGPT 模型与 ChatGLM 模型，并实现了一个 `get_response` 函数，以提交用户输入和提示词并获取模型输出。对于该任务，本研究设计的提示词中包含对知识图谱结构的说明、对 Text-to-CQL 任务目标的说明以及一些指令，令大模型只输出 CQL 语句，不输出无关内容，多条语句以\$分割等。实现 Text-to-CQL 的示意图如图 5-8 所示。

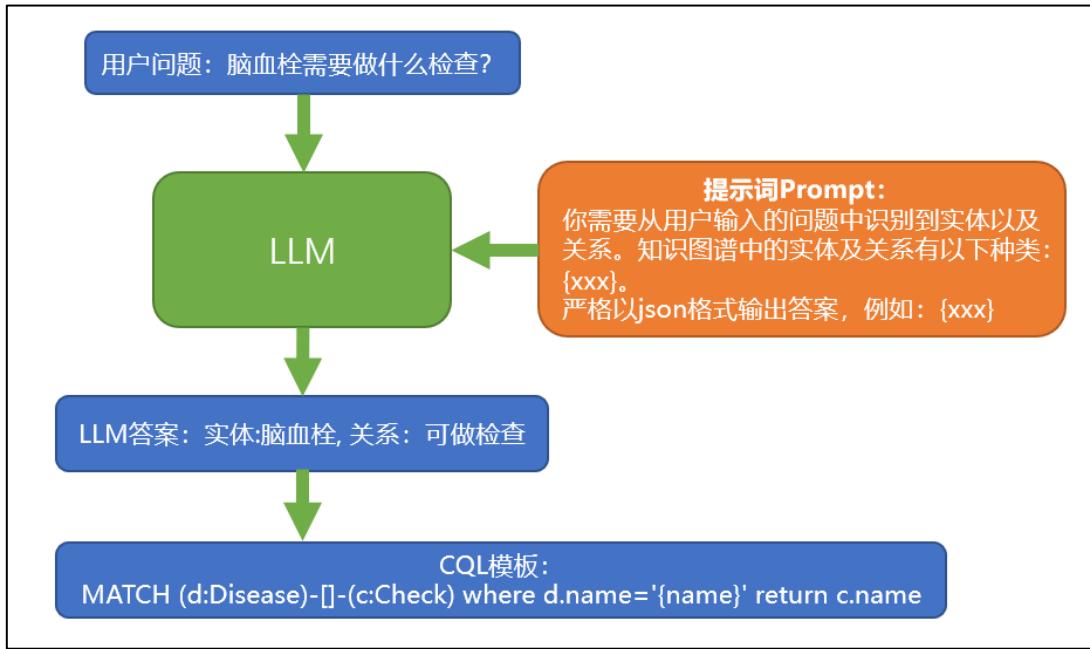


图 5-8 实现 Text-to-CQL 的示意图

其次，答案检索功能在知识图谱中进行答案查询，本研究所依托的知识图谱建立在 Neo4j 图数据库之上。通过引入 Py2neo 库，实现了 Python 语言中与 Neo4j 数据库的连接，并执行 Cypher 查询。由上一步生成的 CQL 语句通过 Py2neo 提供的接口执行，执行后的查询结果以 `py2neo.cypher.Cursor` 对象的形式返回。本研究开发了相关函数，以遍历 `Cursor` 对象并将结果转换为字符串的格式，以便后续使用知识图谱查询结果作为回答的依据传递给大模型。

最后，回答构建功能基于检索结果和用户问题生成回答。在查询结果为空时，即知识图谱中缺乏相关内容时，系统将直接利用大模型的自身知识对问题进行回答。反之，若查询结果非空，表明知识图谱中已存在相关信息，系统将依据这些结果调用大模型生成回答。在前者情况下，用户可通过反馈模块对回答进行点赞，协助管理员补全知识图谱；在后者情况下，用户可通过反馈模块对回答进行点踩反馈，指导管理员对知识图谱进行修正。

问答模块的三个功能均集成于 chat 函数中，该函数接受用户问题作为参数，并返回最终回答、生成的 CQL 语句以及回答状态标志。回答状态共分为三类：无法回答的无关输入、知识图谱中检索到答案的情况，以及知识图谱中未检索到答案的情况。

此外，本研究还为问答模块开发了前端界面，包括对话区域、输入框和发送按钮等组件。用户在输入框中键入自然语言问题，并通过点击发送按钮提交 Ajax 请求至后端。在后端经过一系列处理，返回给前端最终回答、生成的 CQL 语句以及回答状态标志这三种数据，前端将用户输入和最终回答呈现在问答区域中，并将 CQL 语句保存在一个不可见的元素中，以方便后续用户反馈、知识图谱可视化功能对 CQL 语句的使用，前端根据回答状态标志来判断当前问题对应的回答类型，以实现不同情况下的不同处理。当知识图谱中检索到答案时，回答的右侧应为点踩图标和图谱可视化图标，否则，回答的右侧为点赞图标。问答模块的界面利用 HTML +CSS 实现。前端代码主要利用 jQuery 实现，主要负责数据的收发、点击图标时的处理。问答模块的界面设计如图 5-9 所示。



图 5-9 问答模块的界面

在常规操作中，当知识图谱能够检索到相应的答案时，系统将按预期生成回答，并通过绿色文本向用户提示。用户可通过点击对话界面右侧的反馈按钮（点

踩），提交对当前回答的评价。以帮助管理员修改知识图谱。此外，用户还可通过点击图谱按钮，查看当前回答所依据的知识图谱查询结果的可视化表示。相反，在特殊情况下，若知识图谱未能检索到答案，系统将调用大语言模型尝试生成回答，用户可通过点击点赞按钮，提供对当前回答的反馈。帮助管理员补充知识图谱。

为确保即使在页面刷新后，对话内容仍然保留在对话区域内，本研究采取了将对话历史存储在页面会话中的策略。这一设计选择不仅提高了用户体验，而且保持了对话的持久性。如图 5-10 所示。当用户想要清除会话时，本系统提供了清楚对话按钮，以清除绘画中的对话。

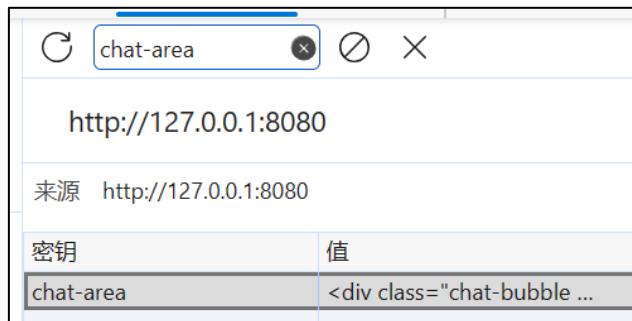


图 5-10 session 中保存的对话区域内容

5.5 疾病名称提示模块的实现

疾病名称提示模块包含实体命名识别、模糊查询和近义词查询三个主要功能，其主要目的是当用户输入时为用户提供可能的疾病名称补全提示，提升用户输入的准确性和便利性。首先，实体命名识别致力于从用户正在输入的文本中提取出与疾病相关的文字，本研究采用 API 调用与提示词（Prompt）相结合的方法，借助大语言模型来完成此项任务。

其次，模糊查询旨在利用从用户输入文本中提取出的与疾病相关的文字，做模糊查询，查询所有疾病中包含这些文字的疾病名称。在 Cypher 语句中有一个名为“CONTAINS”的关键字，它可以寻找某一属性值中是否包含要查找的指定字符串。我们利用该关键字进行模糊查询。同时还利用了 ORDER BY 关键字对疾病名称的长度做排序，LIMIT 关键字对取前十个结果。例如，用户输入为：“我得了病毒性”，提取出来的文字为“毒性”，进行模糊查询的 Cypher 语句为“MATCH (n:Disease) WHERE n.name CONTAINS '病' AND n.name CONTAINS '毒' AND n.name CONTAINS '性' RETURN n.name ORDER BY size(n.name) LIMIT 10”，通过该 Cypher 语

句在知识图谱中进行模糊查询，查询到的结果为“病毒性感冒、病毒性脑炎”等。

最后，利用近义词查询对所输入的疾病名称进行近义词查询。首先对用户的输入中的疾病相关词语进行近义词查询，再选择近义词中存在的近义词，作为近义词查询的结果。目的之一是为用户提供可能的相近的疾病例如，用户输入“肺炎”，系统可利用近义词查询找到与其相关的“支气管炎”疾病。目的之二是或用户输入的不标准的、口语化的描述进行修正，例如知识图谱中存有“腹泻”疾病，但日常口语化的如“拉肚子”之类的描述并没有在知识图谱中，这就会造成查询失败。有了近义词查询后，可利用近义词查询到与用户输入的“拉肚子”相关的“腹泻”提供给用户做选择，如此就可减少因输入口语化、描述不标准造成的查询失败。

近义词查询利用了词向量模型进行。首先需要搜集医疗相关的对话数据集，之后使用分词技术，将数据集中的句子进行分词处理。最后使用 Word2Vec 技术将分词后的一个个词语进行词向量模型训练的操作，训练完成后，每个单词都会对应一个或多个向量，即词向量。在近义词查询时，我们需要计算被查询单词的词向量与所有单词的词向量中距离最近的 n 个，即可找到与其最相近的 n 个单词。词向量计算的示意图如图 5-11 所示。

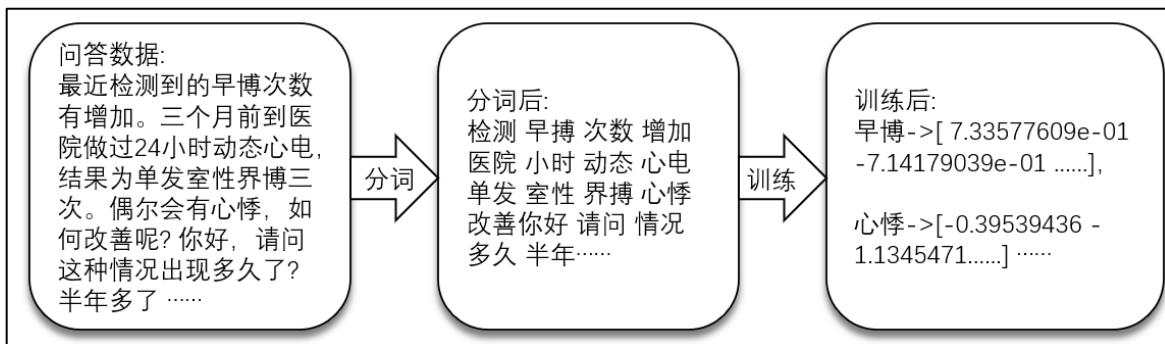


图 5-11 词向量计算的示意图

为了完成分词的操作，本研究引入了 pkuseg 库^[21]进行分词，该库的优点是可以对医学领域进行针对性的分词，效果优于 jieba 库。也利用了停用词表去除停用词。为了完成词向量计算的操作，本研究引入了 gensim 库，利用其 Word2Vec 函数进行词向量计算，并将训练好的模型保存。近义词查询时，读取模型，利用 gensim 库的 most_similar 函数进行查找最相似的 n 个单词的操作。

本研究编写了 search 函数实现了实体命名识别、模糊查询和近义词查询三个功能，函数输入为用户输入的内容，输出为一个个疾病名称补全后的句子。当用户输入时，系统会有间隔地像后端发送用户当前的输入内容，经过 search 函数处理后，将补全后的句子返回前端，在前端的输入框下面进行显示。如图 5-12 所示。

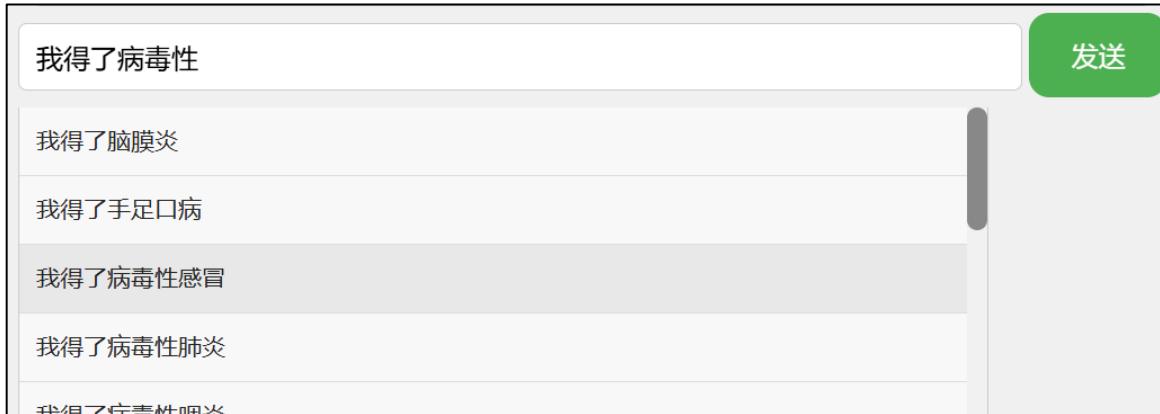


图 5-12 疾病名称补全提示

用户可移动鼠标或按下方向键选择任意一条补全提示内容，鼠标点击或按下回车键后，该条内容将会填入输入框中。

此外，当用户输入时没有选择任何提示的内容，输入的名称在知识图谱中查询不到，但近义词查询可查到相近的疾病时，系统会对用户进行提示，以方便用户。例如用户输入中包含“头疼”，系统已为其推荐了“头痛”作为提示，但用户并未选择，此时会查询失败，但会对用户进行提示，如图 5-13 所示。



图 5-13 对用户的提示

5.6 知识图谱可视化模块的实现

图谱可视化模块可以对知识图谱查询结果进行可视化展现，以便帮助用户的更好的查看知识图谱相关信息。该模块可根据用户的问题或根据用户输入的 CQL 语句展现知识图谱。用户点击系统回答右侧的图谱图标，即可查看当前问题在知识图谱中的查询结果的可视化展现。图谱图标如图 5-14 所示。

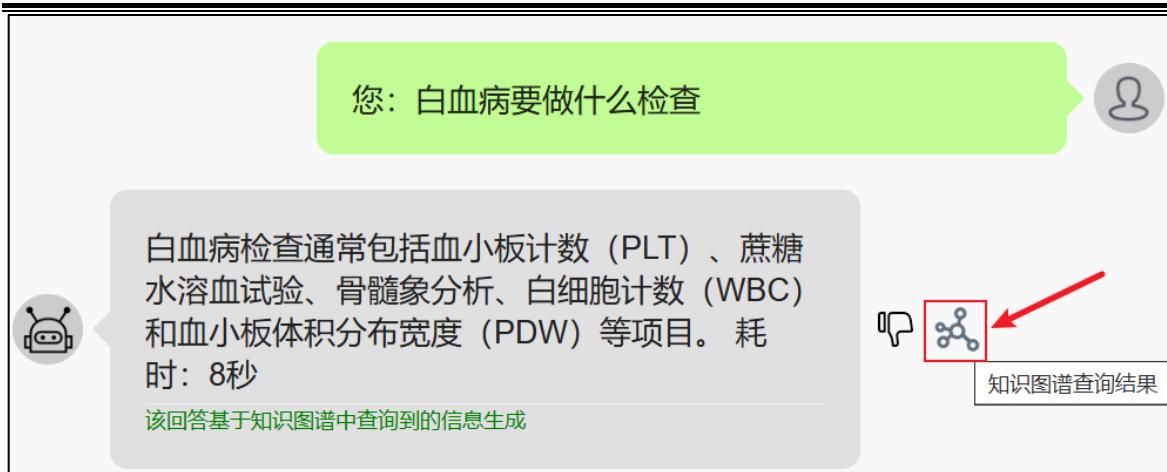


图 5-14 图谱图标

图谱可视化页面如图 5-15 所示，此时展现的是根据问题“白血病要做什么检查”检索到的知识图谱内容。若在上方的输入栏中输入 CQL 语句，点击查询图标，可根据用户输入的 CQL 语句展现查询结果。

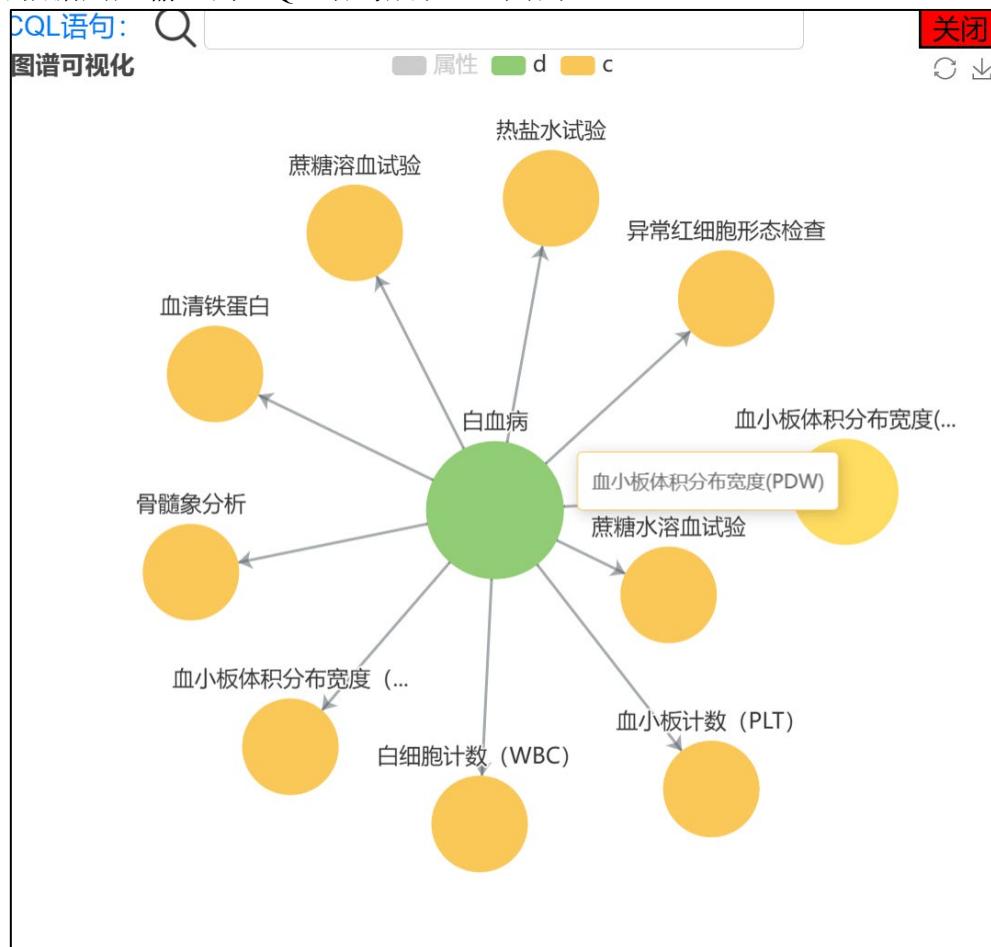


图 5-15 图谱可视化页面

如图 5-16 所示，此时展现的是根据用户输入的 CQL 语句查询到的内容。

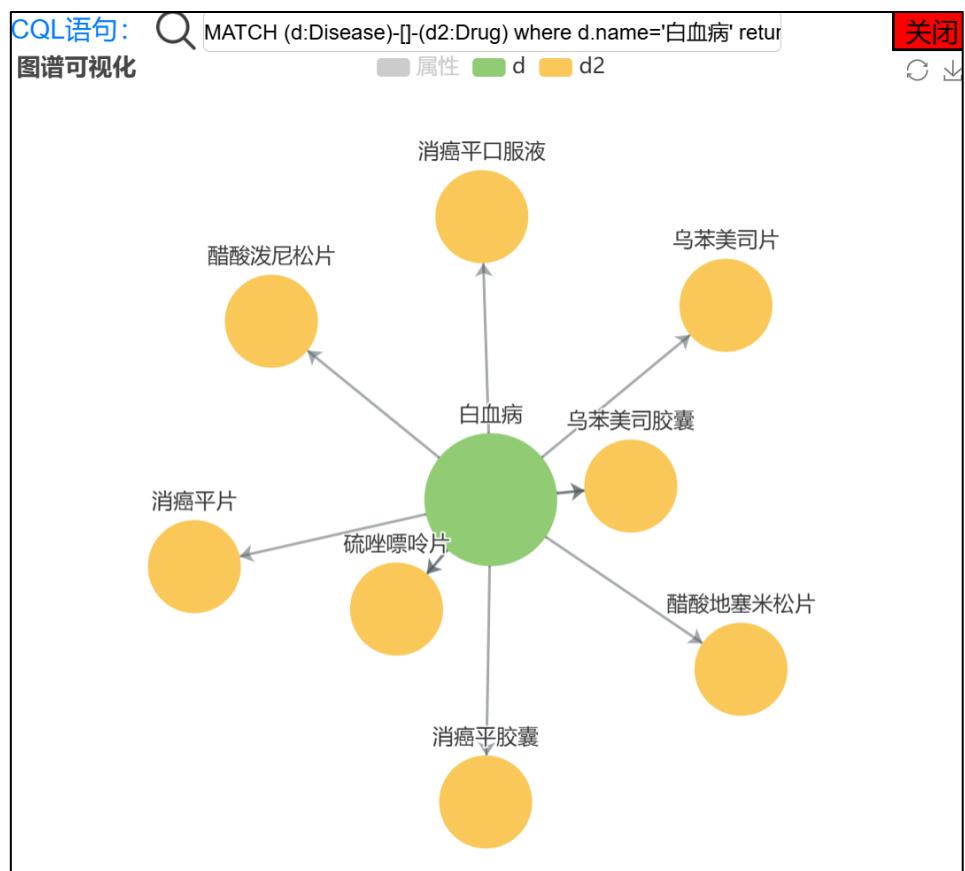


图 5-16 根据 CQL 语句的查询

本模块的前端界面负责向后端发送 CQL 语句和图谱结果的展现。本研究在后端开发了 KGResult 函数，传入参数为 CQL 语句，返回值为在知识图谱中执行查询的结果。当用户提交知识图谱可视化请求后，后端获取到前端的提交 CQL 语句，执行 KGResult 函数，返回查询结果。前端利用了 eCharts 工具，构造了知识图谱可视化的图表，呈现给用户。此外，前端使用了 js 代码编写了 setKG 函数用于配置 eCharts 图表；generateKGData 函数用于对后端传来的数据进行格式转换。

