专 业 学 位 硕 士 学 位 论 文

基于知识图谱的人工智能领域知识问答系统

Knowledge Question Answering System in Artificial Intelligence Field Based on Knowledge Graph

作 者 姓 名：郭振文

工 程 领 域：软件工程

学　　号：41917047

指 导 教 师：单世民

完 成 日 期：2022 年 6 月

大连理工大学

Dalian University of Technology

大连理工大学学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目：_____

作 者 签 名 ：_____ 日期：_____年____月日

摘　要

问答系统是信息检索系统的一种高级形式，它能用准确、简洁的自然语言回答用户用自然语言提出的问题，其研究兴起的主要原因是人们对快速、准确地获取信息的需求。目前较为流行的技术路线主要有基于关系型数据库的方法、基于标准问答对检索式的方法和基于知识图谱的方法。相对于基于关系型数据库的方法在多表联合场景中存在的性能略差问题，基于标准模板问答对检索式的方法需要大量的人工成本用于构建数据集，基于知识图谱的方法可有效提高搜索质量、提升搜索准确度，与其他两种技术路线相比具有较好的性能，且无需人工构建标准模板问答对，受到人们的广泛关注。同时，人工智能技术作为近些年的热点话题，随着其相关数据不断增长，检索用户目标信息的难度也逐渐增加，因此，本文结合项目组在知识图谱方面的工作积累，开展了以人工智能领域数据为依托，基于知识图谱的问答系统研究。

本文重点关注了基于知识图谱进行问句解析方法的改进，提出了将基于预训练模型与知识图谱相结合对问句进行中心实体提取和子图路径排序的问句解析方法。本文方法相对于基于模板的问句解析方法，能减少大量人工构建模板工作，且具有较高的准确率。如此，通过将知识图谱技术与领域知识相结合实现领域知识问答，达到方便用户对当前领域知识高效利用的目的。

进一步的，本文以所提出的问句解析方法为核心，通过分析实际需求场景设计并实现了基于知识图谱的人工智能领域问答系统，并在清晰、准确、友好的可视化用户反馈方面进行了优化。系统共分为问句答案检索、图表构建、结果可视化、图谱数据管理、同义词维护和用户管理六个模块。其中，问句答案检索模块基于问句解析方法获得直接的问句答案；结果可视化模块通过将关联关系可视化展示的方式将答案知识条目的出处、相关实体及数据库中包含问句文字信息的相关语句汇总列出，使用户能一目了然的获取到问句所指向的答案及其相关信息；图表构建模块用于辅助可视化模块，使得结果展示更加详尽。同时，为保证系统数据的扩展性，本文设计并实现了图谱数据管理等数据维护相关模块，方便管理员能对数据库中的数据进行管理和维护。

本文面向人工智能领域构建了验证数据集，对比分析了本文提出问句解析方法的准确率及响应时间等指标，并对本文实现的问答系统进行了测试。通过对实验结果的分析可知，本文所提出的问句解析方法与基于模板解析的方法相比，在只增加少量响应时间的情况下可获得更高的准确率。目前，本文系统已通过测试，并已交付用户。

关键词：知识图谱；领域知识问答；命名实体识别；路径排序；可视化

Knowledge Question Answering System in Artificial Intelligence Field Based on Knowledge Graph

Abstract

Question-Answer System is an advanced form of information retrieval system. It can answer users' questions in natural language with accurate and concise natural language. The main reason for its rise is people's demand for fast and accurate information acquisition. Currently, the most popular technological routes are the relational database-based method, the standard question-and-answer pair-based method and the knowledge graph-based method. Compared with the relational database-based method, which has a slightly worse performance in multi-table federated scenarios, the standard template-based method requires a large amount of labor costs to build datasets. The knowledge graph-based method can effectively improve the search quality and accuracy, and has better performance than the other two technical routes, and does not need to build standard template question-and-answer pairs manually. It has received wide attention. At the same time, artificial intelligence technology as a hot topic in recent years, with its related data increasing, the difficulty of retrieving user target information is also increasing, therefore, this paper combines the project group's work accumulation in knowledge graph, and develops a question and answer system based on knowledge graph, which relies on the field of artificial intelligence data.

This paper focuses on the improvement of question parsing method based on knowledge graph, and puts forward a question parsing method that based pre-training model with knowledge graph to extract the central entity of the question and sort the subgraph paths. Compared with template-based question parsing, this method can reduce a lot of manual template building and has a higher accuracy. In this way, by combining knowledge graph technology with domain knowledge, domain knowledge questions and answers can be achieved, so as to facilitate the efficient use of current domain knowledge by users.

Further, this paper takes the question parsing method proposed as the core, designed and implemented a question and answer system in the field of artificial intelligence based on knowledge graph by analyzing the actual demand scenarios, and optimized the clear, accurate and friendly visual user feedback. The system is divided into six modules: question answer retrieval, chart construction, result visualization, graph data management, synonym maintenance and user management. The Question Answer Retrieval module obtains the direct answer based on the method of question parsing. The result visualization module summarizes and lists the origin of the answer knowledge entries, the related entities and the related

statements in the database that contain the text information of the question by visualizing the relationship, so that users can get the answer and related information that the question points to at a glance. The chart construction module is used to assist the visualization module to make the results more detailed. At the same time, in order to ensure the extensibility of the system data, this paper designs and implements data maintenance related modules such as graph data management, so that administrators can manage and maintain the data in the database.

This paper structured a validation dataset for the field of artificial intelligence, compared and analysed the accuracy and response time of the question parsing method proposed in this paper, and tests the question and answer system implemented in this paper. By analyzing the experimental results, we can see that the question parsing method proposed in this paper achieves a higher accuracy with only a small increase in response time than the template-based parsing method. At present, this system has passed the test and has been delivered to the user.

Key Words： knowledge graph; Domain Knowledge Q & A; Named entity recognition; Path sorting; visualization

目　　录

1 绪论

1.1 研究背景及意义

早在20世纪60年代初期，人们就提出了机器和人类一样使用自然语言智能地回答问题，即实现“人机对话”，这就是问答系统[1]。智能问答系统的基本需求是将用户的问句输入系统然后将对应的答案直接输出，和传统搜索引擎将问题分解为关键词汇后针对词汇检索的方式不同。问答系统在接收到问句后，利用自然语言处理相关技术结合算法和模型将问句分解，通过检索将问句答案直接输出，不像传统搜索引擎输出的是相关网页。所以智能问答系统可以更有效地解决用户提出的问题。

随着科学技术的发展，数字出版物的数目也随之不断增加，人工智能作为近些年的热点话题，与其相关的期刊文章数量也随之增长。对于新人研究员来说，整理获取多个体系间的联系可能需要较长的时间，面对海量的领域文章更是无从下手，这对于想要快速掌握某个体系的人们无疑是一个挑战。因此一个能解答研究人员对于人工智能研究领域问题的系统是很有必要的。

同时，随着互联网时代的到来，互联网中蕴含的数据量也在快速增长，虽然计算机的存储和运算能力都得到了大幅提升，但是大多数问答系统的人机交互方式一直都是传统形式：系统根据用户输入的问题给出一系列与问题相关的包含资源链接的检索结果供用户自行查找，用户使用这种方式使得获取知识的效率相对较低，并且这种检索方式无法深入理解用户问句的意图，因此不能给出准确的答案。在此背景下问答系统的功能被不断发掘。

问答相关技术是自然语言处理的一个重要方向，基于此技术的问答系统是集数据挖掘、深度学习等前沿技术于一体的信息检索系统，它的检索效率更高且返回的结果更精确。由于数据存储方式的不同，可以大概分为基于关系型数据库的方法、基于标准模板问答对的方法和基于知识图谱的方法。其中基于关系型数据库的方法可以根据用户输入的自然语言问句自动生成对应的查询语句，能为非计算机专业的人们在数据查询时提供极大的便利，但由于目前该技术发展还不是很完善，在多表联合应用场景下性能不佳，落地性较差；基于标准模板问答对的方法在问句场景有限的应用中表现很好，但是这需要人工在整理海量数据后得出一系列的标准问句和对应的答案，不仅有较高的人工成本，而且在问句过多的领域应用性不佳；基于知识图谱的方法可有效避免以上两类方法的弊端，且在保证数据质量的前提下具有较好的性能。

知识图谱在图书情报界称为知识域可视化或知识领域映射地图，是显示知识发展进程与结构关系的一系列各种不同的图形，用可视化技术描述知识资源及其载体，挖掘、分析、构建、绘制和显示知识及它们之间的相互联系，从而使得信息能被高效组织和管理[2]。在工程的角度来看，知识图谱像是一种信息管理的架构[3]，基于这种架构的问答系统可以有效利用其中的数据信息。在问句解析的过程中，需要很多与之相关的技术，比如实体链接、路径排序等。将基于知识图谱的问答系统与领域性知识相结合，可以更高效的利用知识图谱的专业知识，协助科研人员更好的进行工作，通过简单的查询操作快速得到目标答案。随着知识图谱技术的发展，基于知识图谱的问答技术也逐渐成为热点。

一个出色的问答系统需要具备两个要素：

（1）对于输入问句的准确理解：用户输入系统的问句是无法穷尽的，但是问句的含义大部分是相同的，所以系统对于问句的准确语义需要学习。

（2）优质的图谱数据：问答任务是图谱构建的下游任务，如果图谱本身质量较差，就会导致查询结果不理想。

知识图谱的概念被提出后，出现了很多大规模图谱，为问答研究提供给了很好的基础。基于知识图谱的问答系统即使用自然语言在图谱中搜寻匹配结果，但要使机器能够识别输入的语句不是一件容易的事，如何使得机器获取问句语义信息，如何根据获取到的信息连接实体间的关系都是具有挑战性的任务。

本文在已有领域图谱的基础上构建问答任务，能够使得系统识别输入问句的语义并给出准确答案以协助研究人员解决研究过程中产生的一些对于当前领域知识的疑问；同时根据已经检索的结果组织数据构建可视化界面，方便研究人员能够一目了然的了解到当前检索目标的相关信息，帮助研究人员更好的了解目标相关体系，推动工作的进行。

1.2 国内外研究现状

1.2.1 知识图谱研究现状

知识图谱的出现是互联网信息爆炸时代的必然结果，其迅速发展也得益于众多领域，诸如专家系统、语义网络、数据库和信息抽取等，是多领域交叉融合发展的产物。“知识图谱”一词在 2012 年由谷歌提出使用，它通过一些相互连接的实体以及它们的属性构成，其本质是表示实体间相互联系的语义网络，其中，将每个实体赋予唯一 ID 作为标识，属性作为刻画实体特征的描述，两个实体间使用关系做连接，由此形成一张巨大的网络，这种图模型可以使用 W3C 提出的资源描述框架表示。还可以将知识图谱与数学和信息科学等领域方法相结合，应用于问答、推荐等领域。

经过多年发展，知识图谱目前可以按类型分为两种：通用型和领域型。通用型知识图谱覆盖的范围广，存储的数据较为全面，面向的是全领域，与领域型数据库相比具有更大的广度，目的是为囊括现实世界的所有知识。较为出名的通用型数据库有 DBpedia[4]、YAGO[5]及 Freebase[6]等。

DBpedia 是一个很特殊的语义网应用范例，它从维基百科的词条里撷取出结构化的资料，以强化维基百科的搜寻功能，并将其他资料集连结至维基百科。透过这样的语义化技术的介入，让维基百科的庞杂资讯有了许多创新而有趣的应用，例如手机版本、关系查询、文件分类与标注等。YAGO 是包含超过 1.2 亿条三元组，覆盖领域包含电影、人物、城市、国家等，是一个具有高质量多源背景的知识库。Freebase 是一个由使用者共同合作组成的大规模知识图谱，到目前为止，Freebase 拥有大约两千万的实体，目前已经被谷歌公司收购。2012 年，考虑到维基百科中许多信息以非结构化的形式存储，进而引发知识存储不一致的现象，影响信息的检索与分析。针对这一问题，维基媒体基金会成立 Wikidata**Error! Reference source not found.**项目，此项目通过多语言表示相同的实体，目的是通过一种全新的数据管理构建方式克服以上维基百科存在的问题，到目前为止该知识库已经有近三千万的实体数量。

领域型知识图谱是针对某个领域的数据建立的知识库，与通用型知识图谱相比，它的覆盖范围更小，面向的对象为当前领域的工作人员，所以其知识质量更高。LinkedMDB 知识库是国外对于知识图谱与电影领域知识结合的优秀作品，其包含的三元组超过六十万，描述了电影、演员等其他相关知识。国内知识图谱与领域知识的结合在医疗领域较为流行，除了中医药平台提供的中医药知识图谱，还有与新冠病毒研究相关的知识库，其共包含 80 万个实体，可以为医学相关学者对新冠病毒的研究提供巨大帮助。

从当前知识图谱的发展来看，虽然知识图谱提出的初衷是提高搜索质量，但是其应用之广泛已经不止于此。当前基于知识图谱的应用逐渐走向各个领域，主要包括搜索、推荐、问答等场景，随着诸如智能问答、推荐系统、个人助手等基于知识图谱的技术在各行业应用的落地，各种基于知识图谱的技术研究工作也得到了大量学者的关注。

知识图谱的研究领域主要围绕数据存储、知识获取、实体链接[8]、本体融合、逻辑推理、问答系统等知识图谱的应用等方面，除此之外，在大数据环境和新基建背景下，数据对象和交互方式的日益丰富和变化，对新一代知识图谱在基础理论和关键技术方面提出新的需求，带来新的挑战。如面对大规模动态图谱图和进行表示学习和预训练模型、在复杂的数据源中如何更快更准地反馈用户的提问等。

1.2.2 问答系统研究现状

人工智能之父图灵于上世纪五六十年代提出了测试机器智能性的测试，自动问答的相关研究热潮也就此展开[9]。早期的问答研究局限于某个特性领域内，也就是专家系统，此类系统主要依赖规则和模板匹配，有代表性质的是由 Green 等人设计并实现的 BASEBALL 系统[10]，该系统能针对美国棒球联赛范围内的知识回答相关问题，是对领域性知识问答的一次良好实践，但这种方式对于信息的利用率不高，实用价值较低。随后对于如何降低问答系统开发的难度和成本，研究人员集中于领域性质的数据集和机器语言方面展开了研究。20 世纪 90 年代后，互联网发展迅猛，问答系统也进入了下一个研究阶段，首先是 2005 年后出现的一些社区问答系统（Community Based Question Answering, CQA）[11]，其中比较有代表性的系统有 Yahoo！Answer[12]等，随后出现的 MIT 的 START 系统、iPhone 的 siri 及微软 Cortana 等，标志着问答系统已经成为了人们工作娱乐的一部分。2011 年，由 IBM 组织研发的机器人 Waston 在一个智力比赛节目的对决中获胜，成为问答系统的里程碑事件。

近年来，伴随着用户对智能应用方面的强大需求，许多公司和机构如谷歌、百度等对获得的高质量数据，采用自动化或半自动化方法设计了一系列完备的知识图谱，同时机器学习与深度学习的发展，为智能问答奠定研究基础。

目前问答系统的主要形式有两种：生成式问答系统和检索式问答系统[13]，比如早期出现的小黄鸡聊天机器人，就是检索式问答系统的代表作品。一个优秀的知识库是检索式问答系统的核心，其主要思想是将问题和对应的答案存储在知识库中，待用户输入问句后与知识库中的问句做匹配得到最终答案。生成式问答系统不同于检索式问答系统，它是一种端到端的问句处理技术，基于深度学习等前沿学科，不需要事先准备好问题对应的答案，而是通过提前准备的相关语料进行训练，最终可以通过训练得到的模型预测问句对应的答案。

为方便非计算机专业人员对于数据的获取，逐渐出现了将自然语言转换为关系型数据库查询语句的需求，tableQA 相关理论也被相继提出，通过与深度学习相关理论的结合，在给定关系型数据库表的前提下，基本实现了单表查询语句生成的需求，但目前成熟的多表联合查询解决方案较少，此种场景的落地应用也有较大困难。

KBQA 特指使用基于知识图谱解答问题的系统。KBQA 实际上是 20 世纪七八十年代对 NLIDB 工作的延续，其中很多技术都借鉴和沿用了以前的研究成果。其中，主要的差异是采用了相对统一的基于 RDF 表示的知识图谱，并且把语义理解的结果映射到知识图谱的本体后生成 SPARQL 查询解答问题。KBQA 的核心问题 Question2Query 是找到从用户问题到知识图谱子图的最合理映射。

国内对于问答系统的研究起步较晚，与国外相比理论和技术都相对不成熟，原因是处理中文语料过程较为复杂，同时不能直接将国外先进的自然语言处理技术应用于中文。在国内自然语言处理领域的专家学者的积极探索下，国内的智能问答技术快速发展，不仅建立了丰富的中文知识库，还发展出了一系列中文处理技术，如复旦大学维护的CN-DBpedia、哈工大提出的语言技术平台 LTP 等，都是较为出色的成果。一些基于知识图谱的问答系统诸如对红楼梦人物关系的问答系统[14]、张楚婷[15]在旅游领域展开的问答系统、曹明宇等人[16]构建的肝癌问答系统等，都是国内问答系统研究发展的优秀作品。国内的一些高科技企业也在自然语言处理的问答领域取得了较多成果，已经落地实施的有百度旗下的“小度”、阿里巴巴旗下的“天猫精灵”、小米旗下的“小爱同学”等，其原理都是将检索式问答与生成式问答做结合进行问答工作。