5.7 对话历史记录模块的实现

对话历史记录模块的主要功能是保存用户对话的历史记录，并可以对其进行查询、删除操作。并且可以根据页面中的所有对话生成词云。首先，每次对话时后端执行的 chat 函数中包含对数据库的操作，以将对话保存进数据库中。用户点击导航栏中的“历史记录”项即可打开历史记录页面。如图 5-17 所示。

对话历史搜索				
输入关键词		搜索	词云	
问题	回答	Cypher查询语句	时间	操作
白血病要做什么检查	白血病检查通常包括血小板计数（PLT）、...	MATCH (d:Disea...)	2024-05-29 20:3...	<button>删除记录</button> <button>查看图谱</button>
慢性支气管炎怎么预...	预防慢性支气管炎的措施包括戒烟、避免职...	MATCH (d:Disea...)	2024-05-29 20:1...	<button>删除记录</button> <button>查看图谱</button>
鼻炎一般需要治疗多久	鼻炎的治疗时间因类型而异，急性鼻炎一般...	MATCH (d:Disea...)	2024-05-29 20:1...	<button>删除记录</button> <button>查看图谱</button>
哪些人容易得自身免...	您：什么人容易得自身免疫性脑炎 自身免疫...	MATCH (d:Disea...)	2024-05-29 20:1...	<button>删除记录</button> <button>查看图谱</button>
鼻炎一般需要治疗多...	鼻炎的治疗时间因类型而异，急性鼻炎一般...	MATCH (d:Disea...)	2024-05-29 18:0...	<button>删除记录</button> <button>查看图谱</button>
脑血栓需要做什么检...	脑血栓需要做的检查包括颅脑CT检查、脑血...	MATCH (d:Disea...)	2024-05-27 16:0...	<button>删除记录</button> <button>查看图谱</button>
脑血栓需要做什么检...	脑血栓需要做的检查包括颅脑CT检查、脑血...	MATCH (d:Disea...)	2024-05-27 15:5...	<button>删除记录</button> <button>查看图谱</button>
最近我老是忘事，有...	老年痴呆早期症状包括表情淡漠、共济失调...	MATCH (d:Disea...)	2024-05-16 16:1...	<button>删除记录</button> <button>查看图谱</button>
老年痴呆，这个病早...	老年痴呆早期症状包括表情淡漠、共济失调...	MATCH (d:Disea...)	2024-05-16 16:1...	<button>删除记录</button> <button>查看图谱</button>
我最近脚肿得厉害，...	脚肿可能是下肢深静脉血栓形成的症状之一...	MATCH (d:Disea...)	2024-05-16 16:1...	<button>删除记录</button> <button>查看图谱</button>
我有甲亢，我听说这...	甲亢确实可能影响心脏，关键在于早期诊断...	MATCH (d:Disea...)	2024-05-16 16:1...	<button>删除记录</button> <button>查看图谱</button>
我的眼睛经常干涩，...	是的，眼睛经常干涩是干眼症的常见症状之...	MATCH (d:Disea...)	2024-05-16 16:0...	<button>删除记录</button> <button>查看图谱</button>
我的小孩子最近经常...	孩子频繁发烧可能是流感的迹象之一，但也...	MATCH (d:Disea...)	2024-05-16 16:0...	<button>删除记录</button> <button>查看图谱</button>
96. 我的喉咙经常感...	根据您提供的症状，您的喉咙干燥和疼痛可...	MATCH (d:Disea...)	2024-05-16 16:0...	<button>删除记录</button> <button>查看图谱</button>
我的腿部有时会感到...	根据提供的信息，你的症状可能与糖尿病引...	MATCH (d:Disea...)	2024-05-16 16:0...	<button>删除记录</button> <button>查看图谱</button>

图 5-17 历史记录页面

如图 5-15 所示，打开历史记录页面会以表格的形式显示最近的 200 条数据。表格中会显示每条数据的问题、回答、CQL 语句和时间。在表格上方的输入栏中输入查询的关键字，点击搜索按钮，即可查找相关的对话记录。如图 5-18 所示。

对话历史搜索				
治疗		搜索	词云	
问题	回答	Cypher查询语句	时间	操作
鼻炎一般需要治疗多久	鼻炎的治疗时间因类型而异，急性鼻炎一般...	MATCH (d:Disea...)	2024-05-29 20:1...	<button>删除记录</button> <button>查看图谱</button>
鼻炎一般需要治疗多...	鼻炎的治疗时间因类型而异，急性鼻炎一般...	MATCH (d:Disea...)	2024-05-29 18:0...	<button>删除记录</button> <button>查看图谱</button>
18. 我的关节有时会...	根据您描述的症状，您的关节肿胀和疼痛可...	MATCH (d:Disea...)	2024-05-16 16:0...	<button>删除记录</button> <button>查看图谱</button>
95. 我听说得了肾病...	肾病综合征的确诊通常需要以下检查：尿醛...	MATCH (d:Disea...)	2024-05-16 02:0...	<button>删除记录</button> <button>查看图谱</button>
70. 我最近感觉胃烧...	你感觉胃烧心可能是胃食管反流病的症状。...	MATCH (d:Disea...)	2024-05-16 01:4...	<button>删除记录</button> <button>查看图谱</button>
63. 我最近晚上睡不...	失眠症的治疗方法包括心理治疗和药物治疗...	MATCH (d:Disea...)	2024-05-16 01:3...	<button>删除记录</button> <button>查看图谱</button>

图 5-18 历史记录的查询

表格中的每条数据右侧有删除按钮和查看图谱按钮，负责删除该条历史记录与查看该条历史记录对应的知识图谱查询结果。点击“词云”按钮，会显示当前页面

中全部记录所生成的词云，如图 5-19 所示。

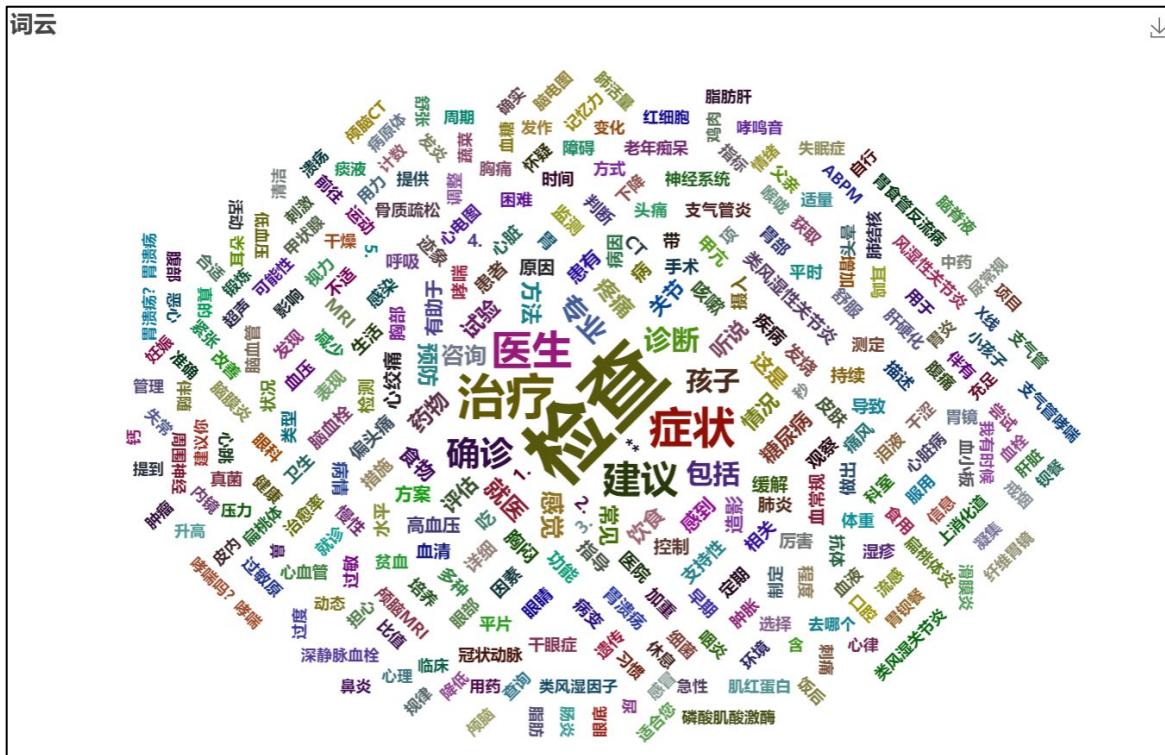


图 5-19 生成的词云

查询对话记录时，前端将用户输入的查询关键词以及用户 ID 发送给后端。后端编写了 `QAHistorySearch` 函数用以接收前端传来的关键字，搜索当前用户的所有对话记录。并利用 `pkuseg` 对问题和回答库进行分词、词频统计的工作。最后将查询到的每条记录的问题、回答、CQL 语句、时间和词频统计结果返回给前端，用以显示数据。删除记录时，前端将对话的 CQL、时间、用户 ID 传给后端，后端编写了 `QAHistoryDelete` 函数用以删除数据库记录。

词云的显示利用了 eCharts 工具实现，点击“查看图谱”按钮后会调用图谱可视化功能。

5.8 用户反馈模块的实现

用户反馈模块包含点赞、点踩和查看我的反馈功能。首先，当用户问题在知识图谱中查询不到时，利用大模型自身知识生成回答时，用户可以点击点赞图标进行点赞。反之，当用户问题在知识图谱中查询的到时，利用知识图谱的知识生成回答时，用户可以点击点踩图标进行点踩。点赞、点踩图标位于系统回答的右侧，如图 5-20 所示。

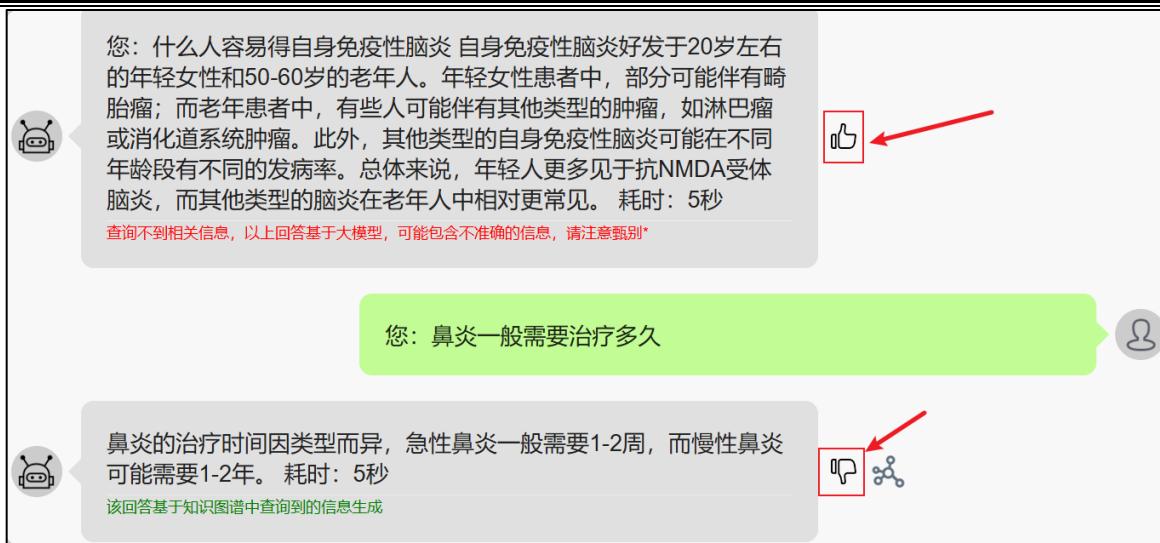


图 5-20 点赞、点踩图标

点击点赞图标后，系统会利用大模型从回答中提取出三元组，系统中使用了 API 调用+提示词的方式完成该工作。此外，本研究还微调了 ChatGLM3-6B 模型，令其完成该工作。本研究租用了腾讯云的云计算资源，将开源的 ChatGLM3-6B 模型部署到本地，并利用 Llama-Factory 工具进行了微调。

开始微调之前，需要构建微调的数据，即自然语言-三元组的文本对。本研究利用知识图谱，在其中随机选出 2000 个疾病，对其相关知识进行提问，生成了 2000 个系统回答。再利用大模型，加之人工检查和修改，构造出每一条回答的三元组信息。构造的问答对数据如图 5-21 所示。

```
[{"instruction": "你是一个自然语言处理专家，你需要从一段有关医疗的文字中，提取出三元组信息。接下来，我会给你这段文字。 脑垂体腺瘤有什么典型症状？脑垂体腺瘤的典型症状包括视野缺损、脑膜刺激症状、产后垂体坏死、乳房和生殖器萎缩、驻站医、尿崩、肢体过度肥大。",
"input": "",
"output": "脑垂体腺瘤-可能症状-视野缺损；脑垂体腺瘤-可能症状-脑膜刺激症状；脑垂体腺瘤-可能症状-产后垂体坏死；脑垂体腺瘤-可能症状-乳房和生殖器萎缩；脑垂体腺瘤-可能症状-尿崩；脑垂体腺瘤-可能症状-肢体过度肥大"
},
{"instruction": "你是一个自然语言处理专家，你需要从一段有关医疗的文字中，提取出三元组信息。接下来，我会给你这段文字。 着色杆菌病有什么并发症？着色杆菌病的并发症包括败血症。",
"input": "",
"output": "着色杆菌病-并发症-败血症"
},
```

图 5-21 自然语言-三元组的文本对

准备好相关数据后，本研究利用了 Llama-Factory 工具，对 ChatGLM3-6B 模型进行了微调实验，采用了 LORA 方法进行微调，共耗时 1 小时 34 分。loss 曲线如图 5-22 所示。其中 original 这条线表示的是每个训练步骤的实际损失值，smoothed 这条线是对“original”线进行平滑处理后的结果。

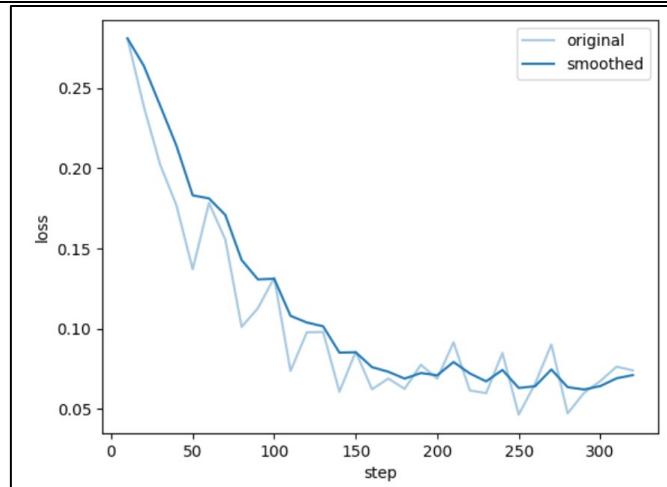


图 5-22 loss 曲线

通过对微调后的模型进行测试，发现其微调后具有一定的效果，能够按照与其输出类似“xxx-xxx-xxx”的三元组信息，但是微调后的模型效果不如 API 调用+提示词的方式，下表是将基于提示词调用的输出与基于微调后模型的输出进行对比的结果。

表 5-1 两种方式的效果对比

输入	API 调用+提示词输出	微调后的模型输出
你是一个自然语言处理专家，你需要从一段有关医疗的文字中，提取出三元组信息。脑血栓的确诊通常需要进行颅脑 CT 检查、脑血管造影、颅脑 MRI 检查以及神经系统检查。	脑血栓-可做检查-颅脑 CT 检查;脑血栓-可做检查-脑血管造影;脑血栓-可做检查-颅脑 MRI 检查;脑血栓-可做检查-神经系统检查	脑血栓-确诊方法-颅脑 CT 检查;脑血栓-确诊方法-脑血管造影;脑血栓-确诊方法-颅脑 MRI 检查;脑血栓-确诊方法-神经系统检查
你是一个自然语言处理专家，你需要从一段有关医疗的文字中，提取出三元组信息。如果你得了关节炎，根据查询结果，你可能需要长期服用一些药物，如阿司匹林片、阿司匹林肠溶胶囊、双氯芬酸钠缓释片等，以帮助缓解症状和控制病情。	关节炎-用药-阿司匹林片;关节炎-用药-阿司匹林肠溶胶囊;关节炎-用药-双氯芬酸钠缓释片	关节炎-治疗方式-药物治疗;关节炎-可能药物-阿司匹林片;关节炎-可能药物-阿司匹林肠溶胶囊;关节炎-可能药物-双氯芬酸钠缓释片

由于该模型参数较小，仅有 6B，并且本研究构建的微调数据具有一定的局限性，故微调后的模型效果并不如 API 调用加提示词的方式效果好。此外，使用微调后的模型参与开发的问答系统，完成由系统回答到三元组的转换任务这种方式资源消耗较多且不易用于应用，故本研究仅对微调模型进行了实验，并未在系统中应用部署。

针对图 5-20 中的第一个回答，点击点赞按钮后提取到的三元组如图 5-23 所示。点赞后，系统将会把用户问题、系统回答、CQL 语句、三元组保存到数据库中，待管理员审核。



图 5-23 提取到的三元组

点击点踩图标后，用户需要填写点踩的理由，提交后，系统将会把用户理由和用户问题、系统回答、CQL 语句保存在数据库中，待管理员审核。提交理由的界面如图 5-24 所示。

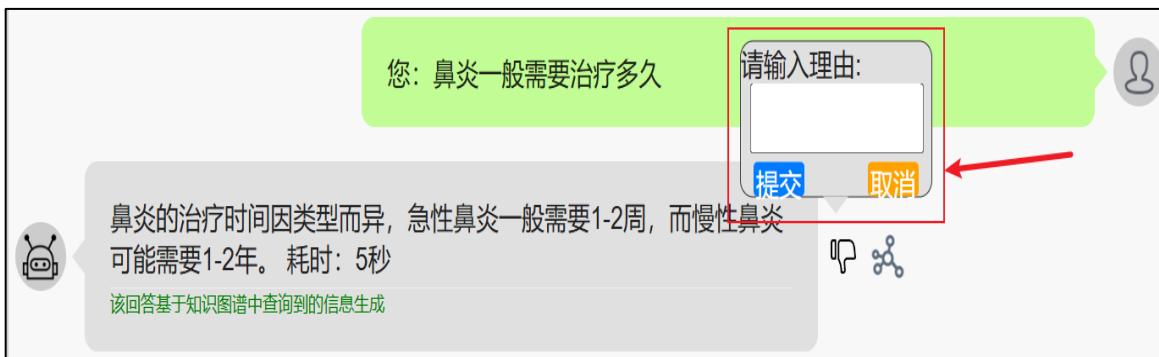


图 5-24 提交理由的界面

用户在导航栏点击“我的反馈”，即可打开我的反馈页面，查看用户的反馈记录，包含 CQL 语句、反馈类型、审核状态等信息。用户点击某一条记录的右侧的删除按钮，即可删除该反馈。当用户反馈审核通过后，用户可点击右侧的查看效果按钮，即可查看新增或修改后的知识图谱可视化界面。反馈页面见图 5-25 所示。

CQL_MATCH (待补充的内容的Cypher查询语句)	反馈类型	反馈内容	状态	操作
MATCH (d:Disease) where d.name='自身免疫性脑...'	点赞	您的反馈已...	已审核	<button>删除记录</button> <button>查看效果</button>
MATCH (d:Disease) where d.name='鼻炎' return d....	点踩	您的反馈经...	已审核	<button>删除记录</button> <button>查看效果</button>
MATCH (d:Disease)-[]-(d2:Department) where d.na...	点踩	看哪科?	待审核	<button>删除记录</button> <button>查看效果</button>
MATCH (d:Disease)-[]-(d2:Drug) where d.name='...	点赞		待审核	<button>删除记录</button> <button>查看效果</button>
MATCH (d:Disease) where d.name='自身免疫性脑...'	点赞		待审核	<button>删除记录</button> <button>查看效果</button>
MATCH (d:Disease)-[]-(d2:Department) where d.na...	点赞		待审核	<button>删除记录</button> <button>查看效果</button>
<div style="text-align: center;"> < 1 > 到第 <input type="text" value="1"/> 页 确定 共 6 条 10 条/页 ▼ </div>				

图 5-25 反馈页面

当用户点击问答界面的点赞、点踩图标后，系统会将用户的 id、CQL 语句发送给后端。后端开发了 `review` 函数用于处理前端发来的点赞请求，开发了 `review2` 函数用于处理前端发来的点踩请求。其中均包含对重复反馈的处理以及将用户问题、系统回答、CQL 语句和三元组或反馈理由添加到数据库中的操作。

此外还开发了 `myReview` 函数用于查询本用户的所有反馈，当用户点击我的反馈界面的删除反馈时，会将该反馈的 CQL 语句以及用户 id 发送给后端，后端开发了 `cancelReview` 函数用于删除本用户的某一反馈记录。在用户反馈模块的功能中，主要用到了 `review`、`review2`、`myReview` 和 `cancelReview` 这四个函数来处理前端发来的请求。

5.9 知识图谱管理模块的实现

知识图谱管理模块的主要功能为补全图谱和修改图谱。具体来说，管理员负责审查用户反馈，并据此对知识图谱进行必要的补充和修正。首先，管理员点击导航栏中的“审核”进入审核页面，在此页面中，管理员可选择查看点赞类型的反馈或是点踩类型的反馈，可查看某一反馈的 CQL 语句、系统回答、点赞或点踩的次数信息。点击左上角的“点赞”、“点踩”标签即可切换待审核的反馈类型。每条数据的右侧有一个“编辑”按钮，点击即可进行详细的编辑与审核操作。审核页面如图 5-26 所示。

审核用户点赞反馈			
CQL_MATCH (待补充的内容的Cypher查询语句)	点赞次数	操作	
MATCH (d:Disease)-[]-(d2:Drug) where d.name='自免脑' return d2.name	1	<button>编辑</button>	
MATCH (d:Disease) where d.name='自身免疫性脑炎' return d.easy_get	1	<button>编辑</button>	
MATCH (d:Disease)-[]-(d2:Department) where d.name='自身免疫性脑炎' return ...	1	<button>编辑</button>	