基于知识图谱的问答系统一般分为以下三个步骤：

（1）系统接收用户输入的问句

（2）系统将接受的问句解析，理解问句意图

（3）构建查询语句，查询并返回查询结果

目前 KGQA 有以下三种主流处理方法：

（1）基于语义解析的方法[17]：该方法的思路是通过语义分析将自然语言表示为一种知识库能“看懂”的逻辑形式，进而通过知识库进行逻辑推理，得出答案。

在 QALD[18]测评任务中，TBSL 提出了一种基于语法树分析和关键词映射的问答方法。该方法首先标注用户提出的问题文本获取语法结构信息，以有无领域依赖将获取的结构分为两组，并在此信息之上进行语义解析，最后和预设模板做匹配得到的结果，完成对实体的标注，流程包括命名实体识别、实体与图谱的链接及实体排序，以此为依据生成多组可能的 SPARQL 查询，最后筛选查询集合，将正确答案返回。

（2）基于信息抽取的方法[19]：该方法首先提取出问句中的实体，然后根据提前设计的特征提取方法提取问句中的相关信息，通过向量化的方式将代表问句的特征向量输入分类器，结合已提取实体的子图和问题类别匹配准备好的模板查询结果。

Frankenstein[20]观察 60 多种 KBQA 系统后总结出了一套处理基于知识图谱的流水线框架，其架构基于四类核心模块：命名实体识别及消歧、实体关系映射、实体分类映射、查询语句构建，该流水线的四个模块将复杂的 QA 系统分解为细粒度的问题，形成了可插拔体系，可利用其他先进技术对其中的任一模块进行改进。

（3）基于向量建模的方法[21]：该方法首先粗略提取一些候选答案，然后将问句和答案全部转换为分布式表达，将得到的分布式表达输入建立的模型训练，使问题和当前

问题对应的匹配结果的得分尽量最高。使用时将问句和一系列候选答案输入模型，得分最高的候选结果为正确答案。

基于该方法下的典型例子为基于 Multi-Column CNN 的工作，该工作同时训练自然语言问句词向量与知识库三元组，将问题与知识库映射到统一语义空间。该工作针对知识库的特点，定义了答案路径、答案上下文和答案类型三类特征，每一类特征都对应一个训练好的卷积神经网络，以此计算问题和答案的相似度。

综合以上方法，检索式问答系统对于问答对数据集的依赖性较为严重，tableQA 当前落地性较弱，但 KBQA 可以在无标准问答对的基础上进行生成式检索，结合项目组的工作积累，本文选择了 KBQA 方向作为切入点构建问答系统。

1.3 研究内容

随着深度学习技术的发展，结合基于向量建模的方法，基于深度学习的 QA 细分为两个方向：将传统问答模块与深度学习相结合和完全基于深度学习的问答模型的研究。

传统的问答系统虽然在某些情景中具有比较不错的准确性，但在处理自然语言问句时对于问句的复杂性和歧义性理解力较差，需要较多的人工成本构建相关的问题处理机制，针对此类问题，深度学习可以直接用于对传统问答模块的改进，如命名实体识别、实体消歧等。DONG L 等人[22]使用深度神经网络对 KBQA 进行了一次探索，他首先使用深度学习的方法对问句进行解析，通过实体识别、实体链接等技术，将问句中的中心词与知识库做连接，生成了对应的向量查询图，然后根据查询图找到其在知识图谱上的映射，将映射向量排序后返回结果得分最高的子项作为问句的答案，取得了较好的成果。

本文主要在传统的 KBQA 流程的基础上加入深度学习方法，结合多个问答系统的长处，在已经构建的人工智能领域的知识图谱上做问答任务，以提高研究人员的工作效率，提升知识利用价值。本文主要工作如下：

（1）研究如何识别用户输入问句的意图，即识别用户需要查询的信息，这是问答系统基本且核心的功能，该功能主要包含以下两个部分：

① 如何准确提取问句中的主实体。主实体即问句中的中心词，确定了中心词才能通过实体链接去图谱中召回候选子图进行排序选择，若仅使用传统方式通过字符串匹配的方式提取候选词，导致主实体识别不准确，后面的一系列工作都会受到影响。工作通过多种方法在本领域数据集的验证后，确定使用 ALBERT+Bi LSTM+CRF 结合词典和启发式规则的方式进行主实体提取。

② 如何排序以保证留下最接近正确答案的候选子图。加入路径排序模块的作用在于提升问句解析流程性能和准确率，若执行了错误的排序策略将正确答案路径剪掉，会极大地影响查询结果的正确性。

（2）研究如何以友好清晰的可视化方式展示答案。若只是简单的将结果列出，固然达到了问答系统的基本要求，但是答案相关联的一些关键信息也是答案的一部分，对于研究人员分析体系结构也会有很大的帮助，本文可视化工作主要包括：

① 准确直接的问句答案、答案学习来源及答案相关信息。

② 部分场景根据问句答案的相关数据集合，使用提前训练的端到端模型选取恰当的图表类型进行数据展示，使得结果或数据趋势更加清晰明了。

（3）设计并开发结合以上两点研究的问答系统。将多个模块集成，才能更好的发挥各个模块的作用,因此本文设计并实现了基于知识图谱的人工智能领域知识问答系统，提升用户对问答系统的使用体验。

1.4 论文结构

论文共分为六个章节，总体结构如下：

第一章：绪论。该部分对问答系统的相关背景作了详细说明，结合实际阐明了当前系统构建的意义和目的，并调研分析了国内外问答系统发展情况级落地应用，最后阐述了本文的具体研究内容。

第二章：相关理论与技术。该部分主要针对问答系统构建所用到的技术和理论做详细介绍。

第三章：系统需求分析。主要从系统的实际需求出发，从功能需求和其他需求及可行性方面对系统进行分析。

第四章：系统设计。该部分为系统设计说明模块，主要讲解基于知识图谱的问句解析流程及系统各模块的具体实现细节。

第五章：系统实现。系统数据存储基于图数据库 HugeGraph 及关系型数据库，核心基于第四章的相关方法，经过对问答系统的需求分析后，完成了基本问答、图表构建、结果可视化、同义词维护、图谱数据维护及用户管理六个功能模块的开发

第六章：系统测试。对第五章实现的问答系统各模块进行测试。

2　相关理论与技术

本章节主要对系统构建过程中用到的主要技术和理论做详细介绍，章节主要分为两部分，第一部分依次对与文本处理相关的注意力机制、基于自注意力机制的 Transformer 模型及基于 Transformer 编码部分的 Bert 相关模型进行介绍；第二部分对数据存储部分用到的数据库进行介绍。

2.1　问答模块

2.1.1　注意力机制

上世纪九十年代，人们在研究人类视觉的过程中发现并提出了注意力机制，在当时它是一种信号处理机制，后来一些人工智能领域的专家学者将这一机制嵌入一些模型训练中，用来自动计算输入的相关数据在输出数据中权重占比的大小，并取得了很好的效果。目前，这一机制被广泛用于机器视觉及自然语言处理领域。

人们在获取所处环境的信息时，总是会先总览全局再着重关注自己关心的焦点区域，通过分析关注的区域得出进一步分析，对其他不相关区域的关注度明显低于焦点区域，这种机制是一种非常高效的信息处理方式，在人们的日常工作学习中起到了很重要的作用。像这样有选择性的处理所接收的所有信息的机制，被称为注意力机制[23]。

在自然语言处理领域，研究人员将注意力机制通过特征工程的方式加入机器学习任务，即提取原始文本的特征作为数值向量，并将提取的特征向量加入模型中，模型通过注意力机制与特征向量的结合，对于非重要的部分投入较少的资源，能有效高效地完成任务。注意力机制的本质思想如下图：

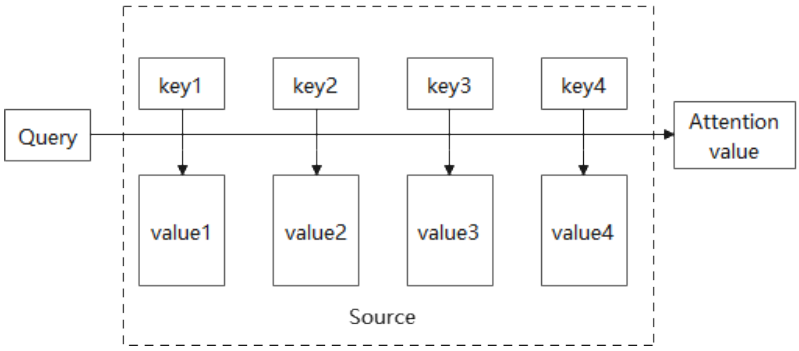

图 2.1　注意力机制的本质思想

Fig. 2.1　The essential thought of attention mechanism

对于图 2.1，我们可以将输入模型的文本 source 理解为一系列的键值对构成，其结构为<key, value>，此时要计算当前元素与其他元素的相关度，即计算 Query 与每个 key 的相关度，通过计算得到每个 key 对应的 value 权重，最后对 value 根据权重比求和，这样就得到了最终的 Attention 值。从概念上理解，注意力机制将少量有用的信息从大量数据中筛选出来，忽略其他不重要的信息，这个筛选聚焦的过程就是权重系数的计算过程，当前元素得到的权重越大，就证明它越重要。注意力机制的具体计算过程分为三个阶段，具体如图 2.2 所示：

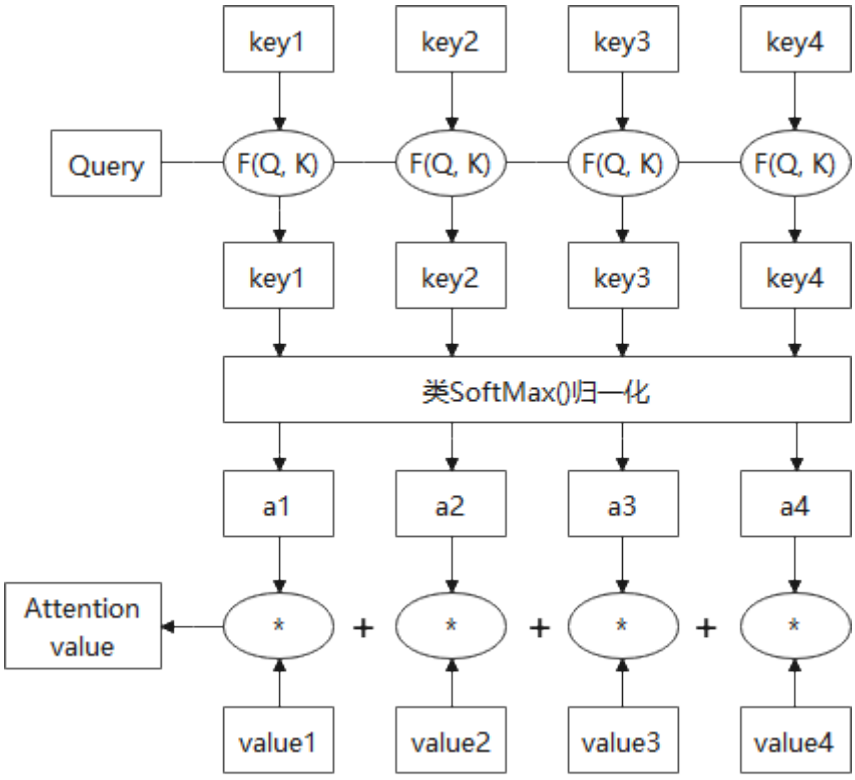

图 2.2　注意力机制计算过程

Fig. 2.2　Attention mechanism calculation process

第一阶段：计算给定序列的 query 与 key 的相关度，将 query 与 key 一同输入计算函数 F(Q, K)中计算得到相关性得分 Similarity，函数 F 最常见的方法有：求 query 和 key 的向量点积(公式 3.1)、求两者的 cosine 相似性(公式 3.2)或者接入额外的神经网络层(公式 3.3)。

$$\text{Similarity}\left(\text{Query}, Key_i\right) = \text{Query} \cdot Key_i \tag{3.1}$$

$$\text{Similarity}(\text{Query}, \text{Key}_i) = \frac{\text{Query} \cdot \text{Key}_i}{\| Query \| \cdot \| \text{Key}_i \|} \tag{3.2}$$

$$\text{Similarity}(\text{Query}, \text{Key}_i) = \text{MLP}(\text{Query}, \text{Key}_i) \tag{3.3}$$

第二阶段：引入类似 softmax 计算机制对第一阶段的结果进行转换计算，一方面可以对第一阶段的计算结果进行归一化，另一方面也可以更加突出重要元素所占的比例权重，本阶段计算公式见(3.4）：

$$a_i = \text{Softmax}(\text{Sim}_i) = \frac{e^{\text{Sim}_i}}{\sum_{j=1}^{Lx} e^{\text{Sim}_j}} \tag{3.4}$$

第三阶段：根据第二步求得的与 value 对应的权重分布，进行加权求和即可得到最终的注意力数值，公式如（3.5）所示：

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} a_i \cdot \text{Value}_i \tag{3.5}$$

由于循环神经网络对于序列中距离较大但是相关性很强特征的捕捉耗时长效率低，基于注意力机制的自注意力机制（self-Attention）由此出现。注意力机制一般是输入 Source 和输出 Target 之间形成的机制，发生在输出的 query 和所有输入元素之间，概括的说就是注意力机制的权重计算过程需要输出 Target 的参与，而自注意力机制并不是 Source 与 Target 之间的机制，而是 Source 或者 Target 自己内部的机制，也可以理解为 Target 与 Source 是相等的特殊情况的注意力计算。这一特性能够有效提升模型对于文本中长距离特征的挖掘，此外，自注意力机制还能使得模型的并行计算能力得到提升，因此得到各领域的广泛应用。

2.1.2 Transformer

2017 年，Google 机器翻译团队提出并发表了 Transformer 模型[23]，该模型完全抛弃了传统的 RNN 和 CNN，仅由自注意力机制（self-attention）和前馈神经网络（Feed Forward Neural Network）组成，模型采用 Encoder-Decoder 结构。

Transformer 模型的整体结构如图 2.3 所示，模型的输入由词向量嵌入和位置编码构成，之所以特别的提出加入词位置编码是因为模型不能获取词在文本中的序列位置，若不加入位置信息模型会丢失句子结构中的潜在特征。每个 Encoder 层由自注意力机制和前馈神经网络组成。数据经自注意力机制计算得出当前序列的注意力权重得分，输入前

馈神经网络。前馈神经网络由两部分组成：线性变换函数和 Relu 激活函数，数据先经线性变换函数向高维度映射后，通过激活函数再通过线性变换降低至原来的维度。模型共使用了 6 个 Encoder，为了避免梯度消失，每一层 Encoder 和 Decoder 都采用了残差神经网络结构。Decoder 虽然也是先进行自注意力得分的计算，与 Encoder 不同的是，在计算完自注意力得分后，将自注意力得分与 Decoders 的输出再计算一次注意力得分，然后再将计算结果输入前馈神经网络。

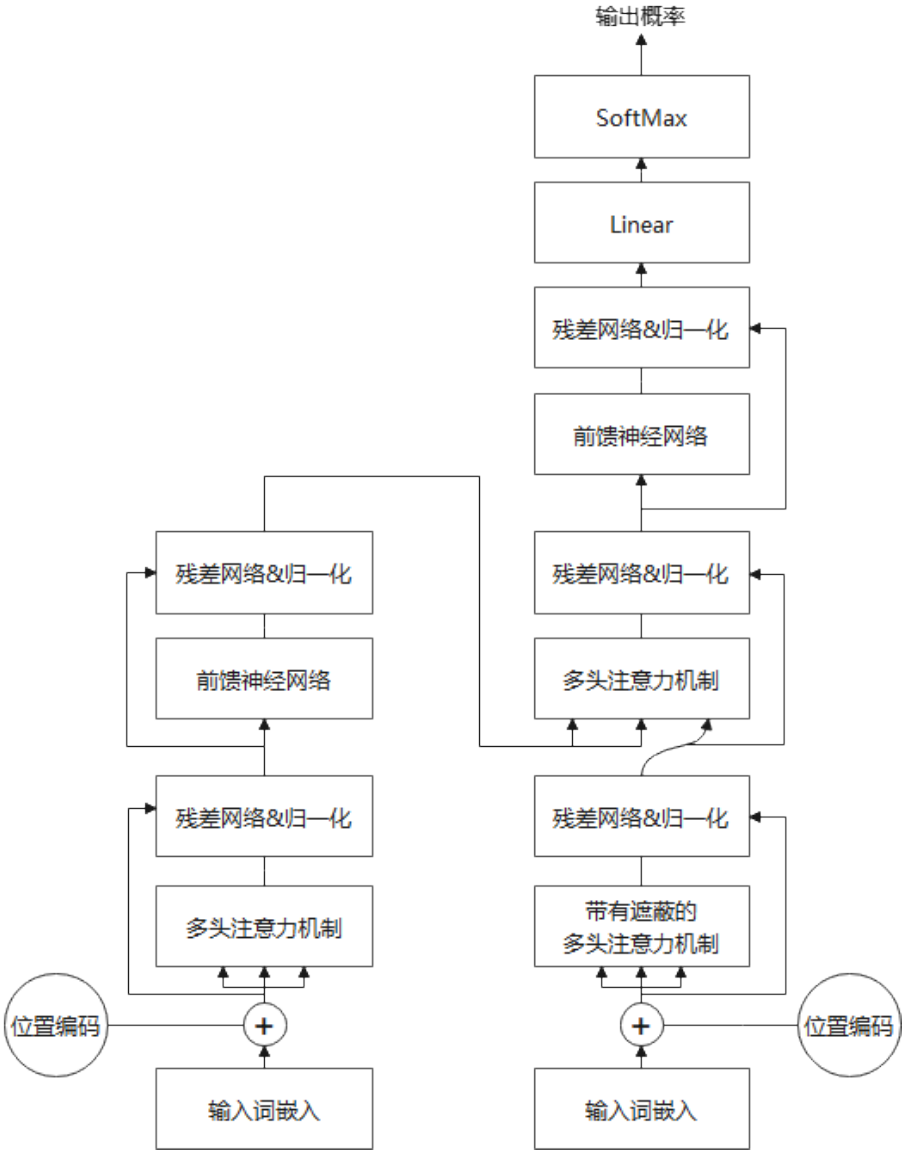

图 2.3　Transformer 的完整结构图

Fig. 2.3　Complete structure diagram of Transformer

2.1.3 BERT

BERT（Bidirectional Encoder Representations from Transformers）是在 2018 年由 Devlin 等[24]提出的预训练模型，作为 Word2Vec 的替代者，其在 nlp 领域 11 个方向取得了出色成果，成为近些年 nlp 领域众多最具突破性成果之一。