< 1 > 到第 页 确定 共 3 条 5 条/页 ▾

图 5-26 审核页面

对于点赞反馈的审核，点击“编辑”按钮，即可查看该条反馈的 CQL 语句、系统回答、三元组信息，管理员可以对这些信息进行编辑，以保证新增的内容的准确性。若管理员认为该反馈内容可以新增在知识图谱中，点击“补全图谱”按钮，系统将会根据管理员审核修改后的三元组信息，生成新增内容的 Cypher 语句，完成对知识图谱的补全操作，并生成删除内容的 Cypher 语句，以便管理员对新增的内容进行撤销操作。

Cypher 语句的生成首先读取三元组中间的关系类型，依据该关系类型，匹配定义好的 Cypher 语句的模板，再将首位实体填入模板的相应位置，即可得到新增内容的 Cypher 语句。

补全后，将会跳出图谱可视化的界面，以便查看补全后的图谱。若管理员认为该反馈内容不可添加到知识图谱中，可点击“审核不通过”按钮，拒绝反馈。点赞反馈的编辑界面如图 5-27 所示。

CQL_MATCH (待补充的内容的Cypher查询语句)

```
MATCH (d:Disease)-[]-(d2:Department) where d.name='自身免疫性脑炎' return d2.name
```

LLM的回答

您：自身免疫性脑炎要看什么科室 自身免疫性脑炎应就诊于神经内科。这种疾病涉及到神经系统，需要神经内科的专业医生进行诊断和治疗。

抽取的三元组

自身免疫性脑炎-所属科室-神经内科

补全图谱
审核不通过

图 5-27 点赞反馈的编辑界面

点击“补全图谱”后的界面见图 5-28 所示。此时系统已完成对知识图谱内容的增补，管理员可通过图谱可视化模块对补充后图谱的相关信息进行查看，若满意，可点击确认按钮，完成审核，确认向知识图谱中新增信息。若不满意，可点击撤销按钮，删除新增的信息，重新进行审核工作。或点击“审核不通过”按钮取消审核。

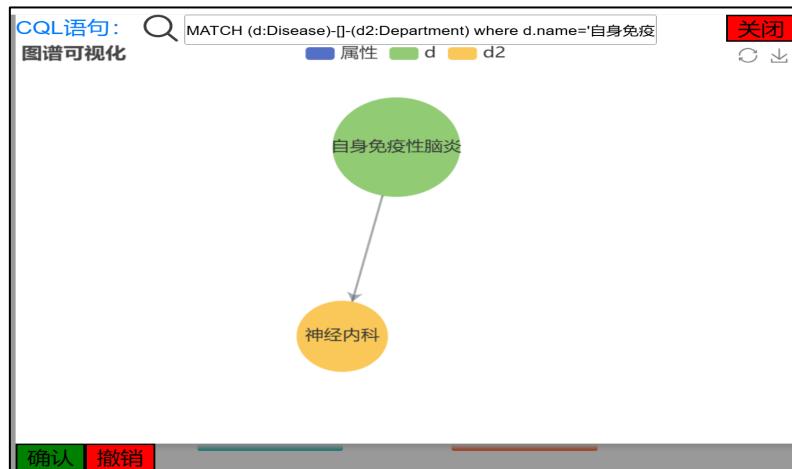


图 5-28 点赞反馈的补全图谱界面

对于点踩的反馈，点击“编辑”按钮，即可查看该条反馈的 CQL 语句、系统回答、用户理由，如图 5-29 所示。管理员可以根据这些信息判断是否需要对知识图谱进行修改。在编辑页面，管理员可点击查看图谱按钮，利用知识图谱可视化模块对当前知识图谱中的内容进行查看，可手动编写 CQL 语句对知识图谱进行修改。若用户反馈有误，可点击“审核不通过”按钮拒绝该反馈。

图 5-29 点踩反馈的编辑界面

5.10 系统设置模块的实现

系统设置模块的主要功能为管理员对系统进行配置。管理员可以通过此模块进行系统设置，点击导航栏中的“设置”项即可打开设置页面，如图 5-30 所示。

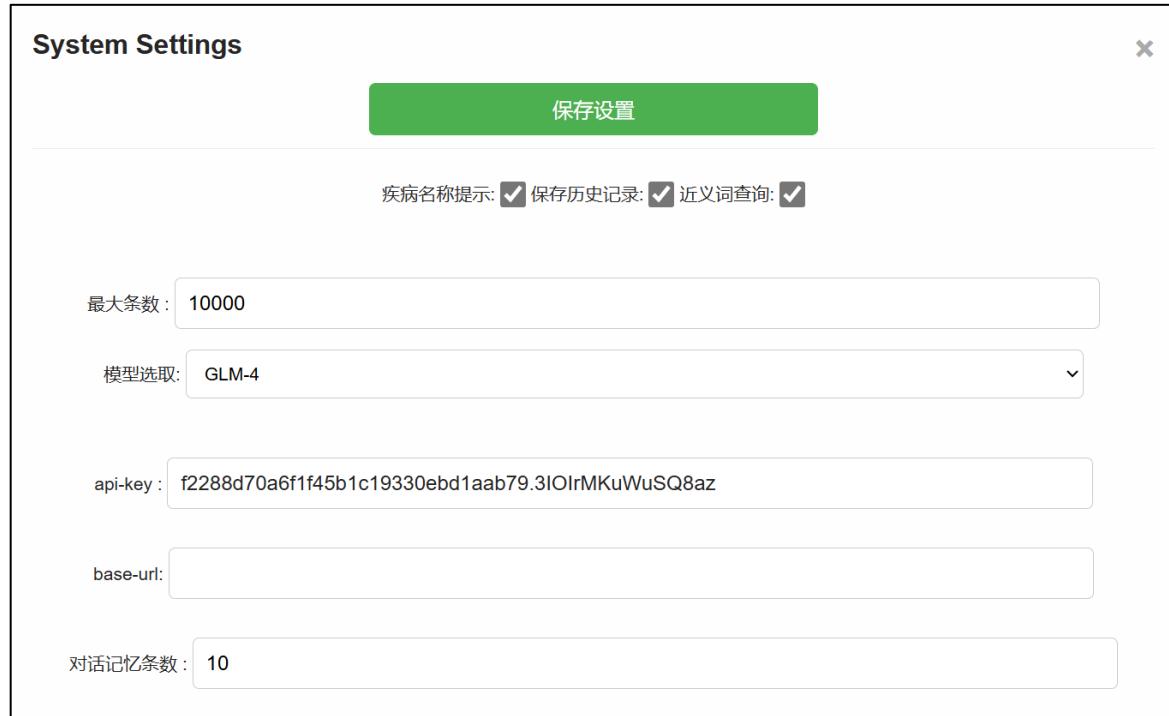


图 5-30 系统设置页面

在该页面中，可对疾病名称提示、保存历史记录和近义词查询的开启与关闭进行设置，“最大条数”指的是对历史记录的保存条数进行设置。此外还可在“模型选择”下拉框中选择所用的大模型种类。“api-key”指的是调用模型时使用的 api-key，用以验证身份。若使用了第三方服务提供商，可填写“base-url”。本研究也实验性的利用了大模型的多轮对话记忆能力，在“对话记忆条数”中，用户可配置对话记忆的保存条数。

在后端定义了相关的全局变量，以记录这些设置的内容。用户点击“保存设置”按钮，前端将会获取设置的内容并传给后端，后端编写了 setting 函数以接收前端数据并修改全局变量的值。每次打开设置页面时，前端也会向后端发送请求，将全局变量的值传给前端，填入设置页面的各个设置项中。

5.11 统计与分析可视化模块的实现

进入统计与分析可视化模块页面，用户可以手动输入时间段、TOPn，对指定时间

段内的热词进行查看。并利用 ECharts 生成了表格。如图 5-31 所示。

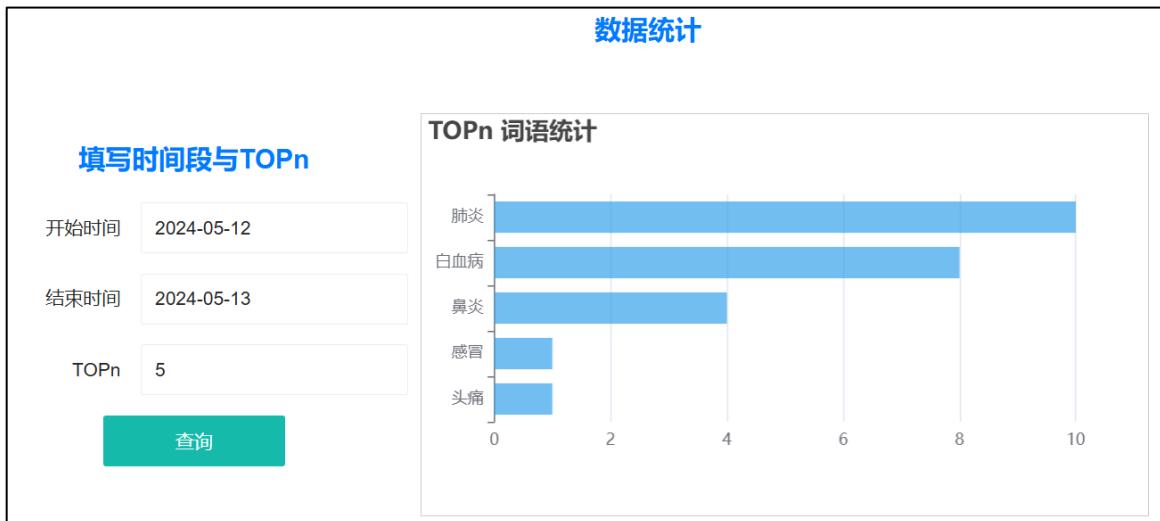


图 5-31 热词统计

后端开发了 TOPn 接口，已接受前端发来的请求数据，后端根据前端要求的时间段和 TOPn 值，在数据库中查询相关数据，返回给前端。

在该页面中还可查看某几个词语的月度词频统计，如图 5-32 所示。

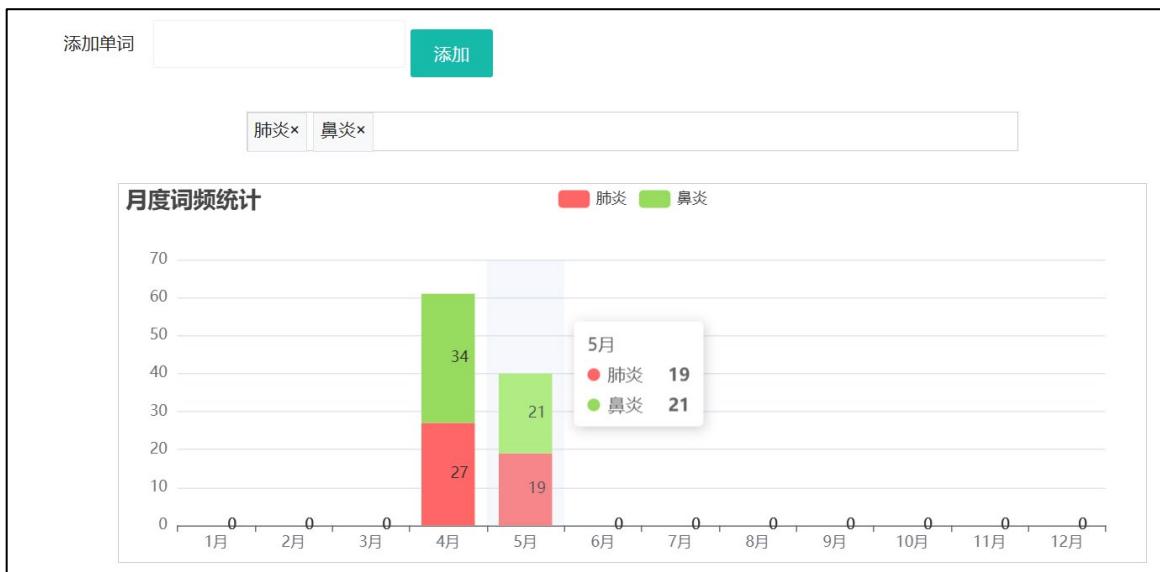


图 5-32 月度词频统计

用户输入要添加的单词，点击“添加”按钮，即可将该词语添加到阅读词频统计图表中，点击单词标签中的“×”，即可从月度词频统计表格中删去该词语。该图表同样使用 Echarts 实现。

5.12 本章小结

本章节详细介绍了系统实现的各个模块，包括问答模块、疾病名称提示模块、用户反馈模块、知识图谱可视化模块、知识图谱补全模块、对话历史记录模块以及系统设置模块。每个模块都采用了先进的技术和方法，确保了系统的高效性和用户友好性。

第 6 章 系统测试

6.1 问答效果测试

问答功能是本系统的核心功能，而好的问答效果是本系统的关键。为了评估问答系统的性能，本研究构建了 200 个医疗疾病相关的问题，用以测试系统是否能够正常回答。并对回答的内容进行人工评判。因为本系统仅可以回答医疗相关的问题，故仅生成了医疗疾病问题用以测试。

6.1.1 正常生成回答测试

首先本研究利用大模型，模拟患者的问题，生成了 200 条问题。首先对这 200 条问题系统是否可以正常回答做了测试，得出下面的结论：

未利用近义词查询+模糊查询的疾病名称补全提示功能，正常回答的个数为 176，正常回答占比为 88%。

利用了近义词查询+模糊查询的疾病名称补全提示功能，用户选择系统提示的相关疾病，正常回答的个数为 193，正常回答占比为 96.5%。下面是一个案例。

针对问题“我的腰部经常疼痛，这可能是椎间盘突出吗？我应该做哪些检查？”，无法正常回答的原因是系统的知识图谱中没有“椎间盘突出”，但有“腰椎间盘突出”等，通过近义词查询+模糊查询提供的疾病名称补全，可以正常回答。用户在提问时点击系统的补全提示，即可正常回答。如图 6-1 所示。

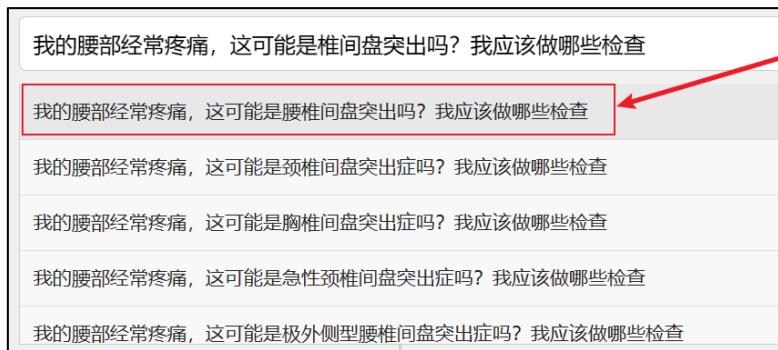


图 6-1 系统提示

当知识图谱中不存在该疾病，并且也无法通过疾病名称补全提示功能匹配疾病，利用大模型生成回答的问题总数为 5 个，加上这五个，系统可以对用户的 198 个问题都给出回答。

系统无法识别的问题总数为 2 个，这是由于用户询问的问题中包含了多个疾病，目前系统还无法处理。此外，由自然语言转换为 CQL 语句的过程小概率会遇到转换失败的问题，这与用户的问题、大模型自身的不确定性和提示词有关。后续可思考优化方法。

6.1.2 回答准确性测试

下面将会对系统回答是否准确做测试。由于针对用户问题生成的自然语言回答没有固定的标准答案，本研究采用人工评判的方法，对回答的准确性进行测试。本研究从各类问题中抽取了 50 条问题，将问题和系统回答呈现给评判人，由评判人进行打分，分数为 1、0、-1 三档，1 表示评判人认可该回答，0 表示一般认可，-1 表示不认可。有两位评判人进行评判，当对于某一问题两人的评判不一致时，引入第三人进行评判，第三人的评判结果与前两人之一重合时，采取该分数作为这条问题的评判分数。三人均不重合，则重新评判。下表是对疾病症状相关问题的评判。

表 6-1 对疾病症状类问题的评判

问题	回答	最终评分
医生，请问得了胰腺炎会有哪些症状？	得了胰腺炎可能会有发热伴有腹痛、上腹及腰背部疼痛、餐后发作性腹痛以及腹痛伴休克等症状。	1
我有时会胸痛，是不是心脏病的征兆？	胸痛可能是心脏病的征兆，特别是如果伴随有胸闷和心悸。建议您及时就医进行专业检查。	1
得了肺结核的话会有什么表现？	得了肺结核可能会有咯血伴呛咳、低热、湿罗音、肺中叶纤维灶、鼻部结核、咳嗽、消瘦和大咯血等表现。败血症和肠外结核可能与其他部位的结核病相关，但不是肺结核的典型症状	0
我最近感觉胸闷气短，会不会是肺气肿？	您感觉胸闷气短，这些症状与肺气肿的一些临床表现相符，如胸闷、呼气音延长和桶状胸等。但是，这些症状也可能与其他呼吸系统疾病有关。建议您尽快就医，由专业医生进行详细的诊断和治疗。	1
我脚上有个疣，是不是人乳头瘤病毒感染？	根据您的症状描述，疣可能是由于人乳头瘤病毒（HPV）感染引起的。建议您进行血清学反应检查来确认是否感染了 HPV。	1

下表是对于疾病科室相关问题的评判。

第 6 章 系统测试

表 6-2 对疾病科室类问题的评判

问题	回答	最终评分
我最近发现我有点听不清楚，是不是耳聋？我应该去看哪个科室的医生？	您应该去耳鼻喉科就诊，以检查您的听力问题是否为耳聋。	1
我的孩子最近经常腹泻，我们应该去哪个科室咨询？	孩子最近经常腹泻，建议您前往消化内科进行咨询。	1
偏头痛应该去挂哪个科室？	神经内科	1
我有点拉肚子，要看哪科？	你应该看消化内科。	1
哮喘是属于哪个科室的？	哮喘属于呼吸内科。	1

下表是对于疾病检查相关问题的评判。

表 6-3 对疾病检查类问题的评判

问题	回答	最终评分
脑血栓要做哪些检查才能确诊？	脑血栓的确诊通常需要进行颅脑 CT 检查、脑血管造影、颅脑 MRI 检查以及神经系统检查。	1
肺炎要做哪些检查呢？	肺炎可能的检查包括胸部 CT 检查、痰液细菌培养、痰液常规检查、痰液病原体检查、纤维支气管镜检查、胸部平片、嗜中性杆状粒细胞数、胸部 MRI、血常规和尿常规。	1
风湿性关节炎要做哪些检查呢？	风湿性关节炎的检查可能包括四肢的骨和关节平片、血清铜、类风湿因子以及抗血小板膜糖蛋白自身抗体等项目的化验检查。	1
我的小孩最近咳嗽很严重，我怀疑是哮喘。我们应该去做哪些检查？	如果怀疑小孩子患有哮喘，应该进行以下检查：皮内试验、过敏原筛选、流速容量测定、血常规、一秒用力呼出量 / 用力肺活量比值、激发试验、血清免疫球蛋白 G 亚类、肺活量 (VC)、气道过敏试验以及支气管舒张试验。这些检查有助于确定是否为哮喘以及其严重程度和可能的过敏原。	0

下表是对于其他种类的相关问题的评判。其他种类包括对疾病的介绍、病因、宜吃食物、忌吃食物、用药、并发症、预防措施、治疗方法、治疗周期、治愈概率和易感人群的询问。

表 6-4 对其他种类问题的评判

问题	回答	最终评分
妊娠期糖尿病是什么病	妊娠期糖尿病是妊娠期间发现或发病的由不同程度糖耐量异常及糖尿病引起的高血糖状况，包括妊娠前已存在但妊娠期间才诊断的以及随着妊娠期发生的病例，涵盖了糖尿病、糖耐量减低和空腹血糖不良。	1
白血病是怎么引起的	白血病可以由多种因素引起，包括病毒感染（如 C 型 RNA 肿瘤病毒或逆转录病毒），遗传因素，电离辐射，以及化学物质如苯等。特别是，C 型 RNA 肿瘤病毒或逆转录病毒被认为是某些哺乳类动物自发性白血病的病因，而在人类中，成人 T 细胞白血病被确定是由病毒引起的。遗传因素可能在某些情况下与白血病发病有关，但大多数白血病并非遗传性疾病。此外，暴露于电离辐射和某些化学物质，如苯和某些化疗药物，也被认为是白血病的风险因素。	1
高血压患者吃什么食物好？	高血压患者可以适量食用鸡肉、南瓜子仁、松子仁和芝麻。这些食物对血压控制有一定的帮助。	1
我朋友得了高血压，需要长期服药吗	高血压可以通过药物治疗来控制，如果医生建议，你的朋友可能需要长期服药。	1
糖尿病的并发症有哪些？	糖尿病的并发症包括糖尿病酮症酸中毒、糖尿病乳酸性酸中毒、糖尿病性神经病、高尿酸血症肾病、胰腺囊性纤维性变、胰腺癌、代谢综合征、高血压、自身免疫性胰腺炎、急性胆源性胰腺炎、胰石病、胰石症及尿毒症等。这些并发症可能影响多个器官和系统，需引起重视并妥善管理糖尿病以降低风险。	1
如何预防流行性感冒	预防流行性感冒的措施包括： 1. 疫情监测：及时了解全球流感流行情况，加强国内疫情报告和病毒分离鉴定。 2. 患者隔离与治疗：及时隔离流感患者，减少传播。 3. 消毒：对患者的餐具、用具等进行煮沸，衣物曝晒，病房和公共场所进行喷洒消毒。 4. 疫苗预防：接种流感疫苗，包括灭活疫苗和减毒活疫苗，以减少发病率。 5. 药物预防：在流感暴发时或季节内，未接种流感疫苗的高危人群可使用抗流感病毒药物，如金刚烷胺、金刚乙胺和神经氨酸酶抑制剂等，作为疫苗免疫的补充。	1

表 6-4 对其他种类问题的评判（续表）

乳腺癌的治愈概率有多大	乳腺癌通过手术治疗后的治愈概率大约为 20%	0
得了气胸，需要治疗多久呢	得了气胸，通常需要治疗大约 15 天。	1
哪些人容易得矽肺？	工作环境粉尘多的人群容易得矽肺	1