与传统模型只能单向预测且不能很好的对上下文信息加以利用不同，BERT 通过对输入文本序列信息的获取，可以对上下文有效信息进行充分利用。模型的设计结构基于上一小节提到的 transformer，主要使用了它的 Encoder 部分，模型的输入采用多个向量结合后的方式，与 transformer 的输入相似，都是将文本序列的字向量、句子表示和位置表示作加和，不同之处是 BERT 的位置表示不采用正弦函数的方式计算。尤其是在中文任务中，使用字符向量的方式使得获取上下文信息更加方便。BERT 模型的输入形式如图 2.4 所示。

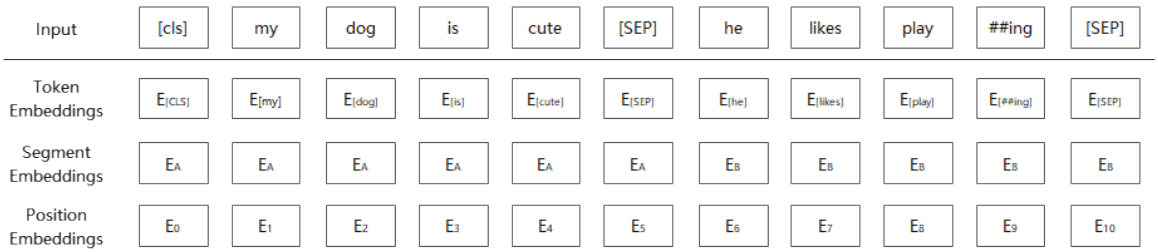

图 2.4 BERT 模型的输入形式

Fig. 2.4 Input form of BERT model

模型接收的输入数据的处理如上图，首先为输入序列的首位添加[CLS]标记，这是模型识别文本为句子类型的标志，为句子之间添加[SEP]标记，这个是告诉模型两个句子分隔的意思，用以区分输入不同文本序列的边界。

一般语言模型都是根据上文序列信息预测当前信息或者根据下文序列信息预测当前信息，BERT 希望能同时根据上下文序列信息对当前信息进行预测，因此 BERT 在训练过程中采用了 MLM(Mask Language Model)策略，即在训练过程中随选取输入句子 token 的 15%进行隐藏，然后将选取的信息再选取 80%随机替换为[MASK]符号，10%替换为其他信息，剩余的 10%保持不变，这个策略使得 BERT 能够充分学习到上下文信息，并使其内部参数得到充分训练。

很多下游任务都是基于两个句子之间的关系为出发点，基于此问题，BERT 为了增强模型对句子之间关联度的判断力，选择的训练数据为两个句子的拼接序列，这些序列中有真正相连的句子对，也有随机拼接的句子对，并采用一半正样本（IsNext）一半负

样本（NotNext）划分正负样例的策略，这个关系保存在[CLS]符号中，用以判断第二句是否为第一句的关联文本，最后使用二分类方法判断句子之间的关联度。

2.1.4 ALBERT

虽然 BERT 的功能强大，作为预训练模型的代表能参与参与多种任务，在模型复杂程度较低的前提下，模型的效果会随着参数的增多而提升，但是达到一个复杂程度的临界点之后，随着参数量的继续提升，模型的训练时间加长效果反而有所下降。为解决这个问题研究人员提出一些对于 BERT 的改进模型，本文设计的系统所使用的 ALBERT 是由 Lan 等人[25]针对 BERT 参数量过多而提出的轻量级模型，从名字可以看出，该模型主要关注模型的轻量化，除此之外还优化了模型的半监督学习，提升了模型的训练速度，降低了模型的训练参数。相似的是，ALBERT 和 BERT 一样提出了不同参数的多种模型，如表格 2.1 所示。

表 2.1 ALBERT 不同版本的参数列表

Tab. 2.1 List of parameters for different versions of ALBERT

模型	参数量	层数	隐藏维度	词嵌入维度
$ALBERT_{BASE}$	12M	12	768	128
$ALBERT_{LARGE}$	18M	24	1024	128
$ALBERT_{XLARGE}$	59M	24	2048	128
$ALBERT_{XXLARGE}$	233M	12	4096	128

ALBERT 在 BERT 基础上共使用了 Embedding 因式分解参数化、跨层参数共享和句子间一致性预测三种策略对训练参数过多问题进行优化。

Embedding 因式分解参数化主要针对 BERT 的输入层词向量进行因式分解。从设计结构来看，词典大小为 V，嵌入词向量矩阵大小为 V*E，BERT 对于词向量嵌入层矩阵 E 设计与隐藏层 H 的大小相同，其策略是将输入的词嵌入使用独热编码映射到隐藏层，在训练中若 H 增大则 E 必须增大。ALBERT 对这种策略进行了改进，先对输入的词嵌入进行运算，将其压缩到一个低维度空间，将得到的低维度向量输入隐藏层，这样就能使得词向量嵌入层的参数规模由 O(V*H)消减到 O(V*E+E*H)。

跨层参数共享主要对 BERT 的参数共享机制进行优化。BERT 对于参数的共享策略是仅共享前馈神经网络层和注意力层的参数，ALBERT 将所有层参数的参数全部共享，并使用余弦相似度和计算 L2 距离来保证跨网络层共享参数的稳定性。因式分解参数化和跨层参数共享策略使得模型训练速度提升了一倍以上。

句子间一致性预测主要对 BERT 的下一句预测任务进行改进。在 BERT 模型中，针对两个句子下一句是否为上一句的承接预测任务提出了标签分类的策略，IsNext 标签为正样本，说明句子来自同一片段，NotNext 为负样本，说明句子对不相关。ALBERT 针对该任务采用和 BERT 相同的正样本采样，不同的是负样本采取连续句子片段前后交换的策略，使得模型学习到了更多详细的文本语义信息，能提升 2%的准确率。

2.2 数据存储模块

2.2.1 HugeGraph

图数据库(Graph Database)是一种使用图数据结构存储数据的数据库，通过节点、属性和边表示存储数据间的关系。经过对图谱数据的研究，最终系统选定百度团队研发的 HugeGraph 作为图谱数据的承载工具。HugeGraph 的技术架构图如图 2.5 所示。

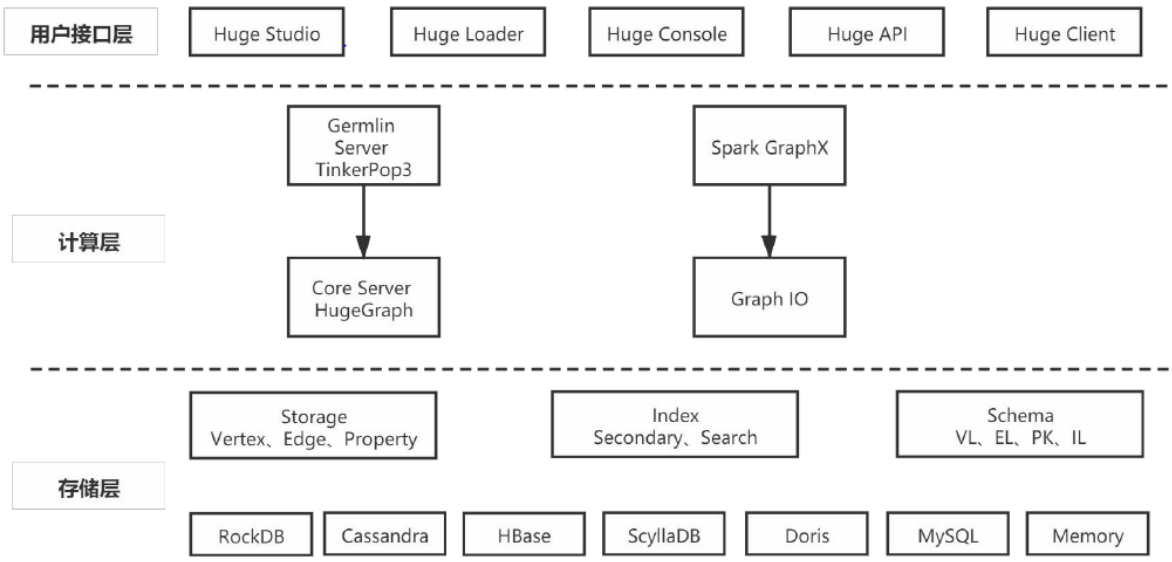

图 2.5 HugeGraph 的技术架构图

Fig. 2.5 Technical architecture of HugeGraph

HugeGraph 支持海量数据的快速导入，最高规模可达百亿以上，并提供毫秒级的数据查询支持（OLTP），而且可以被集成在大数据平台以提供离线分析的部分功能（OLAP），且支持 Gremlin 查询语言，为开发人员提供了丰富灵活但不失标准的图查询规则，支持多种后端引擎，并能使用插件的方式拓展新的后端引擎，支持快速的批量导入、批量导出功能，同时用户可灵活定义导入导出格式，支持 CSV、TXT、JSON 等

格式，支持从 HDFS、MySQL、SQL Server、Oracle、PostgreSQL 等数据源直接导入数据。

2.2.2 Elasticsearch

Elasticsearch[26]是当前流行的搜索和数据分析引擎之一，能够解决多种数据处理方面的用例，是分布式且具有 RESTful 风格的工具。该引擎基于 Lucene 开发，除了提供对海量数据近乎实时性的搜索功能，引擎还提供了对极大数量数据的存储和索引功能，是目前常见的分布式搜索引擎代表，一些知名网站如 github 等都采用了 Elasticsearch 作为基础搜索框架。

该引擎基于 Lucene 开发，不但具有 Lucene 的特性，还通过高级操作接口的封装免去了 Lucene 的复杂性，开发人员仅通过简单的 Restful API 与引擎交互，十分便捷。引擎分布式通过使用底层分片机制来实现，通过分片机制将逻辑上的索引分割成不同分布的分片，就能达到对数据分布式操作的目的。

3 系统分析

对系统的流程和实现细节进行清晰地梳理和阐述，既是软件工程学科的基本要求，也是系统开发的必要准备和关键前提。本章节对基于知识图谱的领域性问答系统的构建流程作详细说明，系统主要解决的问题是研究人员在面对海量文本时无法高效的利用其中的知识，系统基于此问题针对性的给出对文本中蕴含知识的问答功能，从而使得系统用户能够快速获取到想要的知识。

3.1 系统功能需求分析

本文设计的基于知识图谱的领域性知识问答系统主要功能是为广大科研工作者或普通用户提供可以便捷查询对于人工智能领域问题的问答系统，用户仅需输入自然语言问句提问即可，无须进行复杂的选择查询规则或输入查询语句，系统检索出问句答案后，不仅会直接给出对应的答案，并且同时会给出答案对应的关联实体、答案学习来源及答案在其他文本出现的位置和信息，除了文本显示结果，系统还会提供图谱可视化显示结果，可以使用户对于结果的相关信息更加一目了然。对于范围性的问句查询，系统还会给出对应的结果集合数据图表，包含问句查询结果的属性类比较，使得用户可以快速得出查询结果间的关系和发展趋势等其他结论。另外，本文实现并实现的问答系统是基于已经构建的领域知识图谱下游任务，随着科研进程的发展，新知识会不时地涌现，为了避免系统查询不到新知识的局限性，还需要知识导入模块对图谱进行更新，使用户能查询到最新知识。基于对问答相关文献的调研及上述需求的整理，系统的整体用例图如图 3.1 所示。

系统管理员拥有系统管理的最高权限，不仅能够使用系统的全部功能，还能进行用户权限管理操作。系统管理员的用例包括知识图谱数据管理、用户管理及系统维护，具体用例图如图 3.2 所示。一个优质的领域知识图谱是完成问答任务的必要前提条件，同时也不能因为不能更新数据库导致一些新知识无法被检索到，所以对于图谱的更新和维护是非常有必要的。

本文系统将知识图谱的数据库模式抽象为数据层和模式层。模式层代表了图谱的顶层结构设计，规定图谱包含实体类、关系类和属性类三大类别。实体类是一类事物的代表统称，例如：“水果”就是单个类别的代表，是所有水果实例的统称。关系类是实体类之间连接的类型统称，包含关系类的名称和连接方向，关系类和关系有相同的名称，但是关系类主要用于实体类之间，关系用于实体之间。属性类是属性的类别统称，二者

名称相同但属性值不同，实体类和属性类都可以关联属性类，例如：实体类“科研人员”可以关联属性类“身高”，关系类“所属院校”可以关联属性类“组织机构”。

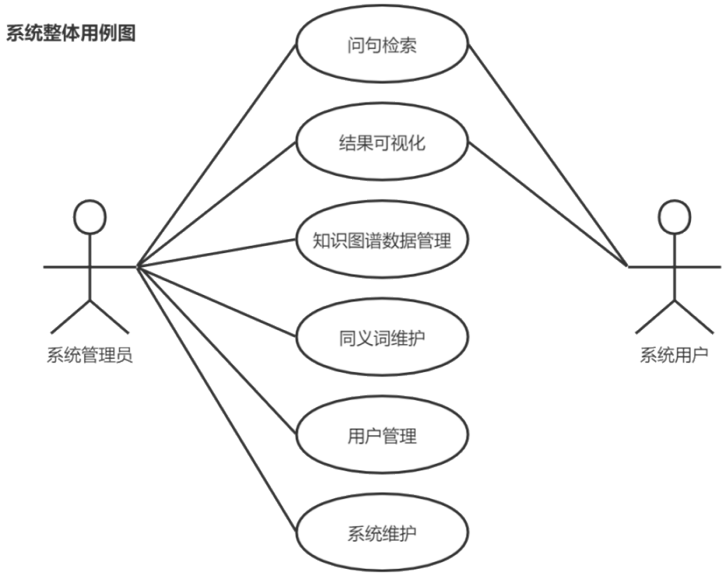

图 3.1　系统整体用例图

Fig. 3.1　System use case diagram

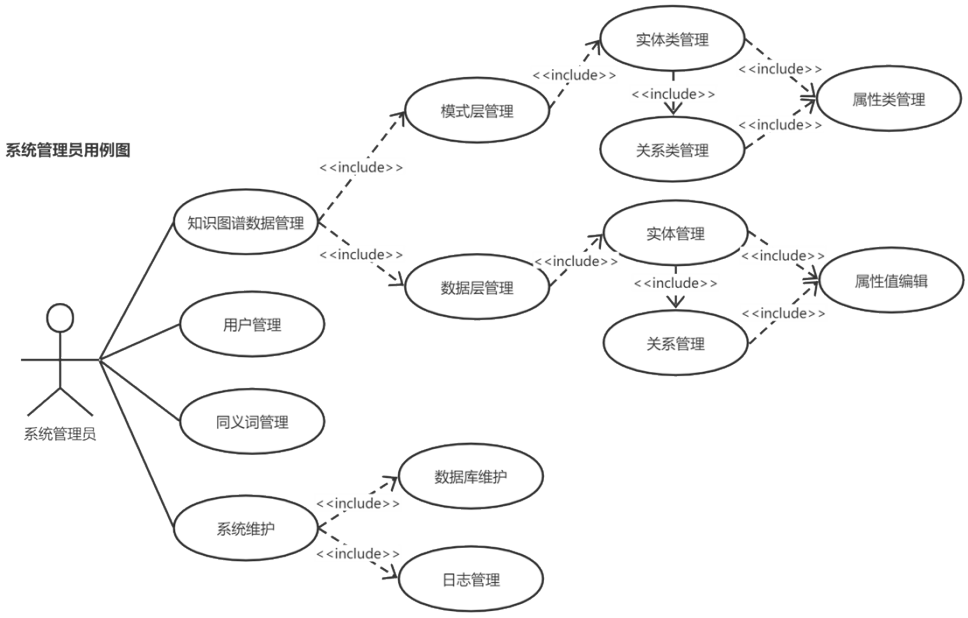

图 3.2　系统管理员用例图

Fig. 3.2　System administrator use case diagram

通过实体类为出发点来管理模式层，可以通过关系类很好地规范实体类之间、实体类和属性类之间的关联关系，从而使得实体类下所属的实体组成内容有一个很好地体系构建。同时实体类也要考虑层次结构，即大类可以包含子类，既可以新建根实体类又能新建子实体类。