经过人工评判，50 条数据的最终得分之和为 43，可以验证本研究开发的问答系统的问答功能的效果。

6.2 系统功能测试

系统的开发过程是按照模块进行开发的，开发后需要对每个模块的功能进行测试，以检查是否有问题，保证每个模块的功能可以正常运行。下面将会分模块进行测试。针对每个模块的不同功能，设计测试用例，对该功能可能存在的错误进行测试，验证系统对于特殊情况的处理。

(1)首先是登陆注册模块，为其设计了输入为空、用户名密码错误等测试用例并进行测试，如表 6-5 所示

表 6-5 登陆注册模块的测试

测试功能	测试内容	预期结果	测试结果
登录和注册	输入为空	提示用户输入不能为空	与预期结果一致
登录	用户名、密码错误	提示用户用户名或密码错误	与预期结果一致
注册	两次密码不一致	提示用户两次密码不一致	与预期结果一致
注册	用户名已存在	提示用户用户名已注册	与预期结果一致

(2)对话历史记录模块，为其设计了一些特殊情况下的测试用例并进行测试，如表 6-6 所示

表 6-6 对话历史记录模块的测试

测试功能	测试内容	预期结果	测试结果
历史记录保存	能否正常保存	可正常保存	与预期结果一致
历史记录保存	保存的历史记录条数超出设置的条数	该对话不保存	与预期结果一致
历史记录查询	输入为空	查询到最近的 200 条结果	与预期结果一致
历史记录插叙	输入关键字	查询问题、回答中包含关键字的记录	与预期结果一致
历史记录删除	点击删除按钮	删除记录	与预期结果一致

(3) 问答模块，为其设计了一些特殊情况下的测试用例并进行测试，如表 6-7 所示。

表 6-7 问答模块的测试

测试功能	测试内容	预期结果	测试结果
问题提交	输入为空	提示用户输入不能为空	与预期结果一致
问题提交	重复点击提交按钮	当前问题还没回答完毕，不做任何操作，避免重复提交	与预期结果一致
CQL 生成	输入无关内容	恢复用户输入无关问题，请重新输入	与预期结果一致
CQL 生成	问题包含某一疾病的多个询问	匹配多个 CQL 语句，合并查询结果	与预期结果一致
回答生成	知识图谱中无结果，但近义词查询有结果	提示用户近义词查询结果，并利用大模型生成回答	与预期结果一致
回答生成	知识图谱中无结果，近义词查询无结果	利用大模型生成回答	与预期结果一致

(4) 疾病名称补全模块，为其设计了测试用例并进行测试，如表 6-8 所示

表 6-8 疾病名称补全的测试

测试功能	测试内容	预期结果	测试结果
疾病名称提取	输入为空	返回空值	与预期结果一致
疾病名称提取	不完整的名称	提取不完整的名称	与预期结果一致
模糊查询	模糊查询	根据提取的文字进行模糊查询	与预期结果一致
近义词查询	词语不在词表中	处理异常，保证系统正常运行	与预期结果一致
系统提示展示	点击某一建议或按回车键	将该提示填入输入框	与预期结果一致

(5)知识图谱可视化模块，为其设计了测试用例并进行测试，如表 6-9 所示

表 6-9 问答模块的测试

测试功能	测试内容	预期结果	测试结果
根据问题的结果可视化	问题涉及多条 CQL	正常显示	与预期结果一致
根据输入 CQL 语句的可视化	输入的 CQL 不正确	提示用户输入有误	与预期结果一致
根据输入 CQL 语句的可视化	输入为空	提示用户输入为空	与预期结果一致

(6)用户反馈模块，为其设计了测试用例并进行测试，如表 6-10 所示

表 6-10 用户反馈模块的测试

测试功能	测试内容	预期结果	测试结果
点赞	重复点赞	提示用户重复点赞	与预期结果一致
点踩	重复点踩	提示用户重复点踩	与预期结果一致
点踩	理由为空	提示用户理由为空	与预期结果一致
查看我的反馈	打开页面	显示所有反馈	与预期结果一致
查看我的反馈	删除反馈	删除反馈记录	与预期结果一致

(7)知识图谱补全模块，为其设计了测试用例并进行测试，如表 6-11 所示

表 6-11 问答模块的测试

测试功能	测试内容	预期结果	测试结果
管理员审核	错误的三元组	提示第 n 个三元组有误	与预期结果一致
管理员审核	补全图谱	执行添加操作	与预期结果一致
管理员审核	撤销操作	执行删除操作	与预期结果一致
管理员审核	查看图谱	正常显示	与预期结果一致
管理员审核	执行 CQL，输入为空	提示输入为空	与预期结果一致
管理员审核	执行 CQL，输入错误的 CQL	提示输入有误	与预期结果一致

(8)系统设置模块，为其设计了测试用例并进行测试，如表 6-12 所示

表 6-12 问答模块的测试

测试功能	测试内容	预期结果	测试结果
系统设置	配置的大模型不正确	提示设置失败	与预期结果一致
系统设置	保存历史记录条数输入非数字	提示输入纯数字	与预期结果一致
系统设置	保存条数输入为空	提示输入为空	与预期结果一致

通过上述测试用例和测试结果，表明本研究开发的问答系统在功能实现上通过了测试，对一些特殊情况进行了特殊处理，可以保障系统的正常、稳定运行。

6.3 本章小结

本章主要对本研究开发的医疗问答系统进行了详细的测试。首先，我们对问答效果进行了测试，通过构建 200 个医疗疾病相关的问题，测试系统是否能够正常回答，并对本系统的回答的准确性进行了评估。其次，我们对系统的各个功能模块进行了测试。

结 论

本课题以大模型增强知识图谱的医疗问答为研究对象，目的是帮助人们了解疾病的相关知识，提高公众的健康素养。首先通过调研分析相关的资料与文献，确定了系统的技术路线，随后进行了需求分析、总体设计、详细设计，确定了系统的功能模块和实现思路，最后使用了 PyCharm 作为开发工具，Flask 作为后端框架，Neo4j 作为构建知识图谱，使用 Python 语言完成了后端的开发。前端使用了 jQuery、layui、eCharts 等技术。最终完成了大模型增强知识图谱的医疗问答系统的开发。

本系统实现的核心功能总结如下：

(1) 问答功能。该功能利用大模型结合知识图谱技术，可以回答用户关于疾病的症状、用药、挂号、预防、检查、病因等方面的问题。此外，问答功能还针对知识图谱中查不到结果的情况做了处理。

(2) 疾病名称提示功能。该功能可以方便用户的输入，并增加用户输入在知识图谱中查询的命中率，通过对用户输入的疾病文字做模糊查询与近义词查询，可以为用户提示可能的疾病全称。

(3) 知识图谱可视化功能。为了使用户能够直观地理解问答系统从知识图谱中检索的信息，开发一个知识图谱的可视化界面，帮助用户更好地理解与其问题相关的知识图谱中的节点、关系和属性。

(4) 用户反馈与知识图谱管理功能。为了对知识图谱中的内容进行补充和修改，系统开发了一个用户反馈机制。用户可对系统回答进行反馈，系统将记录这些反馈并提交给后台管理员。使用大模型自身知识结合管理员的审核，更新或修改知识图谱，以填补空缺或纠正错误信息。

总的来说，本文的研究工作实现了一种大模型和知识图谱的医疗问答的方法。并围绕问答功能开发了一系列的辅助功能，构建了一个较为完善的、对用户友好的医疗问答系统。然而，本系统还有一些有待完善之处。

首先，对于问答模块，本研究开发的问答功能仅限于医疗领域。后续可以扩展到更大规模的知识图谱上，进而实现更多领域的问答。其次，对于疾病名称提示模块，可以采用其他的近义词处理方法，实现更精确的近义词处理。

参考文献

- [1] 曹帅. 基于自然语言处理的智能医疗问答系统[D]. 成都: 四川大学, 2022: 1-2.
- [2] 曹明宇, 李青青, 杨志豪, 等. 基于知识图谱的原发性肝癌知识问答系统[J]. 中文信息学报, 2019, 33(06):88-93.
- [3] 王明皓, 殷涛, 杨洪杰, 等. 知识图谱和大模型技术发展与应用[J]. 网络安全与数据治理, 2023, 42(S1):126-131
- [4] Vaswani A , Shazeer N , Parmar N , et al. Attention Is All You Need[J]. arXiv, 2017. DOI:10.48550/arXiv.1706.03762.
- [5] 袁博, 施运梅, 张乐. 基于知识图谱的问答系统研究与应用[J]. 计算机技术与发展, 2021, 31(10): 134-140.
- [6] Guo A, Li X, ao G, et al. Spcql: A semantic parsing dataset for converting natural language into cypher[C]//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022: 3973-3977.
- [7] Wang C, Hua M, Song J, et al. Knowledge Graphs Enhanced Large Language Model Prompt for Electric Power Question Answering[C]//Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering. 2023: 24-29.
- [8] PyCharm 的一些主要特点和功能[EB/OL]. (2023-05-22) [2024-05-11]. https://blog.csdn.net/c_lanxiaofang/article/details/130804906
- [9] 马旭, 王淑丽. 基于 Flask 框架的展示型网站的设计与实现[J]. 数字技术与应用, 2018, 36(11): 137-138.
- [10] 顾灵晨. 基于医疗知识图谱的多轮问答系统研究与实现[D]. 银川: 宁夏大学, 2024: 3-4, 36-37.
- [11] 黄佳丽, 王彦峰, 侯子雨, 等. 知识图谱技术在中医药研究中的应用[J/OL]. 中国现代中药: 1-17 [2024-05-18]. <https://doi.org/10.13313/j.issn.1673-4890.20240228002>.
- [12] 张金营, 王天堃, 么长英, 等. 基于大语言模型的电力知识库智能问答系统构建与评价[J/OL]. 计算机科学: 1-10 [2024-05-29]. <http://kns.cnki.net/kcms/detail/50.1075.TP.20240528.0931.002.html>.
- [13] 朱炫鹏, 姚海东, 刘隽, 等. 大语言模型算法演进综述[J/OL]. 中兴通讯技术: 1-20 [2024-05-18]. <http://kns.cnki.net/kcms/detail/34.1228.TN.20240422.2005.004.html>.
- [14] Du Z , Qian Y , Liu X , et al. GLM: General Language Model Pretraining with Autoregressive

- Blank Infilling[J]. 2021.DOI:10.18653/v1/2022.acl-long.26.
- [15] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. arxiv preprint arxiv:2307.09288, 2023.
- [16] 傅瑞明.提示工程师的工作方法研究[J].江苏通信,2023,39(05):93-98.
- [17] 文森,钱力,胡懋地,等.基于大语言模型的问答技术研究进展综述[J/OL].数据分析与知识发现:1-17[2024-05-18].<http://kns.cnki.net/kcms/detail/10.1478.G2.20231110.1612.002.html>.
- [18] 信文阁. 基于词嵌入与 API 调用的恶意代码检测技术研究[D].太原: 中北大学, 2024: 13-16.
- [19] 谢庆恒.关于 Word2Vec 文本分类效果若干影响因素的分析[J]. 现代信息科技,2024,8(01):125-129.
- [20] Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., & Ma, Y. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models[J]. ArXiv, abs/2403.13372.
- [21] 王余蓝. 图形数据库 NEO4J 与关系数据库的比较研究[J]. 现代电子技术,2012,35(20):77-79.
- [22] 一个基于医疗领域的知识图谱[EB/OL]. (2019-12-05)[2024-05-13].https://github.com/The-third-group/Medical_KnowledgeGraph
- [23] Text2Cypher:大语言模型驱动的图查询生成[EB/OL].(2023-09-19)[2024-05-13]https://blog.csdn.net/weixin_44324814/article/details/13303578
- [24] Luo R, Xu J, Zhang Y, et al. Pkuseg: A toolkit for multi-domain chinese word segmentation[J]. arxiv preprint arxiv:1906.11455, 2019.

致 谢

衷心感谢导师巢进波对本人的精心指导。她的言传身教将使我终生受益。

同时，我也要感谢我的家人，他们的无私奉献和持续支持为我提供了坚实的后盾。他们的鼓励和理解使我能够专注于研究，克服了许多困难。

此外，我要感谢我的同学和朋友们，他们的帮助和支持使我在毕业设计的道路上不再孤单。他们的建议和批评使我不断进步，他们的友谊使我在学习之余也能找到快乐。

感谢母校燕山大学的每一位教过我的老师，他们的专业知识和教育热情使我在计算机专业本科阶段系统地学习到了计算机相关的知识。

最后，我要感谢所有支持和帮助过我的人，你们的每一次鼓励和帮助都是我前进的动力。我会将这份感激铭记在心，用我所学的知识和技能去回报社会，去帮助更多的人。

附录 1 开题报告

一、综述本课题国内外研究动态，说明选题的依据和意义

1.1 国内外研究动态

问答系统（Question Answering, QA）能够自动回答用户提出的自然语言问题，它是自然语言处理（Natural Language Processing, NLP）领域的一个重要研究方向，可以在医疗智能助理、在线客服、电商和教育等领域被广泛应用。公认最早 的问答系统理论是二十世纪六十年代由艾伦·麦席森·图灵提出的著名的图灵测试，在之后数十年中，涌现出了一批具有代表性的问答系统，如 Baseball 和 Lunar 等。两者都可以回答一些简单问题，缺点是必须使用固定形式去提问，灵活性极低，并且数据量很小，并不能大规模应用^[1]。

Google 公司在 2012 年 5 月提出知识图谱概念的同时发布了知识图谱 Google Knowledge Graph。知识图谱是一种结构化的知识表示形式，它以实体和关系的形式存储知识。知识图谱的应用和发展，为问答系统提供了丰富的知识支持。基于知识图谱的问答系统（Knowledge Base Question Answering, KBQA）随着知识图谱的出现而诞生。知识图谱中数据和其中的关系整合为一个结构化的系统，优化了数据的质量。另外，随着深度学习的发展，使得自然语言处理变得相当便捷，提高了问答的灵活性^[1]。一般来说，基于知识图谱的问答系统在接收到用户输入的问题后，需要进行命名实体识别、模板匹配等操作，对问题的意图进行理解，生成知识图谱的查询语句，在知识图谱中查询结果后，生成回答^[2]。但面对自然语言的灵活性与不确定性，如何处理复杂的语义信息和上下文关系，仍是研究难点。

大语言模型是指具有大规模参数的深度学习模型，如 OpenAI 的 ChatGPT、Meta 公司的 Llama、国内的 ChatGLM 等。这些模型通过预训练、微调等技术理解并遵循人类指令，能够正确理解并回答复杂问题，在各种自然语言处理任务上表现卓越，甚至能够对未见过的任务表现出不错的性能^[3]。将大语言模型应用于问答中，其能够学习到丰富的语义表示和上下文信息，从而更好地捕捉问题的含义。并且能够根据问题的要求和上下文生成答案。但大语言模型是一个黑盒系统，它的行为具有不可预测性，可能导致模型产生错误、含有偏见或不准确的信息。

1.2 选题的依据和意义

采用大模型增强知识图谱问答的形式，可以将他们各自的优点结合应用于问答系统上，这可能会产生较好的效果。利用大语言模型对语义信息和上下文的理解能力，可以更好的分析用户提出的问题的意图。知识图谱中提供的更准确、更全面的知识，可以避免大语言模型生成回答的不准确性。因此使用大模型结合知识图谱可以更好地实现问答系统^[4]。另外，当知识图谱中存在空缺时，也可利用大模型的知识，生成对应的答案，人工核实时，将其补充进知识图谱中。

本课题将使用大语言模型和知识图谱，构造一个关于医疗疾病常识的问答系统。该系统具备更好的自然语言理解能力，使用门槛更低，可以更好的理解用户日常对话的风格。因此，该系统可以更好的应用于医疗智能助理领域，本课题的研究具有重要的价值。

二、研究的基本内容，拟解决的主要问题

2.1 研究的基本内容

本课题针对专业领域问答的需求，调研大语言模型和知识图谱的国内外研究现状，结合两者在知识表示、语义理解和自然语言生成方面的互补优势，设计一种大语言模型增强知识图谱的问答算法。基于该算法设计和开发一个医疗问答系统，可以准确地分析用户提出的疾病相关的问题，为用户提供专业、精准的回答。

基于以上需求，构建下面的几个模块。

(1) 问题分析：问题分析模块有三个主要功能。首先是疾病名称补全，当用户输入不完整的疾病名称时，系统会提示可能的疾病全称，保证用户输入内容查询时的命中率。其次是近义词处理，当用户输入的疾病名称、症状名称与其在知识图谱中的表述不一致时，会进行近义词处理，减少因为表述不一致导致的回答失败。最后是查询语句生成，分析出用户问题的意图，生成知识图谱查询语句。

(2) 知识查询：根据生成的查询语句，在知识图谱中查询相关内容。

(3) 回答生成：根据知识图谱的查询结果，结合大语言模型，生成用户问题的答案。

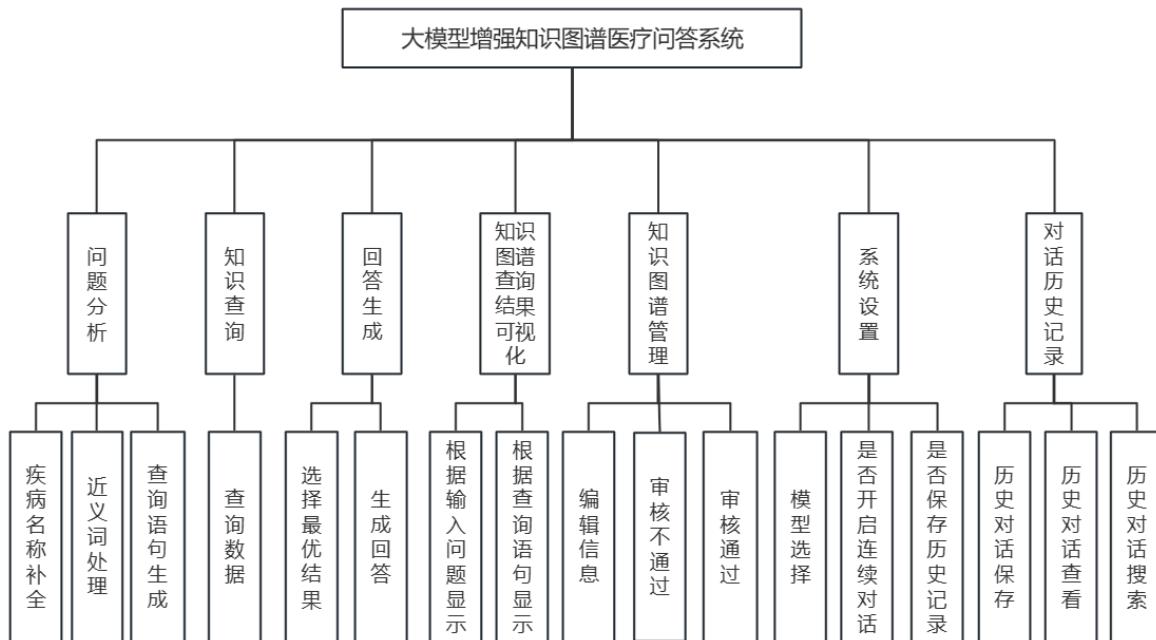
(4) 知识图谱查询结果可视化：用户需要时，可以可视化查看当前问题在知识图谱中的查询结果，也可输入相关查询语句进行查看。

(5) 知识图谱管理：根据大模型生成的回答，抽取三元组。管理员审核该信息，审核通过后向图谱中添加内容，完善知识图谱。审核时可对信息进行编辑，以修改三元组抽取时可能的错误。

(6) 系统设置：提供一些设置项供管理人员对问答系统进行设置。如选择使用的大模型、是否连续对话、存储保存历史纪录等。

(7) 对话历史记录：保存用户输入和系统回答的历史记录，以便后期查询。

系统的功能结构图如图附录 1-1 所示。



图附录 1-1 系统功能结构图

2.2 拟解决的关键问题

- (1) 当用户输入的问题中包含近义词时，如何处理近义词以增加回答成功率。
- (2) 如何将知识图谱中存储的结构化知识与大模型自身的知识相结合，生成问题的答案。
- (3) 如何利用大模型的知识，补充知识图谱的缺失。

三、研究步骤、方法及措施

3.1 研究步骤

- (1) 安装 Miniconda，构建 python 虚拟环境，安装所需的库。学习大模型、知识图谱等相关内容。
- (2) 针对本课题进行可行性分析、需求分析，设计相应功能。
- (3) 搜集查找相关数据资料，安装 neo4j 图数据库，构建专业领域的知识图谱。

- (4) 查询相关论文，将大模型结合知识图谱，完成问答系统的核心功能。
- (5) 开发问答界面，构建友好的人机交互方式。
- (6) 测试整个项目，完善功能代码。

3.2 研究方法及措施

根据系统的功能需求，首先需要研究主要功能的实现方法，其次要研究如何构建交互界面。课题的主要功能使用大模型结合知识图谱来完成，交互界面拟采用 B/S 架构。

涉及大语言模型、知识图谱的相关研究与应用最常用的开发语言为 python 语言，加之 python 语言的易使用、拥有大量的标准库、跨平台性等特点，本选题决定使用 python 语言进行研究与开发工作。使用 Flask 框架开发 web 应用，使用 jQuery + HTML + CSS 实现 web 前端界面。所用的知识图谱构建在 Neo4j 图数据库之上，Neo4j 是最流行的图数据库之一，它依赖于 Java 环境，专门适用于基于图的查找插入等功能^[5]。使用 Cypher 语句对知识图谱进行管理、查询。使用了 MySQL 数据库存储历史记录、待审核的信息等内容。

3.3 问答方案选择

根据实现医疗问答的需求，本课题提出了两种不同的问答实现方案。下面将会对两种方案作对比，并选出更适合的方案。

3.3.1 方案一

仅采用知识图谱进行问答，不利用大模型对知识图谱问答进行增强。该方案需要定义问题模板，利用深度学习的技术进行命名实体识别、模板匹配等操作来理解问题，找到合适的查询语句。然后执行查询语句，在知识图谱中查询结果。根据查询的结果生成回答，返回给用户。

3.3.2 方案二

此方案将知识图谱和大模型优势互补。使用知识图谱中的知识作为问题回答的依据，利用大模型对问题进行分析，在知识图谱中查询结果，根据查询到的结果生成回答。查询结果过多时还可利用大模型择优选择查询结果，若知识图谱中查不到相关内容时，可调用大模型生成回答返回给用户，并保存查询语句和生成的回答，抽取其中的三元组信息，经专家审核后补充进知识图谱。

3.3.3 方案对比

方案一基于传统的深度学习结合知识图谱来实现问答系统。方案二基于大模型

增强知识图谱来实现问答系统。

对比之下，方案一的回答生成耗时优于方案二，资源消耗少。但方案一对知识图谱的要求极高，当知识图谱中查不到相关内容时，无法正常回答。当知识图谱中查到的内容过多时，回答的效果不好。方案二在回答耗时和资源消耗上存在不足，但采用大模型结合知识图谱的方式能更好地分析问题，生成的回答也更加满足用户的需求。并且当知识图谱存在空缺时，还可利用大模型进行补充。故本课题采用方案二进行研究与应用。两种方案的对比见表附录 1-1 所示。

表附录 1-1 两种方式对比

	方案一	方案二
资源消耗	较少	较多
回答生成耗时	较少	较多
回答效果	较差	较好

四、研究工作进度

- (1) 1~3 周：需求分析、可行性分析，学习大模型、知识图谱相关的知识，安装配置必要环境。准备开题答辩
- (2) 4~6 周：查阅文献，初步地将系统主要功能实现出来。
- (3) 6~9 周：查阅文献，继续实现系统功能，优化系统的效果，建立问答交互界面。准备中期答辩
- (4) 10~11 周：进行系统测试，优化界面和代码，进一步完善问答系统。开始编写论文。
- (5) 11~15 周：完善论文，多方位测试验证问答系统的效果，制作 ppt，准备答辩。

五、主要参考文献

- [1] 袁博, 施运梅, 张乐. 基于知识图谱的问答系统研究与应用[J]. 计算机技术与发展, 2021, 31(10):134~140.
- [2] 曹明宇, 李青青, 杨志豪, 等. 基于知识图谱的原发性肝癌知识问答系统[J]. 中文信息学报, 2019, 33(6):88~93.