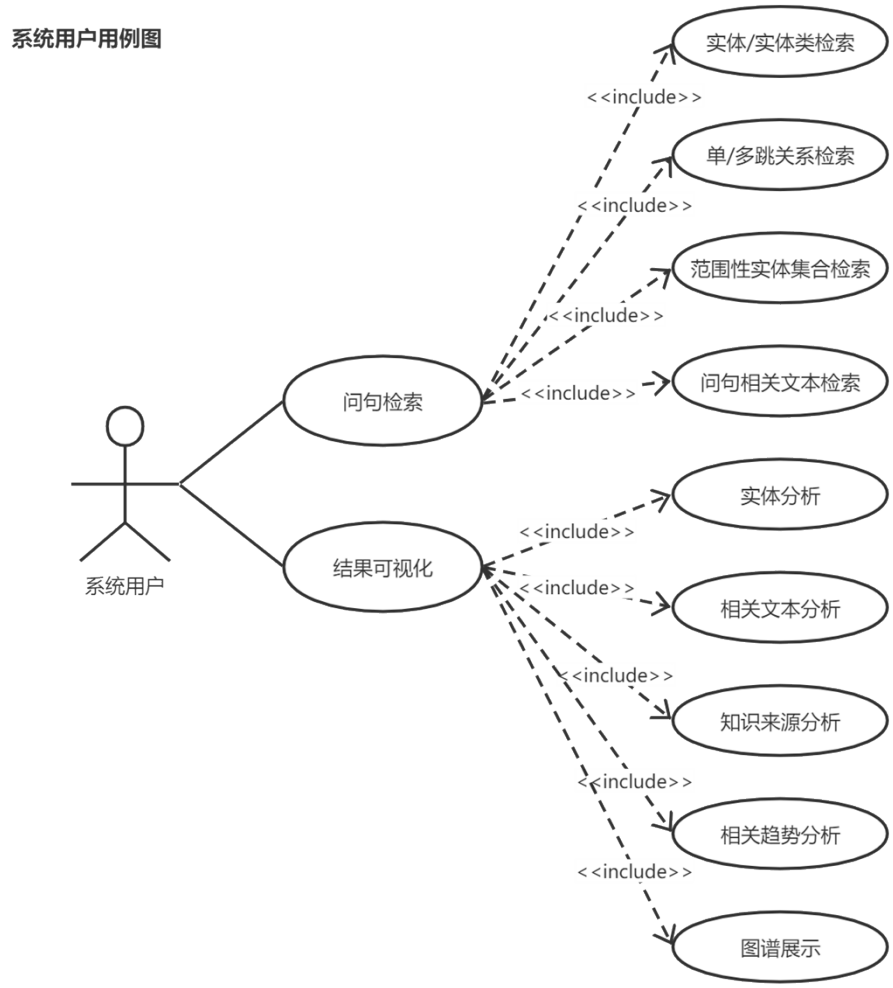

图 3.3　系统用户用例图

Fig. 3.3　System user use case diagram

系统用户用例图如图 3.3 所示。系统用户是本系统的主要使用人员，可以使用除知识图谱数据库维护、系统维护、同义词维护及用户管理之外的全部功能。系统用户关联的两项用例分别为问句检索和结果可视化，其中问句检索包含实体/实体类检索、含单/

多跳关系的问句检索、范围性实体检索和与问句相关联的文本检索，结果可视化包含实体分析、相关文本分析、知识学习来源分析、答案相关趋势分析和图谱展示。

3.2 系统非功能需求分析

除去功能需求分析以外，非功能需求分析也是系统分析的一个核心组成部分，和功能需求为互补式关系，为保证系统正常运行而存在，从各个角度考量系统能否长期稳定的为用户提供良好的服务环境，所以在系统设计的过程中，不仅要满足功能性需求，还要考虑非功能性需求。

3.2.1 性能需求分析

性能优劣是衡量系统质量的一个必要标准，在完成系统功能需求的前提下，应尽可能的提高系统的性能。以系统架构为出发点，系统性能的优劣可能会影响刀前端页面布局、对于检索结果的展示方式、后端的数据接口处理等模块；以系统技术为出发点，通过分析系统性能需求可能会根据所要处理数据量的规模影响前后端技术选型、数据库的设计及服务器部署等相关功能。

本文设计的问答系统主要供个人用户使用，问答系统对于检索结果的快速响应性能要高于其他系统，因此更要考虑系统响应的实时性。因涉及到图表检索功能，会有较大规模的数据量需要处理，因此可以先展示能快速检索到的实体等信息，而不用等全部信息都返回后再渲染页面，减少页面白屏时间。

3.2.2 可靠性需求分析

系统可靠性分为硬件可靠性与软件可靠性。

硬件可靠性是系统正常运转的基本保障，硬件出现的问题可能会导致系统运行环境的崩溃，因此系统需要一个能正确执行系统指令且能全天候正常运行的硬件环境，即使某一个硬件环境出现问题，也要有备用机制保证系统的正常运转。

软件可靠性指系统要具有一定的容错性和恢复性，且能保护系统数据的安全。对于一些用户输入的错误指令或不合规范的操作，系统要能给出合理提示，协助用户理解问题并解决问题，同时也要提供回退机制，即使有批量错误操作，也能恢复到之前的某个时间节点，保证系统运行。在开发过程中，也要做到必要的模块分离，以便在出现故障时，工程师能快速定位故障模块并解决问题。

3.2.3 可扩展与可维护需求分析

可扩展性指在系统后续功能开发的过程中，在不影响之前已经开发功能的基础上添加心得功能模块。因此本文系统前端采用模块化开发模式，后端以服务为单位进行划分，尽量减少模块间的耦合程度，以提高系统的可扩展性。

可维护性不仅包括可扩展性，同时也包括对于旧模块的更改和功能修正，包括模块化维护、复用性维护。可维护性的基本出发点是系统代码的可解释性，要求系统代码清晰易读，结构明了；模块化和易用性利用设计模式的思想，可以节省后期的修改优化时间，且减少代码量，提高工程师的工作效率。

3.2.4 易用性分析

易用性指在用户使用系统的过程中以尽量少的易理解的操作达到使用目的。本文系统仅需用户输入自己的问题即可给出合理答案，方便用户将更多的精力放到对于疑问检索到的答案，达到了易操作的根本目的。

3.3 可行性分析

系统的可行性分析是系统设计和开发的出发点，不仅要考虑时间成本、资源问题、利润因素，还要考虑系统的当前市场和用户群体，当前市场是否已存在已经能够满足当前面向用户群体的完善系统体系，当前技术路线是否支持系统的全部开发工作，系统开发完成后是否能带来盈利目的或社会效益等。

3.3.1 市场对比分析

当前市场的搜索系统主流为根据用户输入的问题给出一系列与问题相关的答案链接，谷歌和百度已经尝试在搜索系统中加入知识图谱构建体系来直接回答用户的问题，取得了很好的反响。目前市场仍然没有针对人工智能领域的智能问答系统，因此一个能针对用户问题直接给出答案的问答系统可以很好的接入这片相对空白的市场。

3.3.2 技术可行性

系统开发前端使用 vue 框架，后端使用 java 语言，数据存储使用 HugeGraph 和 Elasticsearch。vue 框架简单易学，是近几年前端开发的三大流行框架之一，开发过程中不仅对开发者十分友好，而且还使用了虚拟 dom 算法，能够避免浏览器重复渲染相同元素，提升了加载效率，能提供优秀的用户体验。java 是上世纪九十年代出现的面向对象的高级编程开发语言，经过三十余年的完善，其生态已经非常健壮，另外，java 运行在虚拟机上，其可移植性非常强大，同时 java 能够为系统提供强有力的 web 交互性能，提升用户体验。HugeGraph 是一款百度推出的开源图数据库，支持百亿以上数量级的数

据导入，对于以后系统的数据拓展提供强有力的支持，而且它的查询响应速度极快，复合问答系统需求的基本特性。Elasticsearch 是一个分布式多用户能力的全文搜索引擎，基于 java 开发，有极优秀的检索实时响应性，非常契合问答系统的文本检索需求理念。

另外，系统采用 Tomcat 轻量级服务器作为服务平台，框架使用 SpringBoot 将 Spring 集成在同一个系统中，且 SpringBoot 内嵌 Tomcat 服务，避免了一系列因环境不同或配置不同导致的各种不能预料的问题。

综上所述，系统有良好的技术可行性。

3.3.3 经济可行性

本文设计的系统使用实验室电脑及服务器作为开发设备，模型训练使用实验室提供的含 GPU 处理器的服务器进行运算训练，开发过程使用的网络环境均为校园网，所用技术均为免费开源，训练数据为实验室同学标注。

本问答系统第一版开发完成并上线后，后期需求迭代所需环境不变，维护费用较低，能为广大科研工作者解答人工智能领域问题的大部分问题，有较好的社会效益。

4 系统设计

本章节为问答系统设计部分，主要借助自然语言处理的相关主流技术实现对系统接收问句的解析，以达到理解问句语义并给出问句结果的最终目的。系统使用 ALBERT-BiLSTM-CRF 模型配合词典及启发式规则实现对问句候选实体的提取，通过实体链接结合问句分类召回相关子图，并生成一系列候选答案路径，最后通过路径排序得到最终答案。

为了保证知识的更新，系统使用图谱数据管理模块接收文档形式的知识输入，通过基于预训练语言模型对文档进行自动解析后进行知识入库，同时引入同义词管理系统保证问答系统对问句信息提取的准确率。最后使用管理员数据维护机制，除管理员有权限管理数据外，只有经管理员批准的用户才有权限进行知识入库操作，保证入库数据的可靠性。