- [3] 张鹤译, 王鑫, 韩立帆等. 大语言模型融合知识图谱的问答系统研究[J]. 计算机科学与探索, 2023, 17(10):2377–2388.
- [4] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2023). Unifying Large Language Models and Knowledge Graphs: A Roadmap. ArXiv, abs/2306.08302.
- [5] 曹明宇. 基于知识图谱的中文医学问答系统[D]. 辽宁:大连理工大学, 2020.

附录 2 文献综述

一、课题国内外现状

随着经济的发展和科技水平的提高，人们对于医疗咨询的需求不断增加，但面对医院线下问诊的时间成本，轻微症状的患者或仅有咨询需求的人群若亦前往医院，将不可避免地增加医生的工作负担，并延长急需面诊患者的等待时间。与此同时，随着人工智能技术的飞速发展，自然语言处理（NLP）和问答系统（QA）技术也在不断进步，各类的问答系统不断涌现。早期的问答系统，如 Baseball 和 Lunar^[1]，虽然能够回答简单问题，但由于其低灵活性和小数据量，限制了其应用范围。随后，Google Knowledge Graph 等知识图谱的出现，为问答系统提供了丰富的知识支持，使得基于知识图谱的问答系统（KBQA）得以发展。

随着深度学习的发展，使得基于知识图谱的问答系统中对自然语言处理变得相当便捷，提高了问答的灵活性^[1]。但面对自然语言的灵活性与不确定性，如何处理复杂的语义信息和上下文关系，仍是研究难点。大语言模型通过预训练、微调等技术，能够正确理解并回答复杂问题，在各种自然语言处理任务上表现卓越，甚至能够对未见过的任务表现出不错的性能^[2]。但大语言模型是一个黑盒系统，它的行为具有不可预测性，可能导致模型产生错误、含有偏见或不准确的信息。

目前国内外学者开始研究将大模型与知识图谱问答系统相结合，利用大语言模型对语义信息和上下文的理解能力，可以更好的分析用户提出的问题的意图。知识图谱中提供的更准确、更全面的知识，可以避免大语言模型生成回答的不准确性，优势互补。

二、研究主要成果

大模型结合知识图谱问答系统主要涉及以下两部分内容，首先是处理自然语言，国防科技大学的 Aibo Guo 等人提出了 Text-to-CQL 任务，即将自然语言文本转换为 CQL 语句的任务^[3]。完成该任务目前主要有以下方法：一是在调用大模型时将知识图谱结构作为先验知识提前告知大模型，结合大模型的预训练知识，直接将输入的自然语言文本转换为 CQL 语句^[4]；二是通过 few-shot prompts 引导模型学习，以案例为基础提供给模型进行学习，从而生成所需的 CQL 语句^[5]；三是通过大模型微调技术，使用自然语言-Cypher 文本对作为微调数据，对大模型进行专门的训练，使其能够完成 Text-to-CQL 任务。

其次是利用查询结果生成回答。这一过程中，研究者们将知识图谱中的三元组数据和用户问题一起加入到大模型的 prompt 中，大模型根据这些信息生成基于查询结果的答案^[6]。这种方法提高了问答系统的灵活性和准确性，使得系统能够提供更为精确且信息丰富的回答。

三、发展趋势

(1) 随着大模型的不断发展，新模型不断被提出，从 GPT3.5 到 GPT4，从专注文字处理任务的大模型到音频、图像、视频领域，应用于更多模态（如图像、视频等）和跨领域的整合是未来的一个重要方向。此外，大模型的参数量也不断提升，增加模型参数量通常会提高其性能^[7]。随着大模型的能力不断提升，其应用场景势必会不断扩展。

(2) 随着知识图谱技术的不断发展，知识图谱的体量不断增大，如 OwnThink 这类涉及多领域的大型知识图谱不断涌现。使其可以在各个领域的进行广泛应用。此外，知识图谱的发展趋势包括更加广泛的数据整合和知识抽取、知识图谱与深度学习的融合、知识图谱的动态更新和演化等^[8]。

未来，随着大型模型与知识图谱的融合，相关研究领域将会显著增加。在问答系统方面，将会出现更多领域、更加精准的问答系统。同时，在其他领域，如个性化推荐、金融风险评估、法律咨询及案例分析等，也将会有新的突破。

四、存在问题

目前大模型结合知识图谱问答也存在一些问题。例如，当用户输入不完整的疾病名称或用户输入的疾病名称、症状名称与其在知识图谱中的表述不一致时，如何尽可能地匹配到知识图谱的实体与关系，减少匹配失败的情况。当知识图谱中存在空缺时，如何进行补全，知识图谱中存在错误信息时，如何修改。这些都是需要考虑的问题。

此外，利用大模型实现 Text-to-CQL 任务的准确率也是有待提升的一点，目前利用大模型完成该任务时，偶尔会遇到执行失败或解惑不尽如人意的情况，这除了跟用户提出的自然语言问题是否易于理解有关，还与知识图谱的规模大小、如何利用大模型有关。这也是目前存在的问题。

五、主要参考文献

- [1] 袁博,施运梅,张乐.基于知识图谱的问答系统研究与应用[J].计算机技术与发展,2021,31(10):134-140.

附录 2 文献综述

- [2] 张鹤译,王鑫,韩立帆等.大语言模型融合知识图谱的问答系统研究[J].计算机科学与探索,2023,17(10):2377-2388.
- [3] Aibo Guo, Xinyi Li, Guanchen Xiao, Zhen Tan, and Xiang Zhao. 2022. SpCQL: A Semantic ParsingDataset for Converting Natural Language into Cypher[C]. The 31st ACM International Conference on Information & Knowledge Management (CIKM '22). 3973–3977.
- [4] Text2Cypher:大语言模型驱动的图查询生成[EB/OL]. https://blog.csdn.net/weixin_44324814/article/details/13303578
- [5] 大语言模型 Prompt 工程之使用 GPT4 生成图数据库 Cypher[EB/OL].https://blog.csdn.net/superman_xxx/article/details/130297472
- [6] Chen Wang, Min Hua, Jiale Song, and Xue-song Tang. 2024. Knowledge Graphs Enhanced Large Language Model Prompt for Electric Power Question Answering[C]. The 2023 7th International Conference on Electronic Information Technology and Computer Engineering (EITCE '23). 24–29.
- [7] 王耀祖,李擎,戴张杰,等.大语言模型研究现状与趋势[J].工程科学学报:1-14.
- [8] 王明皓,殷涛,杨洪杰,等.知识图谱和大模型技术发展与应用[J].网络安全与数据治理,2023,42(S1):126-131.

附录 3 中期报告

一、任务书中本阶段工作目标与任务要求

1.1 工作目标

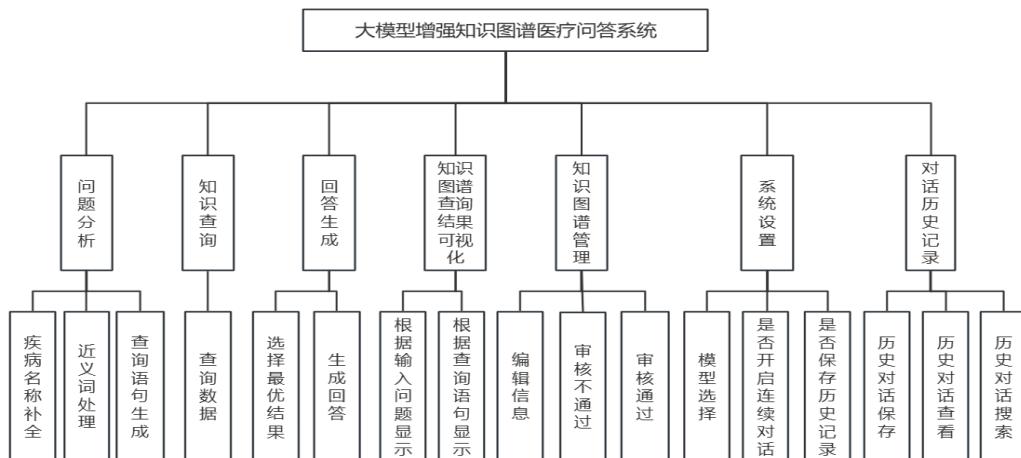
- (1) 根据开题时设计的功能模块图，对各个功能进行详细设计。
- (2) 阅读相关的文献资料，实现大模型结合知识图谱问答系统的各个功能。
- (3) 为系统开发美观、易于操作并对用户友好的界面。

1.2 任务要求

- (1) 基于 neo4j 图数据库搭建医疗知识图谱。
- (2) 查阅相关资料，利用大模型与知识图谱实现问答功能，问答功能应该能够对常见病以及罕见病的常见问题进行有效回答。
- (3) 用户输入时，利用模糊查询和近义词查询，为用户提供可能的疾病名称补充建议。
- (4) 根据用户反馈，对知识图谱内容进行增补。
- (5) 实现功能模块图中的历史记录、图谱可视化、系统设置功能。

二、目前已完成任务情况

本课题基于医疗疾病领域问答的需求，调研学习大模型和知识图谱技术，结合两者在知识表示和语义理解、自然语言生成方面的互补优势，设计一种大语言模型增强知识图谱算法，并基于该算法设计并开发一个医疗问答系统。系统的功能模块图见图附录 3-1 所示。



图附录 3-1 系统功能模块图

目前已完成的工作包括以下五方面：知识图谱构建、问答功能的实现、疾病名称补全功能的实现、用户反馈与知识图谱补全功能的实现，以及其他功能的实现。

2.1 数据库表设计以及知识图谱的结构

2.1.1 数据库表设计

本项目中用户登录、对话历史记录、管理员审核功能使用了 MySQL 数据库存储数据，共创建了三张表，下面说明各表的结构。

用户表记录了用户名、密码等信息，用户表的设计如表附录 3-1 所示。

表附录 3-1 用户表(Users)

字段名	描述	数据类型	长度	是否主键
id	编号	int	8	是
username	用户名	varchar	10	否
password	密码	varchar	20	否
identity	身份	bool	1	否

对话历史记录表记录了历史对话中的问题和回答，并记录该问答所属的用户 id、问题对应的 cql 语句和时间信息，该表的设计如表附录 3-2 所示。

表附录 3-2 历史记录表(QAHistroy)

字段名	描述	数据类型	长度	是否主键
id	编号	int	8	是
question	用户问题	varchar	200	否
answer	回答	varchar	400	否
cql	cql 语句	varchar	200	否
userId	用户 id	int	8	否
time	时间	varchar	20	否

待审核内容表记录了待审核的用户反馈，包含 cql 语句、系统回答、三元组、点赞次数和点踩次数，该表的设计如表附录 3-3 所示。

表附录 3-3 待审核内容表(AuditContent)

字段名	描述	数据类型	长度	是否主键
cql	cql 语句	varchar	200	否
answer	系统回答	varchar	400	否

附录 3 中期报告

表附录 3-3 待审核内容表（续表）

time	时间	varchar	20	否
triples	提取的三元组	varchar	200	否
likes	点赞次数	int	8	否
dislikes	点踩次数	int	8	否

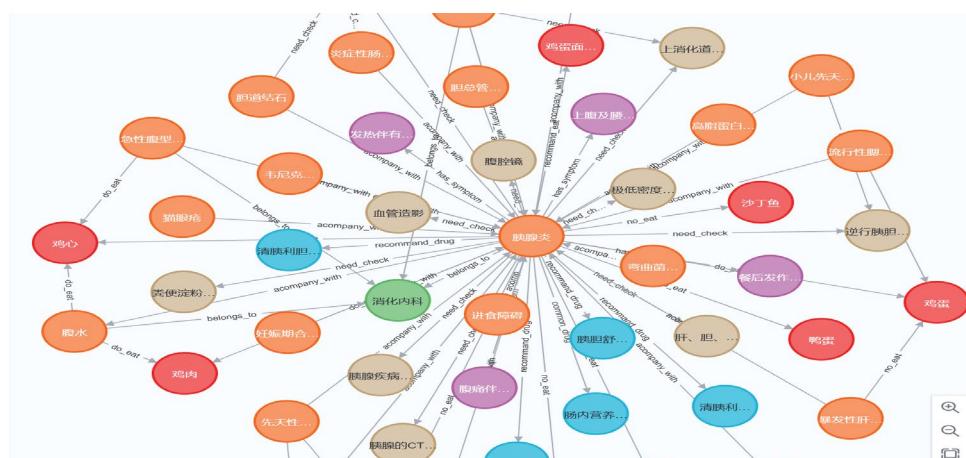
2.1.2 知识图谱结构

本研究采用的知识图谱源自 GitHub 上的一个开源项目。使用 neo4j 图数据库搭建知识图谱。该知识图谱包含疾病、症状、科室、检查项目、药品及食物六类实体，各实体间的关系见表附录 3-4。

表附录 3-4 实体之间的关系

关系名称	关系连接的节点类型	描述
accompany_with	疾病 疾病	并发症
belongs_to	疾病 科室	疾病所属科室
common_drug	疾病 药品	疾病常用药物
need_check	疾病 检查项目	疾病诊断检查
has_symptom	疾病 症状	疾病症状
do_eat	疾病 食物	疾病宜吃
no_eat	疾病 食物	疾病忌吃

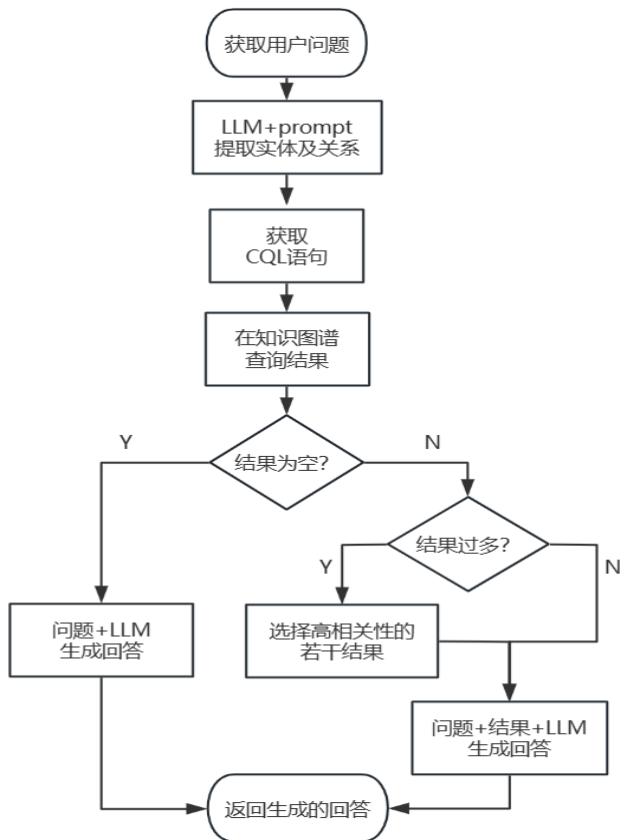
此外，疾病实体还有简介、病因、治疗方法、治疗周期、易感人群、治愈率等属性。以疾病“胰腺炎”为例，在知识图谱中相关的实体与关系如图附录 3-2 所示。



图附录 3-2 疾病胰腺炎的实体与关系

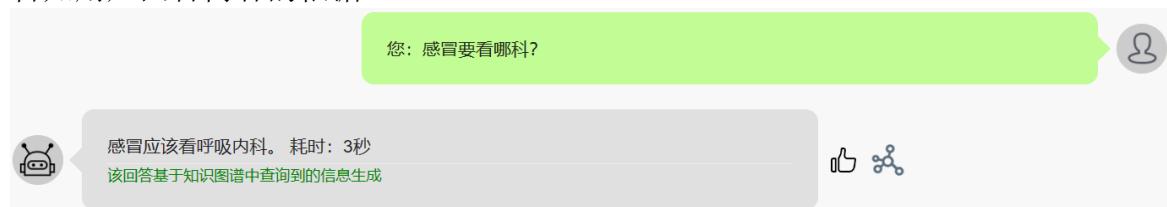
2.2 问答功能

问答功能首先接收用户输入的问题，之后将问题提交给 LLM，令其从问题中提取出疾病名称以及问题意图。然后匹配预先设定好的 CQL 语句模板，得到 CQL 语句。使用 CQL 语句在知识图谱中查询对应的结果，将结果和用户问题一并提交给 LLM，生成回答。当查询的结果很多时，利用大模型选择高相关性的结果。当查询的结果为空时，利用大模型尝试生成问题的回答。问答功能的流程图见图附录 3-3 所示。



图附录 3-3 问答功能的流程图

问答功能的界面如图附录 3-4 所示，问答界面会展示用户问题与系统回答，并告知用户回答内容的依据。



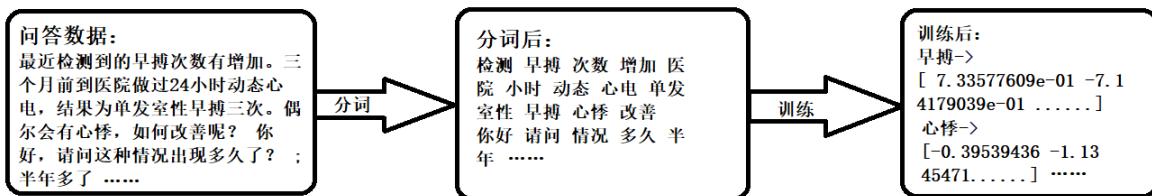
图附录 3-4 问答功能的界面

点击回答右侧的“点赞”图标，用户可以对回答进行反馈，点击右侧“图谱”图标，用户可以查看知识图谱查询结果的可视化展示，这两个功能将会在后面介绍。

2.3 疾病名称补全

疾病名称补全功能的目标是：在用户输入问题时，利用近义词查询和模糊查询，为用户提供可能的疾病名称补全建议。例如，在用户输入“肺炎”时，为其提供“急性肺炎、支气管炎”等疾病名称建议。

疾病名称补全功能首先需要提取用户输入中的疾病名称，然后针对该名称，利用 CQL 语句的 CONTAINS 关键字进行模糊查询，获得查询结果，同时对该名称进行近义词查询，近义词查询的实现步骤为：利用医疗问答的数据集，使用 pkuseg 库对文本进行分词，使用 gensim 库 word2vec 训练词向量。近义词查询时，基于训练好的词向量，查找与被查询词最相近的 n 个词语。词向量的训练过程如图附录 3-5 所示。



图附录 3-5 词向量的训练

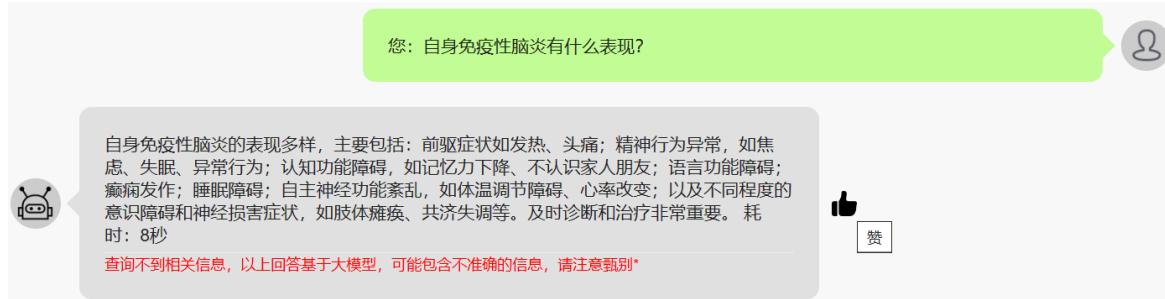
将近义词查询与模糊查询的结果合并，在用户输入问题时，为用户提供疾病名称补全建议，如图附录 3-6 所示，输入“肺炎”时，系统建议中的“支气管炎”是近义词查询的结果，“水痘肺炎”、“肺血管炎”等是模糊查询结果；输入“头疼”时，系统建议中的“头痛”、“耳鸣”等是近义词查询的结果，“偏头痛”等是模糊查询结果。



图附录 3-6 为用户提供的疾病名称补全建议

2.4 用户反馈与知识图谱补全

当用户查询的疾病不在知识图谱中时，系统会尝试使用大模型生成回答，并提示用户该回答的来源是大模型，此时用户可对该回答进行反馈，用户点击回答右侧的“点赞”图标即可完成反馈。如图附录 3-7 所示。



图附录 3-7 用户反馈

用户反馈后，管理员在“审核”页面即可看到待审核的反馈信息，包含查询不到内容时的 CQL 语句，大模型的回答和从回答中提取出的三元组信息。其中从回答中提取出的三元组利用大模型实现。管理员点击“详情”按钮查看详细信息并审核、修改三元组信息。管理员审核的详情页面如图附录 3-8 所示。

返回 补全图谱 审核不通过 关闭

上一页 当前位置: 第 1 / 1 页 下一页 每页显示: 15 条数据 确定

CQL_match:

```
MATCH (d:Disease)-[]-(s:Symptom) where d.name='自身免疫性脑炎' return s.name
```

reply: 大模型生成的回答

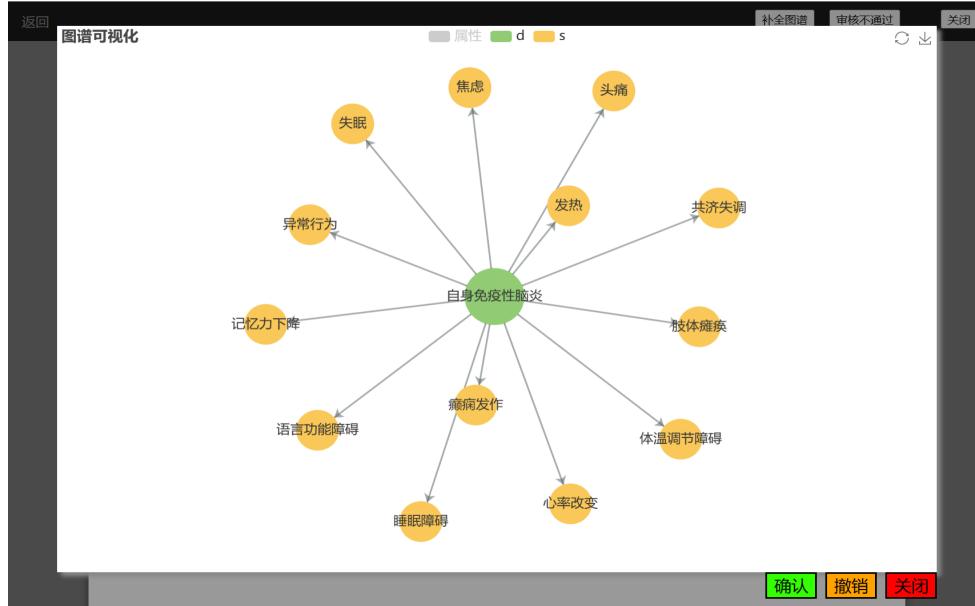
您：自身免疫性脑炎有什么表现？ 自身免疫性脑炎的表现多样，主要包括：前驱症状如发热、头痛；精神行为异常，如焦虑、失眠、异常行为；认知功能障碍，如记忆力下降、不认识家人朋友；语言功能障碍；癫痫发作；睡眠障碍；自主神经功能紊乱，如体温调节障碍、心率改变；以及不同程度的意识障碍和神经损害症状，如肢体瘫痪、共济失调等。及时诊断和治疗非常重要。

Triplets: 从回答中抽取出来的三元组

```
自身免疫性脑炎-可能症状-头痛;
自身免疫性脑炎-可能症状-焦虑;
自身免疫性脑炎-可能症状-失眠;
自身免疫性脑炎-可能症状-异常行为;
自身免疫性脑炎-可能症状-记忆力下降;
自身免疫性脑炎-可能症状-语言功能障碍;
```

图附录 3-8 管理员审核的详情页面

点击右上角的“补全图谱”按钮，将会参照审核修改后的三元组，对知识图谱进行补全，并显示补全后的结果，如图附录 3-9 所示。点击“确认”按钮将会确认补全的内容，点击“撤销”按钮将会撤销补全的内容。



图附录 3-9 管理员审核的补全结果可视化页面

2.5 其他功能

2.5.1 登录注册功能

为了给系统增加身份验证，设计开发了登录注册功能，普通用户可进行除了审核、系统设置外的其它操作，管理员用户拥有审核、系统设置操作的权限。用户的信息存在数据库中，表结构见表附录 3-1。登陆界面见图附录 3-10 所示。

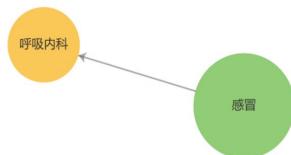
The screenshot shows a login form titled "登录" (Login). It has two input fields: "用户名:" (Username) containing "LZG" and "密码:" (Password) containing ".....". Below the inputs are two green buttons: "登录" (Login) on the left and "注册" (Register) on the right.

图附录 3-10 登录界面

2.5.2 图谱可视化功能

图 9 展示了图谱可视化应用于审核页面的情况，此外，图谱可视化功能还可用在查看用户问题在知识图谱中查询的结果、根据 CQL 语句查看知识图谱查询结果

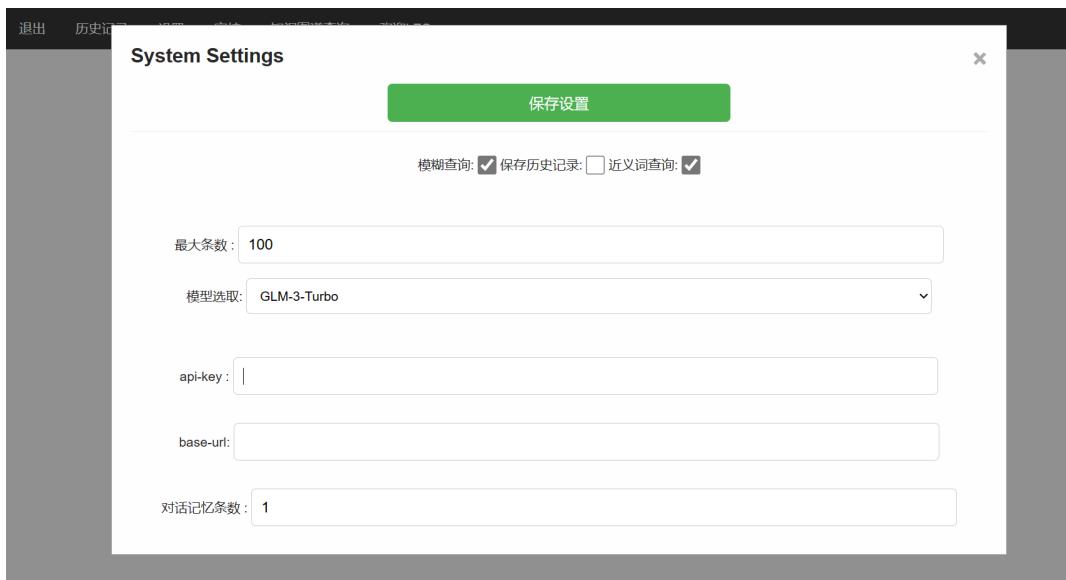
这两方面，在图附录 3-4 中点击右侧的图谱图标，即可查看问题“感冒要看哪科”对应的知识图谱查询结果，如图附录 3-11 所示。在上方的输入框中输入 CQL 语句，也可查看 CQL 语句查询到的结果。图谱可视化功能基于 Apache ECharts 工具实现。



图附录 3-11 图谱可视化功能

2.5.3 系统设置功能

管理员登陆后，可点击“设置”按钮进入系统设置页面，管理员可对系统进行配置，配置项包括是否开启历史记录、历史记录条数、大模型的选择、是否开启模糊查询和近义词查询、对话历史记录的保存条数。设置功能的界面如图附录 3-12 所示。



图附录 3-12 系统设置页面

2.5.3 对话历史记录查询

点击“历史记录”按钮进入历史记录页面，用户可对对话的历史记录进行查

附录 3 中期报告

询，如图附录 3-13 所示。

问题	回答	时间
巨大肥厚性胃炎的易感人群？	好发于50岁以上人群	2024-03-30 20:40:18
胃炎的病因？	胃炎的病因可由化学因素、物理因素、微生物感染或细菌毒素等引起，Hp感染是慢性胃炎的主要病因，易感体质和环境因素也与其相关。	2024-03-30 19:22:12
糜烂性胃炎吃什么食物不好？	糜烂性胃炎不适宜食用白扁豆、红葡萄酒、白酒和啤酒。	2024-03-30 19:15:45
肠胃炎要看哪科？	肠胃炎要看消化内科。	2024-03-30 18:54:28

图附录 3-13 历史记录页面

三、存在的问题和拟解决方法

(1) 界面不够美观，需要进一步优化。

拟解决方法：进一步学习 CSS 知识，或学习使用 layui、elementui 等工具，对页面进行美化。

(2) 目前仅实现了对知识图谱内容的补充，没有内容修改功能。

拟解决方法：考虑新增用户点踩功能，用户对回答不满意时可提交相关的反馈，经管理员审核后修改知识图谱的内容

(3) 项目中利用大模型实现三元组提取和实体识别任务时，均采用 api 调用+prompt 实现，缺乏大模型微调技术的应用。

拟解决方法：目前正在尝试微调 chatglm3-6b 模型实现三元组提取和实体识别任务，后面尝试对基于微调的实现方式与基于 api 调用+prompt 的实现方式作比较，观察哪种效果更好。

燕山大学本科毕业设计（论文）

附录 4 外文原文

Knowledge Graphs Enhanced Large Language Model Prompt for Electric Power Question Answering

ABSTRACT

With the continuous development and digital transformation in the field of electric power, the application of large language models in the electric power industry has become a remarkable trend. The electric power industry is an information-intensive domain involving extensive data processing, predictive analysis, and decision-making. Therefore, the application of large language models in the electric power sector is of great significance. Current large language models such as GPT3.5 and GLM can perform well in tasks such as question answering dialogues. However, these models still face challenges such as answer hallucination and inaccurate responses. This paper proposes a method to enhance question answering in large language models using knowledge graphs, aiming to improve the accuracy and reliability of these models in question answering tasks in the electric power domain. The proposed method first utilizes local electric power data to extract triplets and generate a question answering dataset specific to the electric power domain using a large language model. Then, the relationships of the knowledge graph triplets are incorporated into the question prompt to enhance the quality of the model's answers. Furthermore, we fine-tune the large language model using the expanded question set derived from the triplets as knowledge enhanced data. Subsequently, we conduct experiments on both an electric power question answering dataset and a knowledge graph question answering dataset. The experimental results demonstrate that our method significantly improves various metrics of the large language model in the electric power question answering task. This research provides new insights and approaches to enhance the effectiveness of question answering systems in the electric power domain. Future studies can further explore and optimize this prompt expansion method for application in broader domains and tasks.

1 INTRODUCTION

In recent years, deep learning has achieved numerous break-throughs. In 2012, Hinton et al. successfully trained the deep convolutional neural network (CNN) AlexNet on the

ImageNet dataset, which attracted global attention to deep learning [1]. In 2015, Kaiming He et al. proposed a novel algorithm called ResNet, enabling successful training of deep neural networks with 100 layers [2]. The emergence of AlphaGo and AlphaGo Zero showcased the remarkable achievements of artificial intelligence in the field of Go, with deep learning playing a vital role [3].

Transformers are a model framework based on self-attention mechanisms, initially introduced by Vaswani et al. in 2017 [4]. In essence, transformers can process input sequences in parallel and do not suffer from long-range dependency issues, leading to improved computational efficiency and better performance compared to previous models. In the field of natural language processing, transformers have been widely applied to various tasks, including language representation, machine translation, sentiment analysis, and question answering systems. Notably, the GPT-3 model represents one of the most representative transformer language models. In 2020, the GPT-3 model was developed by the AiBeKe team and is currently one of the most advanced natural language processing models worldwide. With 175 billion parameters, it can perform multiple tasks such as text generation, question answering, and dialogue generation [5]. GPT-3 has broken performance records in tasks such as text generation, language translation, and dialogue generation, demonstrating its powerful application potential and making large language models a hot topic in artificial intelligence research.

With the advancement of large language models, pre-trained language models have shown great potential in performing question answering tasks [16]. The fundamental principle of large language models in question answering lies in leveraging the knowledge learned during pre-training to provide answers. However, current large language models still have limitations in terms of answer accuracy and answer hallucination. To overcome these limitations, we propose a method that involves incorporating knowledge graph triplet information into the question prompt of large language models.

In the electric power question answering dataset, we utilize GPT3.5 to match each question-answer pair with the most relevant triplets from the knowledge graph. We incorporate these triplets into the prompt description of the question and use the question-answer pairs augmented with triplets as a knowledge enhanced dataset for fine-tuning the

large language model. This framework enhances the accuracy of the large language model’s answers by leveraging the information from the knowledge graph and improves the quality of responses.

Therefore, we present a knowledge enhanced fine-tuning framework that utilizes knowledge from the knowledge graph for fine-tuning domain-specific datasets. We generate an electric power question answering dataset using a large language model and conduct experiments on this dataset. Additionally, we validate the generality and effectiveness of our framework on a knowledge graph question answering dataset. The experimental results demonstrate the effectiveness of our framework in improving the accuracy of the large language model’s answers.

2 RELATED WORK

2.1 Artificial Intelligence

Artificial Intelligence (AI) is a technology that simulates human intelligence and can perform various tasks through learning, reasoning, adaptation, and self-correction. AI has been widely applied in fields such as natural language processing, computer vision, robotics, and gaming. Recently, the development of deep learning techniques has led to significant advancements in AI in many domains [6]. Among them, Deep Neural Networks (DNNs) have become one of the most popular AI technologies, achieving tremendous success in various applications. For example, Convolutional Neural Networks (CNNs) have become a standard approach in computer vision. Additionally, Recurrent Neural Networks (RNNs) have achieved significant success in natural language processing [7]. The emergence of Transformer models has greatly advanced natural language processing and other sequence modeling tasks [17].

2.2 Large Language Models

Large Language Models (LLM) are AI models designed to understand and generate human language [11]. They are trained on massive amounts of textual data and can perform a wide range of tasks, including text summarization, translation, sentiment analysis, and more. LLM are characterized by their enormous scale, with billions of parameters, enabling them to learn complex patterns in language data. LLM are built on the Transformer architecture, with multi-head attention layers stacked in a deep neural network. Current LLM primarily

adopt a similar architecture (i.e., Trans- former) and pre-training objectives as smaller language models, with notable differences in model size, pre-training data, and computational requirements. As model size surpasses a certain threshold, the model's capability experiences unpredictable improvements, known as "emergent behavior" [16].

2.3 Knowledge Graphs

Knowledge Graphs are a knowledge representation approach based on graph structures, organizing entities, relationships, and attributes into a large-scale graph to represent knowledge in the real world [8]. Knowledge Graphs can aid in better understanding and utilization of knowledge and find applications in various domains, such as natural language processing, search engines, intelligent question answering, and more. Constructing a Knowledge Graph requires substantial data and algorithmic support. Several well-known knowledge graphs have been established, including the Baidu Baike Knowledge Graph [9].

2.4 Electric Power Knowledge Graphs

The construction of Power Knowledge Graphs aims to integrate scattered and complex knowledge in the electric power domain, forming a unified knowledge system to facilitate industry development, support decision-making, and enhance work efficiency [10]. To achieve this, various data related to electric power need to be collected. Based on the characteristics of the electric power domain, corresponding ontologies are established, and natural language processing techniques are used to analyze literature and extract knowledge, which is then added to the knowledge graph. Finally, relationships between entities are established within the knowledge graph and visualized for better understanding and utilization. This helps individuals quickly locate the required information, improve work efficiency, and provide decision-makers with comprehensive and accurate information support, assisting them in making informed decisions.

2.5 P-tuning v2

Prompt tuning introduces a method for learning "soft prompts," which involves using a frozen language model to perform specific downstream tasks [20]. Prefix-Tuning proposes a fine-tuning approach that appends a continuous vector sequence tailored for a specific task to the input [13]. Compared to traditional methods, prefix fine-tuning only optimizes the prefix, resulting in minimal overhead for each additional task. P-tuning proposes an

automatic template construction method [14], transforming the template construction into a continuous parameter optimization problem, and achieves excellent performance in zero-shot and few-shot scenarios. Building on the inspiration from previous fine-tuning approaches, P-Tuning v2 [12] is an improved version that combines prefix fine-tuning and P-tuning, deepens the parameter layers, and enhances the effectiveness of fine-tuning.

3 METHODS

Figure 1 presents the overall framework of the knowledge graph-enhanced large language model prompt. Firstly, we use the LLM to generate a question-answering dataset by leveraging a local knowledge base. Next, we match the triples from the knowledge graph with the questions and incorporate the matched triple knowledge into the prompts, enhancing the information contained in the questions. Finally, we fine-tune the LLM using the knowledge enhanced dataset.

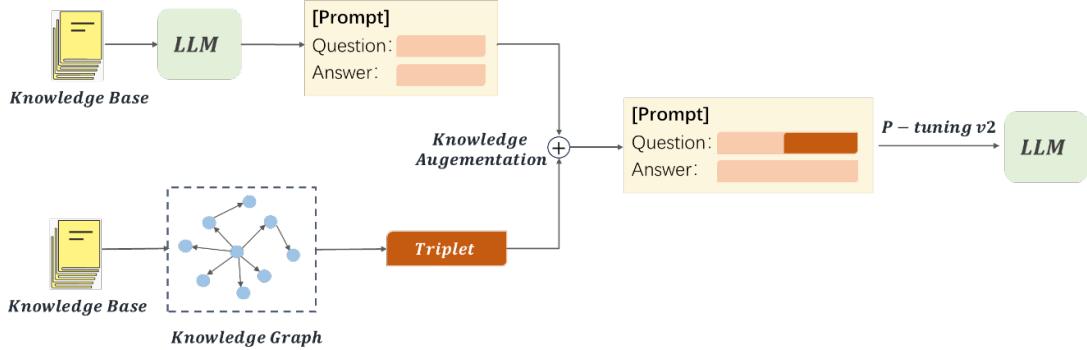


Figure 1: Overall Framework of Knowledge Graph Enhanced Large Language Model

Prompt

3.1 Electric Power QA Dataset Generation

Large language models typically predict the next token based on the preceding tokens, known as autoregressive language models. For instance, given an input question x , a question-answering (QA) system returns an answer y , where x and y are token sequences: $x = [w_1, w_2, \dots, w_{|x|}]$. Let P be the QA model based on generative language model (LM), and the conditional probability of generating answer y given question x is denoted as $P(y|x)$. In the process of generating the electric power QA dataset, we utilize OpenAI's GPT-3.5 Turbo model, which has been widely used and demonstrated excellent language understanding capabilities. Using local electric power related texts as

input, we call the API with specific prompts to let the model summarize the knowledge in the texts and provide answers to the questions, thereby generating a high quality electric power QA dataset.

3.2 Knowledge Graphs Enhanced Large Language Model Prompt

This example is taken from the Kgclue dataset. In Figure 2, we can observe that without the addition of triple information, the LLM provides an incorrect answer. However, after incorporating the relevant triple information into the question, the LLM provides a correct answer.

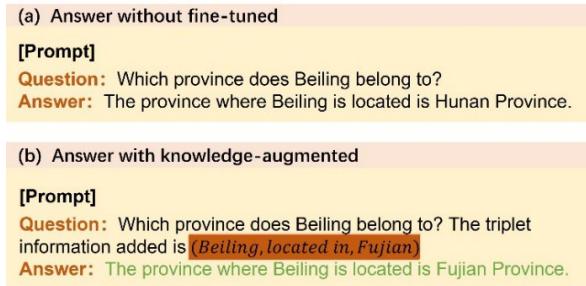


Figure 2: An example of knowledge graph enhancement

The knowledge of LLM relies on the knowledge encoded in the pre-training parameters, which may not be accurate as knowledge evolves over time. Therefore, we utilize an external knowledge base to update the parameters, and our framework provides a high quality updating approach. For each question in the dataset, we match it with an informative triple. In the KgCLUE dataset, this informative triple corresponds to the dataset's label. In the electric power dataset, the matched triple is generated by GPT3.5.

3.3 P-tuning v2 fine-tune LLM

Currently, full-parameter fine-tuning of large language models is computationally expensive. Therefore, we employ an efficient parameter fine-tuning method called p-tuning v2 [15]. This approach involves freezing the original model parameters and introducing trainable prefix parameters in front of each layer. This idea is inspired by prefix-tuning and p-tuning. In the model structure of p-tuning v2 shown in Figure 3, the yellow part represents the adjustable prefix parameters, while the blue part represents the frozen parameters of the large language model.

In prefix fine-tuning, the autoregressive model is represented as $z = [prefix; x; y]$. A matrix P_θ is constructed, where $P_\theta \in \mathbb{R}^{|P_{idx}| \times \text{dim}(hi)}$. This part of the parameters is used

to store the prefix parameters:

$$h_i = \begin{cases} P_\theta [i, :], & \text{if } i < P_{idx} \\ LM_\varphi(z_i, h_{<i}), & \text{otherwise} \end{cases}$$

Equation (1) indicates that the parameters of tokens in the prefix part are trainable, while the parameters of other tokens are fixed, i.e., the original model parameters are frozen. The optimization objective function of this method is:

$$\max_\phi \log p_\phi(y|x) = \sum_{i \in Y_{idx}} \log p_\phi(z_i|h_{<i})$$

In Equation (2), x represents the input sequence, y represents the output sequence, and $P_\phi(y|x)$ is the Transformer-based autoregressive language model. Y_{idx} denotes the indices sequence corresponding $h_i = LM_\varphi(z_i, h_{<i})$ is used to calculate the distribution of the next token. $p_\phi(z_i|h_{<i}) = \text{softmax}(W_\phi h_i^{(n)})$

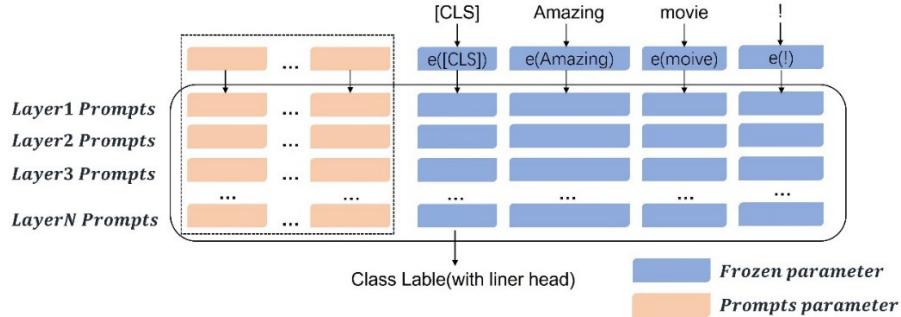


Figure 3: Depicts the structure of p-tuning v2

P-tuning v2 combines prefix-tuning with p-tuning. It constructs multiple layers of consecutive prompt model templates, where each layer's consecutive prompt template is defined as:

$$\{h_0, \dots, h_i, e(x), h_{i+1}, \dots, h_m, e(y)\}$$

In Equation (3), h_i represents the continuous prompt representation of p-tuning, e represents a pre-trained language model, x represents the original input data, and y represents the data labels. When facing downstream tasks, the model optimization is achieved by optimizing h :

$$\hat{h}_{0:m} = \operatorname{argmin} L(M(x, y))$$

The fine-tuning utilizes a dataset from the electric power domain, incorporating a large-scale corpus of electric power related texts such as research reports, innovation reports, and work reports after processing them through a knowledge graph and adding them to the

prompts.

4 EXPERIMENTS

4.1 KgCLUE Dataset

KBQA (Knowledge-based Question Answering) leverages the rich semantic relationships provided by knowledge graphs to gain a deep understanding of user questions and provide answers. It has attracted significant attention from both academia and industry. The main task of KBQA is to map natural language questions (NLQ) to structured queries using various methods and retrieve answers from the knowledge graph.

KgCLUE serves as a Chinese KBQA benchmark dataset, consisting of 3,121,457 entities, 245,838 relation types, and 20,559,652 triples. The knowledge base is derived from encyclopedic data, specifically factual triples extracted from encyclopedic search pages. The original dataset is in knowledge graph question-answering format. To adapt it for the QA task, we consider the last entity in each triple as the answer. The dataset takes the following form:

{"question": "Do you know which company produces the TV series 'Overwatch Starlight'?", "answer": "Han Yuling Original Music"}.

4.2 Electric Power Dataset

Knowledge in the field of electric power is derived from various sources, including traditional electric power knowledge engineering systems, structured data from expert knowledge bases, as well as semi-structured/unstructured data such as electric power standards, regulations, laws, and the expertise of professionals and technicians. In this study, we utilize a large amount of data material from the Power Institute, which includes both paper-based scanning materials and electronically archived data. Through entity and relation recognition, we obtained 15,860 entities, 12,639 relation types, and 20,552 triples. The electric power QA dataset was generated by GPT summarization and consists of 1000 QA pairs. Among them, 800 pairs are used as the training set, and 200 pairs are used as the test set. The original dataset takes the following form: {"content": "What upgrades have been made to the digital certificate system by State Grid Corporation of China to meet national requirements for algorithm issues?", "summary": "The State Grid Corporation of China has upgraded the root CA system, operating CA system, KM system, headquarters

RA system, encryption device, and electronic key in the digital certificate system.”}.

4.3 Base Model

The Base Model used in this study is the General Language Model (GLM), which is a blank-filling-based autoregressive language model. ChatGLM-6B is a bilingual dialogue language model based on the GLM architecture, supporting both Chinese and English. It has 6.2 billion parameters [15]. ChatGLM-6B employs techniques similar to ChatGPT, optimized for Chinese question answering and dialogue. It has been trained on approximately 1T bilingual tokens and fine-tuned using supervised fine-tuning, feedback self-training, and human feedback reinforcement learning.

4.4 Experiments on KgCLUE Dataset and Electric Power Dataset

We conducted a series of experiments to evaluate the performance of the model under different training datasets and fine-tuning methods. We selected a public dataset and a self-constructed electric power dataset as the experimental datasets. The experiments involved the following fine-tuning methods: unfine-tuned model, model fine-tuned using the original dataset, model fine-tuned using a dataset with randomly added triples, and model fine-tuned using a dataset with matched triples. The matched triples in the KgCLUE dataset were obtained from the dataset labels, while the matched triples in the electric power QA dataset were obtained by matching the labels with the electric power knowledge graph using GPT. We evaluated the model’s performance using Bleu-4 [18], Rouge-1, Rouge-2, and Rouge-L [19] metrics, which measure the “similarity” between the model’s answers and the labels. First, we used the unfine-tuned model as the baseline model and evaluated it on both the public dataset and the electric power dataset. Then, we fine-tuned the GLM using the original dataset, allowing the model to adapt to the language and knowledge specific to the electric power domain, aiming to improve the model’s performance. Additionally, to compare the effectiveness of adding triples, we conducted experiments using randomly added triples and matched triples.

The experimental results are presented in Tables 1 and 2, where each row represents the following models: unfine-tuned model, model fine-tuned using the original dataset, model fine-tuned using a dataset with randomly added triples, and model fine-tuned using a dataset with matched triples.

Table 1: Results of GLM on KgCLUE Dataset

Methods	Bleu-4	Rouge-1	Rouge-2	Rouge-L
Original	3.87	10.37	3.92	7.66
p-tuning v2	32.33	41.47	21.41	41.04
Random+p-tuning v2	10.37	15.58	3.51	15.25
Ours+p-tuning v2	60.01	69.36	46.51	68.91

Table 2: Results of GLM on Electric Power QA Dataset

Methods	Bleu-4	Rouge-1	Rouge-2	Rouge-L
Original	7.51	30.37	11.04	16.24
p-tuning v2	28.87	53.65	33.35	45.94
Random+p-tuning v2	23.41	45.21	22.70	37.64
Ours+p-tuning v2	32.46	54.21	34.31	47.54

From Table 1, we observe that for the unfine-tuned model on the KgCLUE dataset, it was not subjected to any special treatment or optimization. According to the experimental results, this model exhibited relatively lower performance across all metrics. This could be attributed to the inconsistency between the test set labels, which are usually short phrases or words, and the long and stylistically different sentences generated by the large language model. As a result, the similarity between the generated sentences and the labels was relatively low.

For the model fine-tuned using the original dataset, this approach showed some performance improvement across all metrics. Specifically, we observed improvements in the Bleu-4, Rouge-1, Rouge-2, and Rouge-L scores. This indicates that after fine-tuning, the language length and style of the generated sentences by the large language model became closer to the labels in the dataset. However, the accuracy of the generated outputs by the large language model was still not high.

Regarding the model fine-tuned using a dataset with randomly added triples, we can see a decrease in performance compared to the model fine-tuned using the original dataset across all metrics. However, it showed improvement compared to the unfinetuned model. This suggests that the randomly added triple information introduced some interference in the

model's generated answers, which resulted in lower performance compared to the model fine-tuned using the original dataset. Nevertheless, the fine-tuned outputs exhibited a higher similarity to the language style of the test set labels compared to the original model, indicating an improvement in answer similarity.

For our proposed model fine-tuned with matched triples, according to the experimental results, this approach achieved significant improvements across all metrics. This indicates that the large language model not only learned the language style of the dataset answers but also acquired knowledge from the provided triples. Compared to the model fine-tuned using the original dataset, our method enhanced the model's ability to extract knowledge from triples, resulting in more accurate answers.

From Table 2, we can see that on the Power QA dataset, although the unfine-tuned model exhibited lower similarity, it performed better compared to the unfine-tuned model in Table 1. This is because the answers in the Power QA test set were generated using the GPT-3.5-turbo model interface, thus having a certain degree of similarity in language style.

The model fine-tuned using the original dataset showed improvements across all metrics. However, after fine-tuning with a dataset containing randomly added triples, the metrics decreased. This suggests that for more complex electric power QA tasks, introducing incorrect triple information can interfere with the model's output and reduce its accuracy.

Regarding our proposed method, all metrics showed improvements. However, due to the higher difficulty and complexity of the electric power QA task and dataset, the improvements were not as significant as those in Table 1. Figure 4 illustrates the comparison of loss curves, and by examining the convergence of the three curves, we can observe that our method achieved the lowest final loss value, indicating better performance during the model's training process.

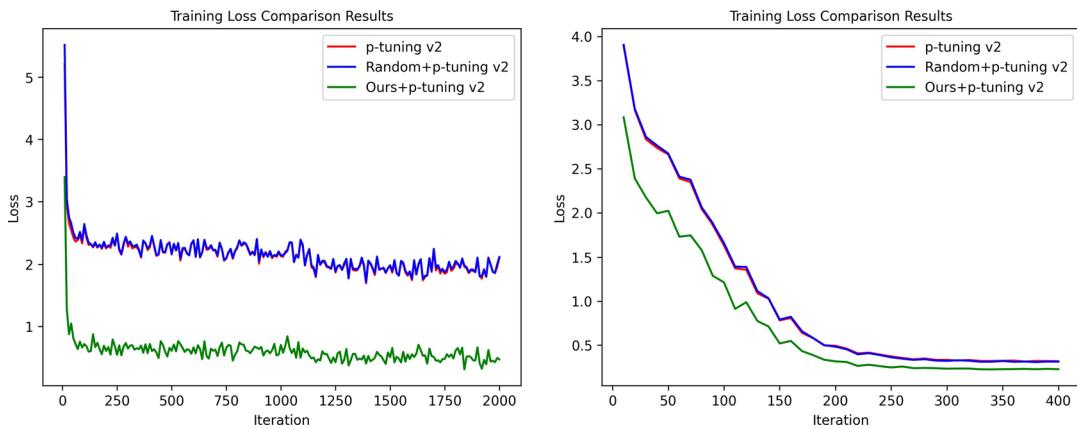


Figure 4: Comparison of Loss Curves on KgCLUE Dataset (left) and Power QA Dataset (right)

5 CONCLUSION

In this paper, we propose a method that enhances large language models with knowledge graph prompts for electric power knowledge question answering tasks. This method is simple yet effective in improving the accuracy of model answers and addresses the issue of inaccurate responses to some extent. We conducted a series of experiments using a electric power QA dataset and a knowledge graph QA dataset. The experimental results demonstrate that the proposed method is straightforward and effective, enabling the model to extract valuable information from triples and improve the accuracy of answers.

However, there are still some limitations in this work. For instance, the current matching method employs the GPT-3.5 model, and it would be beneficial to train a dedicated matching model to enhance the accuracy of matching. Additionally, the fine-tuning methods and large language models used in this experiment can be replaced, and different fine-tuning techniques and large language models can be explored to verify the effectiveness and generality of this approach.

REFERENCES

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. <https://doi.org/10.1145/3065386>
- [2] He, Kaiming, et al. Deep residual learning for image recognition. 2016. IEEE

Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] Silver David, Huang Aja, et al. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489, 2016. <https://doi.org/10.1038/nature16961>

Ashish Vaswani, Noam Shazeer, et al. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS’17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[4] Tom B. Brown, Benjamin Mann, et al. 2020. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS’20). Curran Associates Inc., Red Hook, NY, USA, Article 159, 1877–1901.

[5] Heaton, J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genet Program Evolvable Mach* 19, 305–307, 2018. <https://doi.org/10.1007/s10710-017-9314-z>

[6] Dhruv, P., Naskar, S. 2020. Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review. In: Swain, D., Pattnaik, P., Gupta, P. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1101. Springer, Singapore. https://doi.org/10.1007/978-981-15-1884-3_34

[7] Chen, Zhe and Wang, Yuehan and Zhao, Bin and Cheng, Jing and Zhao, Xin and Duan, Zongtao. 2020. Knowledge Graph Completion: A Review. *IEEE Access*, vol. 8, pp. 192435–192456, 2020, doi: 10.1109/ACCESS.2020.3030076.

McHugh, Justin, et al. Integrated access to big data polystores through a knowledge-driven framework. 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017.

[8] Tang, Yachen, et al. 2019. Enhancement of power equipment management using knowledge graph. 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia).

Alec Radford, Jeffrey Wu, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[9] Liu, Xiao, et al. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Dublin, Ireland. 61–68. doi :10.18653/v1/2022.acl-short.8

[10] Li, Xiang Lisa, and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for

Generation. 2021. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 4582—4597.doi: 10.18653/v1/2021.acl-long.353

[11] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, Jie Tang. 2021. GPT understands, too. arXiv preprint arXiv:2103.10385.

Du, Zhengxiao, et al. 2022. GLM: General language model pretraining with autoregressive blank infilling. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Dublin, Ireland. 320–335

[12] Wayne Xin Zhao, Kun Zhou, et al. 2023. A Survey of Large Language Models. arXiv preprint arXiv:2303.18223.

[13] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient Transformers: A Survey. ACM Comput. Surv. 55, 6, Article 109 (June 2023), 28 pages. <https://doi.org/10.1145/3530811>

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA.

[15] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain.

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic.

附录 5 外文翻译

电力问答领域的知识图谱增强大语言模型提示词

摘要

随着电力领域的持续发展和数字化转型，大型语言模型在电力行业的应用已成为一个显著的趋势。电力行业是一个信息密集型领域，涉及广泛的数据处理、预测分析和决策。因此，大型语言模型在电力领域的应用具有重大意义。目前的大型语言模型，如 GPT3.5 和 GLM，在诸如问答对话等任务中表现良好。然而，这些模型仍面临着回答幻觉和不准确回答等挑战。本文提出了一种利用知识图谱增强大型语言模型中问答能力的方法，旨在提高这些模型在电力领域问答任务中的准确性和可靠性。所提出的方法首先利用本地电力数据提取三元组，并使用大型语言模型生成特定于电力领域的问答数据集。然后，将知识图谱三元组的关系纳入问题提示中，以提高模型答案的质量。此外，我们使用从三元组派生的扩展问题集作为知识增强数据来微调大型语言模型。随后，我们在电力问答数据集和知识图谱问答数据集上进行实验。实验结果表明，我们的方法显著改善了大型语言模型在电力问答任务中的各种指标。这项研究为提高电力领域问答系统的有效性提供了新的见解和方法。未来的研究可以进一步探索和优化这种提示扩展方法，以应用于更广泛的领域和任务。

1 引言

近年来，深度学习取得了许多突破。2012 年，Hinton 等人成功地在 ImageNet 数据集上训练了深度卷积神经网络（CNN）AlexNet，引起了全球对深度学习的关注。2015 年，Kaiming He 等人提出了一种名为 ResNet 的新算法，使得具有 100 层的深度神经网络能够成功训练。AlphaGo 和 AlphaGo Zero 的出现展示了人工智能在围棋领域取得的显著成就，深度学习在其中发挥了关键作用。

Transformer 是一种基于自注意力机制的模型框架，最初由 Vaswani 等人于 2017 年提出。从本质上讲，Transformer 可以并行处理输入序列，不会受到长距离依赖问题的影响，从而提高了计算效率并改善了性能，相较于之前的模型。在自然语言处理领域，Transformer 已广泛应用于各种任务，包括语言表示、机器翻译、情感分析和问答系统。值得注意的是，GPT-3 模型是最具代表性的 Transformer 语言模型之一。2020 年，AiBeKe 团队开发了 GPT-3 模型，它是目前全球最先进的自然语言处理模型。

之一。该模型具有 1750 亿个参数，可以执行文本生成、问答和对话生成等多个任务。GPT-3 在文本生成、语言翻译和对话生成等任务中创下了性能记录，展示了其强大的应用潜力，使大型语言模型成为人工智能研究的热门话题。

随着大型语言模型的发展，预训练的语言模型在执行问答任务方面显示出巨大的潜力。大型语言模型在问答中的基本原则在于利用预训练期间学到的知识来提供答案。然而，当前的大型语言模型在答案准确性和答案幻觉方面仍存在局限。为了克服这些局限，我们提出了一种方法，即将知识图谱三元组信息纳入大型语言模型的问题提示中。

在电力问答数据集中，我们利用 GPT3.5 将每个问答对与知识图谱中最相关的三元组进行匹配。我们将这些三元组纳入问题的提示描述中，并使用带有三元组的问答对作为知识增强数据集，用于微调大型语言模型。这一框架通过利用知识图谱中的信息提高了大型语言模型答案的准确性，并改善了回答的质量。

因此，我们提出了一种利用知识图谱中的知识来微调领域特定数据集的知识增强的 Fine-tuning 框架。我们使用大型语言模型生成了一个电力问答数据集，并在该数据集上进行了实验。此外，我们还在知识图谱问答数据集上验证了我们框架的普适性和有效性。实验结果表明，我们的框架显著提高了大型语言模型答案的准确性。

2 相关工作

2.1 人工智能

人工智能（AI）是一种模拟人类智能的技术，通过学习、推理、适应和自我纠正来执行各种任务。AI 已广泛应用于自然语言处理、计算机视觉、机器人技术和游戏等领域。近年来，深度学习技术的发展在许多领域取得了重大进展。其中，深度神经网络（DNNs）已成为最受欢迎的 AI 技术之一，在各种应用中取得了巨大成功。例如，卷积神经网络（CNNs）已成为计算机视觉中的标准方法。此外，循环神经网络（RNNs）在自然语言处理方面取得了显著的成功。Transformer 模型的出现极大地推动了自然语言处理和其他序列建模任务的发展。

2.2 大型语言模型

大型语言模型（LLM）是旨在理解和生成人类语言的 AI 模型。它们在大量文本数据上进行训练，可以执行广泛的任务，包括文本摘要、翻译、情感分析等。LLM 的特点是规模庞大，具有数十亿个参数，使其能够学习语言数据中的复杂模式。LLM 基于 Transformer 架构，其中多头注意力层堆叠在深度神经网络中。当前的 LLM 主要采用

类似的架构（即 Transformer）和较小语言模型的预训练目标，但在模型大小、预训练数据和计算需求方面存在明显差异。随着模型大小超过一定阈值，模型的能力会出现不可预测的改进，这被称为“emergent behavior”。

2.3 知识图谱

知识图谱是一种基于图结构的知识表示方法，将实体、关系和属性组织成大规模的图，以表示现实世界中的知识。知识图谱有助于更好地理解和利用知识，并在自然语言处理、搜索引擎、智能问答等各个领域中应用。构建知识图谱需要大量的数据和算法支持。已经建立了一些知名的知识图谱，包括百度百科知识图谱。

2.4 电力知识图谱

电力知识图谱的构建旨在整合电力领域中分散且复杂的知识，形成一个统一的知识体系，以促进行业发展、支持决策并提高工作效率。为实现这一目标，需要收集与电力相关的各种数据。根据电力领域的特点，建立相应的本体，并使用自然语言处理技术分析文献并提取知识，然后将其添加到知识图谱中。最后，在知识图谱内建立实体之间的关系，并进行可视化，以便更好地理解和利用。这有助于个人快速定位所需信息，提高工作效率，并为决策者提供全面准确的信息支持，帮助他们做出明智的决策。

2.5 P-tuning v2

Prompt tuning 引入了一种学习“软提示”的方法，其中使用一个冻结的语言模型来执行特定的下游任务。Prefix-Tuning 提出了一种微调方法，将针对特定任务的连续向量序列附加到输入中。与传统方法相比，前缀微调仅优化前缀，从而对每个附加任务的开销最小。P-tuning 提出了一种自动模板构建方法，将模板构建转化为连续参数优化问题，并在零样本和少样本场景中取得了出色的性能。基于之前微调方法的启发，P-Tuning v2 是一个改进版本，它将前缀微调和 P-tuning 相结合，加深了参数层次，并增强了微调的效果。

3 方法

图 1 展示了知识图增强的大型语言模型提示的整体框架。首先，我们使用 LLM 通过利用本地知识库生成一个问答数据集。接下来，我们将知识图中的三元组与问题进行匹配，并将匹配的三元组知识融入提示中，增强了问题中包含的信息。最后，我们使用增强知识的数据集对 LLM 进行微调。

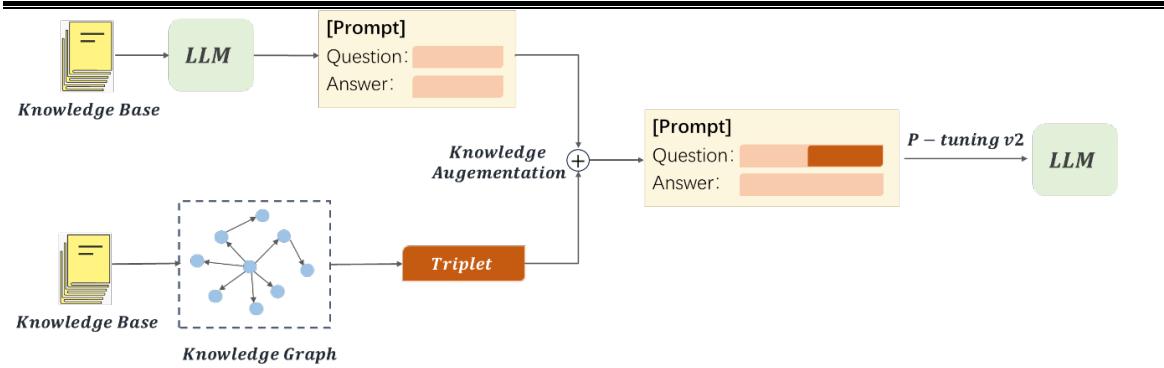


图 1 知识图谱增强大语言模型提示总体框架图

3.1 电力问答数据集的生成

大型语言模型通常基于前面的词元来预测下一个词元，这被称为自回归语言模型。例如，给定一个输入问题 x ，一个问答（QA）系统返回一个答案 y ，其中 x 和 y 是词元序列： $x=[w_1, w_2, \dots, w|x|]$ 。设 P 为基于生成性语言模型（LM）的 QA 模型，那么在给定问题 x 的条件下生成答案 y 的条件概率表示为 $P(y|x)$ 。在生成电力问答数据集的过程中，我们使用了 OpenAI 的 GPT-3.5 Turbo 模型，该模型已被广泛使用并展示了出色的语言理解能力。使用与本地电力相关的文本作为输入，我们通过特定提示调用 API，让模型总结文本中的知识并提供问题的答案，从而生成高质量的电力问答数据集。

3.2 知识图谱增强的大型语言模型提示

这个例子取自 Kgclue 数据集。在图 2 中，我们可以观察到，在没有添加三元组信息的情况下，LLM 提供了一个错误的答案。然而，在将相关的三元组信息加入到问题中后，LLM 提供了一个正确的答案。

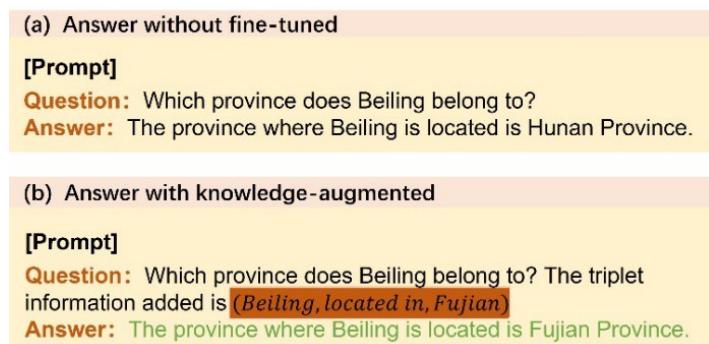


图 2：知识图谱增强的一个案例

LLM 的知识依赖于预训练参数中编码的知识，随着时间的推移，这些知识可能不再准确。因此，我们利用外部知识库来更新参数，我们的框架提供了一种高质量的更新方

法。对于数据集中的每个问题，我们都会匹配一个信息丰富的三元组。在 KgCLUE 数据集中，这个信息丰富的三元组对应于数据集的标签。在电力数据集中，匹配的三元组由 GPT3.5 生成。

3.3 P-tuning v2 微调 LLM

目前，大型语言模型的全参数微调在计算上非常昂贵。因此，我们采用一种称为 p-tuning v2 的高效参数微调方法。这种方法涉及冻结原始模型参数，并在每一层前引入可训练的前缀参数。这个想法受到前缀调整和 p-tuning 的启发。在 p-tuning v2 的模型结构中，黄色部分代表可调整的前缀参数，而蓝色部分代表大型语言模型的冻结参数。

在前缀微调中，自回归模型表示为 $z = [prefix; x; y]$ 。构造一个矩阵 P_θ ，其中 $P_\theta \in \mathbb{R}^{|P_{idx}| \times \dim(h)}$ 。这部分参数用于存储前缀参数：

$$h_i = \begin{cases} P_\theta[i, :], & \text{if } i < P_{idx} \\ LM_\phi(z_i, h_{<i}), & \text{otherwise} \end{cases}$$

这里的方程（1）表明前缀部分的令牌参数是可训练的，而其他令牌的参数是固定的，即原始模型参数被冻结。该方法的优化目标函数如下：

$$\max_\phi \log p_\phi(y|x) = \sum_{i \in Y_{idx}} \log p_\phi(z_i|h_{<i})$$

在方程（2）中， x 表示输入序列， y 表示输出序列，而 $P_\phi(y|x)$ 则是基于 Transformer 的自回归语言模型。 Y_{idx} 表示与 $h_i = LM_\phi(z_i, h_{<i})$ 对应的索引序列，用于计算下一个令牌的分布。 $p_\phi(z_i|h_{<i}) = \text{softmax}(W_\phi h_i^{(n)})$ 。

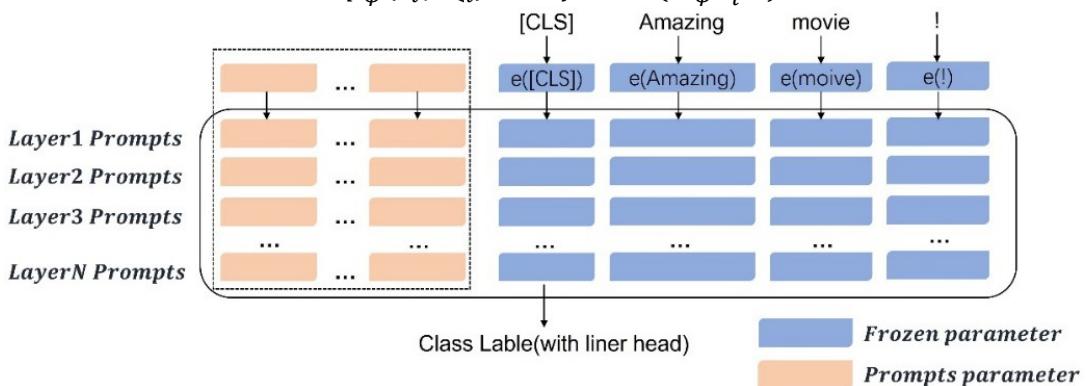


图 3: p-tuning v2 的结构

P-tuning v2 是一种将 prefix-tuning 与 p-tuning 结合的方法。它构建了多层连续的提示模型模板，其中每一层的连续提示模板定义如下：

$$\{h_0, \dots, h_i, e(x), h_{i+1}, \dots, h_m, e(y)\}$$

在方程(3)中, h_i 表示 p-tuning 的连续提示表示, e 表示预训练的语言模型, x 表示原始输入数据, y 表示数据标签。在面对下游任务时, 模型优化通过优化 h 来实现:

$$\hat{h}_{0:m} = \operatorname{argmin} L(M(x, y))$$

微调使用来自电力领域的数据集, 该数据集包括大规模的电力相关文本语料库, 如研究报告、创新报告和工作报告, 在通过知识图谱处理后将其添加到提示中。这种方法有助于模型更好地理解特定领域的内容, 并为下游任务提供优化。

4 实验

4.1 KgCLUE 数据集

基于知识的问答 (KBQA) 利用知识图谱提供的丰富语义关系来深入理解用户问题并提供答案。它已经吸引了学术界和工业界的显著关注。KBQA 的主要任务是使用各种方法将自然语言问题(NLQ)映射到结构化查询, 并从知识图谱中检索答案。KgCLUE 作为一个中文 KBQA 基准数据集, 包含 3,121,457 个实体, 245,838 种关系类型和 20,559,652 个三元组。知识库来源于百科全书数据, 特别是从百科全书搜索页面提取的事实三元组。原始数据集采用知识图谱问答格式。为了适应 QA 任务, 我们将每个三元组中的最后一个实体视为答案。数据集采取以下形式: {“问题”: “你知道哪家公司制作了电视剧 ‘守望星光’ 吗?”, “答案”: “韩玉玲原创音乐”}。

4.2 电力数据集

电力领域的知识来自各种来源, 包括传统的电力知识工程系统、专家知识库的结构化数据, 以及电力标准、法规、法律以及专业人员和技术人员的专业知识等半结构化/非结构化数据。在这项研究中, 我们利用了来自电力研究所的大量数据材料, 包括纸质扫描材料和电子存档数据。通过实体和关系识别, 我们获得了 15,860 个实体, 12,639 种关系类型和 20,552 个三元组。电力 QA 数据集是通过 GPT 总结生成的, 包含 1000 个 QA 对。其中 800 对用作训练集, 200 对用作测试集。原始数据集采取以下形式: {“内容”: “国家电网公司为满足国家算法问题的要求对数字证书系统进行了哪些升级?”, “摘要”: “国家电网公司升级了根 CA 系统、运营 CA 系统、KM 系统、总部 RA 系统、加密设备和数字证书系统中的电子钥匙。”}。

4.3 基础模型

本研究中使用的基础模型是通用语言模型 (GLM), 这是一个基于填空的自回归语言模型。ChatGLM-6B 是基于 GLM 架构的双语对话语言模型, 支持中文和英文。它

有 62 亿参数[15]。ChatGLM-6B 采用类似于 ChatGPT 的技术，针对中文问答和对话进行了优化。它已经在大约 1T 双语令牌上进行了训练，并使用监督式微调、反馈自训练和人类反馈强化学习进行了微调。

4.4 对 KgCLUE 数据集和电力数据集的实验

我们进行了一系列实验，以评估模型在不同训练数据集和微调方法下的性能。我们选择了一个公共数据集和一个自构建的电力数据集作为实验数据集。实验涉及以下微调方法：未微调模型，使用原始数据集微调的模型，使用随机添加三元组的数据集微调的模型，以及使用匹配三元组的数据集微调的模型。KgCLUE 数据集中的匹配三元组是从数据集标签中获得的，而电力 QA 数据集中的匹配三元组是通过使用 GPT 将标签与电力知识图谱匹配获得的。我们使用 Bleu-4 [18]、Rouge-1、Rouge-2 和 Rouge-L [19] 指标评估模型的性能，这些指标衡量模型答案与标签之间的“相似性”。首先，我们使用未微调模型作为基线模型，并在公共数据集和电力数据集上进行评估。然后，我们使用原始数据集微调 GLM，使模型适应电力领域特定的语言和知识，旨在提高模型的性能。此外，为了比较添加三元组的有效性，我们使用随机添加的三元组和匹配的三元组进行了实验。实验结果展示在表 1 和表 2 中，每一行代表以下模型：未微调模型，使用原始数据集微调的模型，使用随机添加三元组的数据集微调的模型，以及使用匹配三元组的数据集微调的模型。

表 1: GLM 在 KgCLUE 数据集上的结果

Methods	Bleu-4	Rouge-1	Rouge-2	Rouge-L
Original	3.87	10.37	3.92	7.66
p-tuning v2	32.33	41.47	21.41	41.04
Random+p-tuning v2	10.37	15.58	3.51	15.25
Ours+p-tuning v2	60.01	69.36	46.51	68.91

表 2: GLM 在 Electric Power QA 数据集上的结果

Methods	Bleu-4	Rouge-1	Rouge-2	Rouge-L
Original	7.51	30.37	11.04	16.24
p-tuning v2	28.87	53.65	33.35	45.94
Random+p-tuning v2	23.41	45.21	22.70	37.64
Ours+p-tuning v2	32.46	54.21	34.31	47.54

从表 1 中，我们观察到对于 KgCLUE 数据集上未经微调的模型，它没有经过任何特殊处理或优化。根据实验结果，这个模型在所有指标上表现相对较低。这可能是由于测试集标签与大型语言模型生成的长句和风格不同的句子之间的不一致性。因此，生成的句子与标签之间的相似性相对较低。

对于使用原始数据集微调的模型，这种方法在所有指标上都显示出了一些性能改进。具体来说，我们观察到 Bleu-4、Rouge-1、Rouge-2 和 Rouge-L 分数的提高。这表明经过微调后，大型语言模型生成的句子的语言长度和风格更接近数据集中的标签。然而，大型语言模型生成的输出的准确性仍然不高。

关于使用随机添加三元组的数据集微调的模型，我们可以看到与使用原始数据集微调的模型相比，在所有指标上的性能都有所下降。然而，与未微调模型相比，它显示出了改进。这表明随机添加的三元组信息在模型生成的答案中引入了一些干扰，导致性能低于使用原始数据集微调的模型。尽管如此，微调后的输出与测试集标签的语言风格相比原始模型有更高的相似性，表明答案相似性有所改进。

对于我们提出的使用匹配三元组微调的模型，根据实验结果，这种方法在所有指标上都取得了显著的改进。这表明大型语言模型不仅学习了数据集答案的语言风格，还从提供的三元组中获得了知识。与使用原始数据集微调的模型相比，我们的方法增强了模型从三元组中提取知识的能力，从而得到了更准确的答案。

从表 2 中，我们可以看到在电力 QA 数据集上，尽管未微调模型显示出较低的相似性，但与表 1 中的未微调模型相比，它的表现更好。这是因为电力 QA 测试集中的答案是使用 GPT-3.5-turbo 模型接口生成的，因此在语言风格上有一定程度的相似性。

使用原始数据集微调的模型在所有指标上都显示出了改进。然而，在使用包含随机添加三元组的数据集进行微调后，指标下降了。这表明对于更复杂的电力 QA 任务，引入不正确的三元组信息可能会干扰模型的输出并降低其准确性。

关于我们提出的方法，所有指标都显示出了改进。然而，由于电力 QA 任务和数据集的难度和复杂性更高，改进并不像表 1 中那样显著。图 4 展示了损失曲线的比较，通过检查三条曲线的收敛情况，我们可以观察到我们的方法实现了最低的最终损失值，表明在模型的训练过程中表现更好。

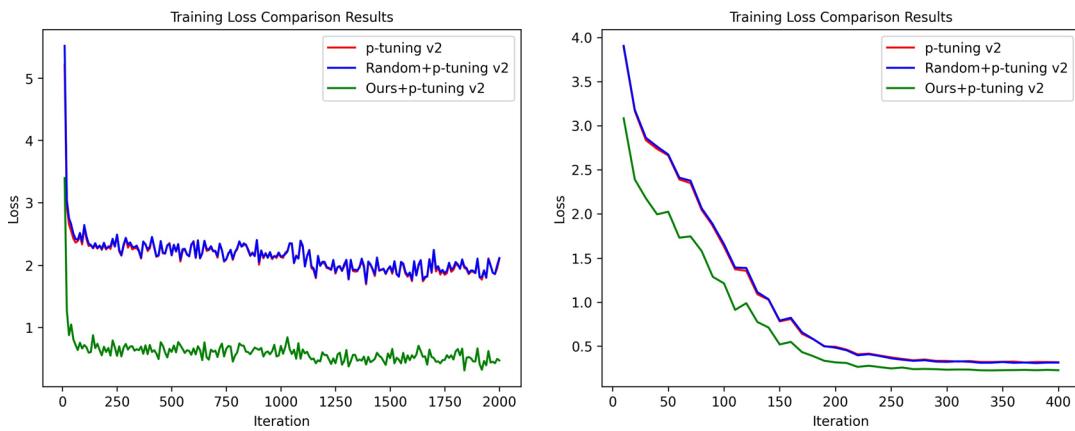


图 4: KgCLUE 数据集 (左) 和 Electric Power QA 数据集 (右) 损耗曲线的比较

结论

在本文中，我们提出了一种方法，通过知识图谱提示增强大型语言模型，用于电力知识问答任务。这种方法简单而有效，可以提高模型答案的准确性，并在一定程度上解决了不准确响应的问题。我们使用了电力 QA 数据集和知识图谱 QA 数据集进行了一系列实验。实验结果表明，所提出的方法直接有效，使模型能够从三元组中提取有价值的信息，并提高答案的准确性。

然而，这项工作仍存在一些局限性。例如，当前的匹配方法使用了 GPT-3.5 模型，训练一个专门的匹配模型以提高匹配的准确性将是有益的。此外，本实验中使用的微调方法和大型语言模型可以替换，可以探索不同的微调技术和大型语言模型，以验证这种方法的有效性和普适性。

引用

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. <https://doi.org/10.1145/3065386>
- [2] He, Kaiming, et al. Deep residual learning for image recognition. 2016. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [3] Silver David, Huang Aja, et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489, 2016. <https://doi.org/10.1038/nature16961>
- Ashish Vaswani, Noam Shazeer, et al. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS’17). Curran Associates Inc.,

Red Hook, NY, USA, 6000–6010.

[4] Tom B. Brown, Benjamin Mann, et al. 2020. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS’20). Curran Associates Inc., Red Hook, NY, USA, Article 159, 1877–1901.

[5] Heaton, J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genet Program Evolvable Mach 19, 305–307, 2018. <https://doi.org/10.1007/s10710-017-9314-z>

[6] Dhruv, P., Naskar, S. 2020. Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review. In: Swain, D., Patnaik, P., Gupta, P. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1101. Springer, Singapore. https://doi.org/10.1007/978-981-15-1884-3_34

[7] Chen, Zhe and Wang, Yuehan and Zhao, Bin and Cheng, Jing and Zhao, Xin and Duan, Zongtao. 2020. Knowledge Graph Completion: A Review. IEEE Access, vol. 8, pp. 192435-192456, 2020, doi: 10.1109/ACCESS.2020.3030076.

McHugh, Justin, et al. Integrated access to big data polystores through a knowledge-driven framework. 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017.

[8] Tang, Yachen, et al. 2019. Enhancement of power equipment management using knowledge graph. 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia).

Alec Radford, Jeffrey Wu, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[9] Liu, Xiao, et al. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Dublin, Ireland. 61–68. doi :10.18653/v1/2022.acl-short.8

[10] Li, Xiang Lisa, and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. 2021. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 4582—4597.doi: 10.18653/v1/2021.acl-long.353

[11] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, Jie Tang. 2021. GPT understands, too. arXiv preprint arXiv:2103.10385.

Du, Zhengxiao, et al. 2022. GLM: General language model pretraining with autoregressive blank infilling. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:

附录 5 外文翻译

Long Papers). Dublin, Ireland. 320–335

[12] Wayne Xin Zhao, Kun Zhou, et al. 2023. A Survey of Large Language Models. arXiv preprint arXiv:2303.18223.

[13] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient Trans- formers: A Survey. ACM Comput. Surv. 55, 6, Article 109 (June 2023), 28 pages. <https://doi.org/10.1145/3530811>

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA.

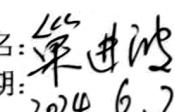
[15] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain.

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic.

燕山大学毕业设计（论文）指导教师意见书

学院：信息科学与工程学院

专业(方向)：计算机科学与技术

题目名称		大模型增强知识图谱的医疗问答研究与应用				
学号		202011040104	姓名	刘泽广	班级	计算机 20-4
序号	评审项目	指标			满分	评分
1	项目过程的管理意识 5分	(2分) 1.1 能独立查阅文献和调研；能较好地写出开题报告和中期报告，项目实施过程中各要素组织基本齐全、合理； (3分) 1.2 有综合、收集和正确利用各种信息及获取新知识的能力，能考虑非技术指标因素。			5	4
2	自我学习意识和工作态度 5分	(2分) 2.1 参加了毕业设计（论文）的各个训练环节，按期独立完成规定的工作任务，可以很好的综合运用所学专业知识解决遇到的问题；能够独立查阅相关资料内容，可以很好的解决实际问题。 (3分) 2.2 遵守纪律，学习认真；尊敬指导教师，能够听取指导教师的意见和指导；不存在学术不端行为。			5	4
是否同意参加答辩： <input checked="" type="checkbox"/> 同意 <input type="checkbox"/> 不同意					总分	8
评语：						
 指导教师签名：  日 期：2024.6.2						

燕山大学毕业设计（论文）评审表（评阅人用）

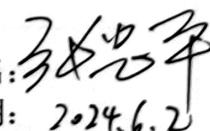
学院：信息科学与工程学院

专业（方向）：计算机科学与技术

题目名称		大模型增强知识图谱的医疗问答研究与应用				
学号	202011040104	姓名	刘泽广	班级	计算机 20-4	
序号	评审项目	指标			满分	评分
1	报告和论文的撰写质量 10分	(5分) 1.1 逻辑结构。毕业设计（论文）主题基本明确、结构基本合理、有一定层次性，引用资料基本准确，附录材料与毕业设计（论文）基本相符。 (5分) 1.2 写作能力。毕业设计（论文）语言基本流畅，整体内容能够体现设计主题；文字表述、语法应用、书写格式、图表注释、缩略词等基本符合国家相关标准；参考文献标注基本规范、来源清楚。			10	8
2	外文资料阅读与翻译 5分	(2分) 2.1 阅读能力。外文原文与毕业题目相关，来源于计算机领域期刊或者会议、不少于6页。 (3分) 2.2 翻译能力。中文翻译与外文原文对应，翻译准确、流畅。			5	4
3	创新和发展意识 5分	(2分) 3.1 创新意识。设计能够体现一定的创新思路。 (3分) 3.2 发展意识。设计方案正确，设计方法得当，体现一定分析解决本专业实际问题的综合设计能力。			5	4
					总分	16

评语：

论文撰写规范。

评阅人签名：
日期：2024.6.2

燕山大学毕业设计（论文）答辩评审表

学院：信息科学与工程学院

专业（方向）：计算机科学与技术

题目名称		大模型增强知识图谱的医疗问答研究与应用				
学号	202011040104	姓名	刘泽广	班级	计算机 20-4	
指导教师姓名		巢进波	职称	讲师		
序号	评审项目	指标			满分	评分
1	开发工具运用能力 10分	(5分) 1.1 设计能力。可以很好的应用开发工具设计符合国家或行业标准的毕业设计(论文)，体现一定分析解决本专业实际问题的综合设计能力，基本达到选题的目标要求。 (5分) 1.2 系统能力。体现出较为扎实的专业基本知识，基本能够综合运用开发工具解决毕业设计(论文)相关问题。软件系统能够正常运行，数据正确，代码命名规范、缩进风格统一，能够准确的无误的讲解代码的运行逻辑。			10	/0
2	工作总结和成果展示 10分	(5分) 2.1 工作总结。答辩表述清楚，语言基本流畅，整体内容能够体现毕业设计(论文)主题；体现出较为扎实的专业基本知识，核心概念基本明确。 (5分) 2.2 成果展示。态度认真，对完成的毕业设计(论文)内容完全掌握；回答问题有理有据，基本概念清楚，主要问题回答准确，有一定深度。			10	9
						总分 19

评语：

该生研究，设计完善。

答辩委员会小组成员

姓名职称（签名）：刘泽广 教授
 姓名职称（签名）：巢进波 教授
 姓名职称（签名）：陈东平 讲师

姓名职称（签名）：
 姓名职称（签名）：
 姓名职称（签名）：

答辩委员会（小组）负责人签名：

2024年6月3日

燕山大学毕业设计（论文）评分表

开题考核：

开题考核（满分 20 分）			
当前社会发展现状的认识 (5 分)	专业领域技术发展现状的调研 (5 分)	问题分析和解决方案 (10 分)	开题成绩

中期考核：

中期考核（满分 30 分）			
设计方案的可行性分析 (8 分)	问题研究能力的评价 (10 分)	系统设计能力的考核 (12 分)	中期成绩

燕山大学毕业设计（论文）评分表

指导教师评分：

导师评分（满分 10 分）		
项目过程的管理意识 (5 分)	自我学习意识和工作态度 (5 分)	导师评分

指导教师签字：_____ 年 月 日

毕业论文答辩：（满分 40 分）

报告和论文的撰写质量 (10 分)	外文资料阅读与翻译 (5 分)	创新和发展意识 (5 分)

开发工具运用能力 (10 分)	工作总结和成果展示 (10 分)	答辩成绩

答辩组组长签字：_____ 年 月 日

综合其开题成绩、中期成绩、导师评分、答辩成绩，

该本科生毕业设计(论文)的总成绩为：_____。

(A+ A B+ B C+ C D+ D F)