4.1 问句解析方法

传统的基于模板的问答系统通过定义一系列带变量的模板语句与输入的问句进行匹配，经过映射形成查询语句。这样做的最大优点是简化了问题分析的步骤，简单可控且解析时长短，能绕过复杂的语法解析，因而在工业界被广泛使用。

本文系统在设计之初也使用了基于模板的思想对问答模块进行了搭建，图 4.1 为之前基于模板搭建问答系统的一个问句匹配示例，该流程主要包含以下步骤：首先通过 jieba 分词工具加载图谱中的所有实体类、关系类、属性类、实体和关系，然后将问句通过预加载图谱知识的分词工具进行分词，得到带有标签的分词结果，最后再将分词结果与模板进行匹配，通过分词在模板内的填充得到最终查询语句。

这种方法可以有效应对某种类型的问题，但是其缺点也很明显：越是成熟的问答系统就需要越多的模板做支撑，比如将图中的问句更改为“哪个国家制造的 xxx 智能水下机器人”，就需要再做一个模板来对应其中的文本信息。著名的 True Knowledge 网站就是基于 1200 多个模板的方式运行的，人工成本的支出非常大，同时又因为一个问题可以使用多个不同的模板进行回答，所以还要在模板数量过多时维护一个基于问题的模板排序功能，否则可能会发生反馈冲突的问题。

因此虽然基于模板的问答系统查询响应速度快，对问句的回答准确率高，但需要极大数量的模板匹配用户对于同一问题的不同问法，需要投入大量的时间和精力，成本过高，且模板过多也会降低系统的查询效率。

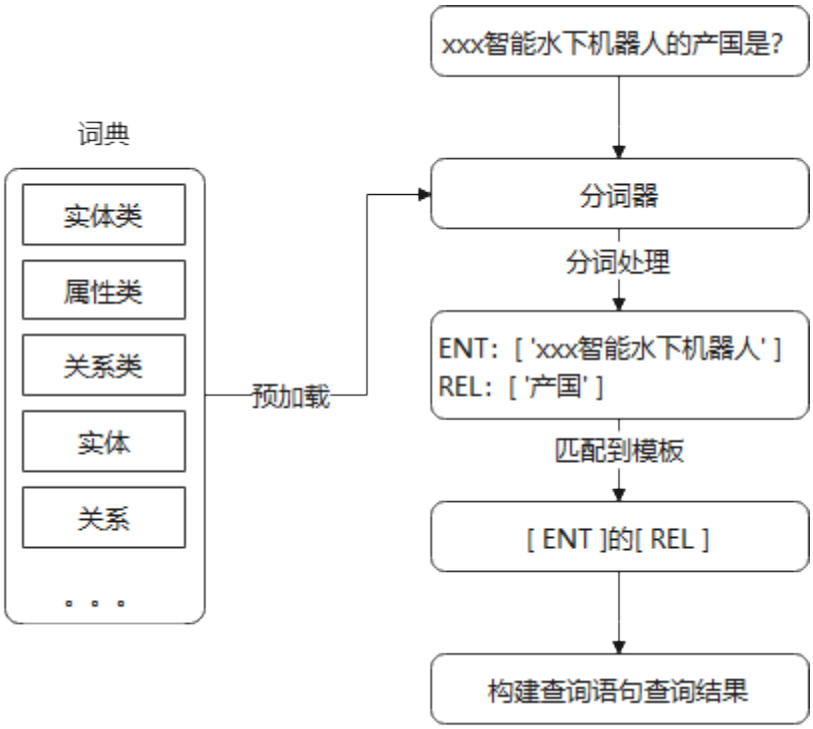

图 4.1　基于模板解析示例图

Fig. 4.1　Template-based parsing of sample diagrams

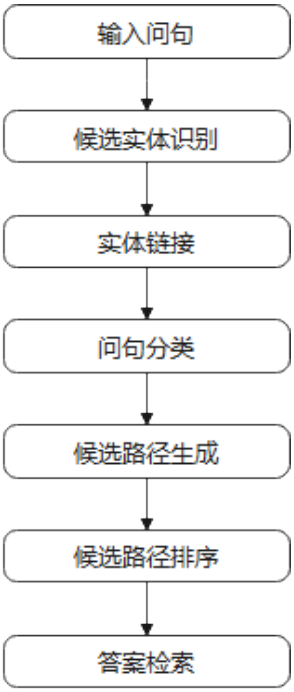

图 4.2　问句解析流程

Fig. 4.2　Question parsing process

随着深度学习在自然语言处理领域的不断发展，基于深度学习的问答任务也随之出现并被不断改进。本文系统基于知识图谱和深度学习，任务主要为接受一个自然语言问句，通过命名实体识别、理解问句中实体间的关系、构建实体关系查询语句，从而获取针对问句的最终答案。例如问句“智能水中目标识别一书的作者是谁？”，根据知识图谱数据库中存储的三元组<智能水中目标识别，作者，曾向阳>，系统通过深度学习的方式分析问句后通过检索给出答案“曾向阳”。

根据第一章的描述，本文系统主要使用基于深度学习的检索式方法，并在候选实体提取及路径排序两个模块针对当前数据集进行工程性应用，本模块的具体流程设计如图 4.2 所示，以下分别对各步骤作详细介绍。

4.1.1 候选实体识别

候选实体即问句中提到的命名实体和属性的集合，系统为全面的提取问句中的相关信息，避免错漏关键信息，采取了模型识别+词典识别+启发式规则三种方式相结合的策略，对于问句中候选实体提及的提取顺序依次为模型识别、词典匹配、启发式规则，具体流程如图 4.3 所示，下文分别对这三种方式做详细介绍。

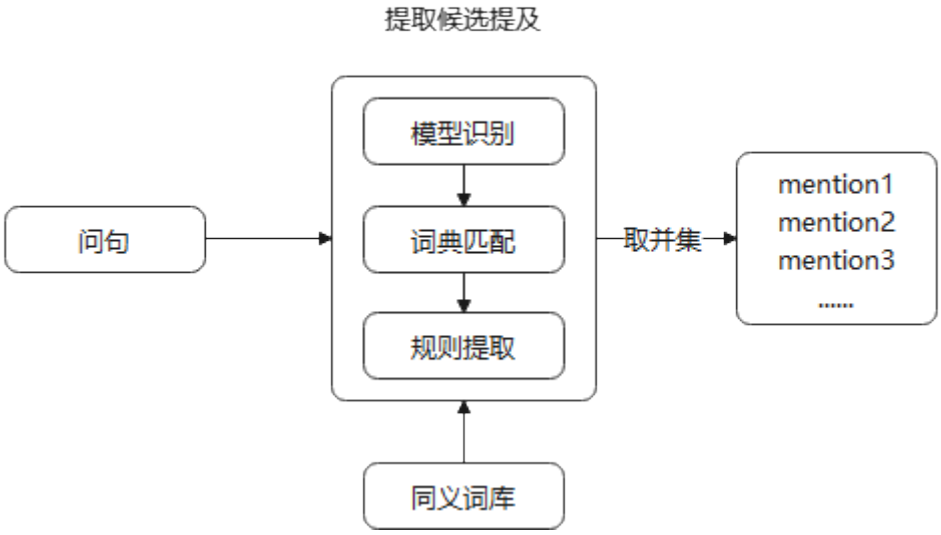

图 4.3 提取候选提及流程

Fig. 4.3 Extract candidate mention process

（1）ALBERT-Bi LSTM-CRF 模型

传统的 BERT 配合 Bi LSTM-CRF 模型[27]在命名实体识别领域虽然有很好的成绩，但 BERT 隐藏层增加其词嵌入层参数也要随之增加的弊端导致其训练性能随着项目数

据量规模的增大而不断降低，根据本文第二章介绍的相关知识，ALBERT 作为 BERT 的优化迭代模型，不但解决了参数量过大导致的一系列问题，还在保证模型效果的前提下降低了训练时间，提升了模型性能。本文选取 ALBERT-Bi LSTM-CRF 作为候选实体识别模型，通过 ALBERT 将问句中蕴含的语义关系特征向量化，使用 BiLSTM 对获取的特征进行提取，通过模型运算得出每个序列对于各标签的权重标注，最终通过 CRF 层添加整体预测的全局条件，进行最后的标签推理。

同时，本文还尝试了省略 BiLSTM 层直接使用 ALBERT+CRF 模型和直接使用 K-BERT[28]模型两种方法在当前工程图谱数据集进行了候选实体提取工作的探索。

因 ALBERT 本身就具有提取句子中蕴含的文本语义和进行词编码的能力，因此去除 BiLSTM 层可以减少大量该层产生的参数，提升训练模型的速度。

K-BERT 提出的初衷契合领域图谱知识问答的思想，它主要针对 BERT 模型在缺乏特定领域知识的前提下，通过领域数据集训练结果欠佳的情况。在阅读领域文本时，专家会通过自己掌握的相关领域知识进行知识推理。为使机器也能利用这一机制，Liu[28]等人提出将专家级领域图谱三元组作为领域知识与句子相结合，方便模型进行知识推理，同时为克服句子引入过多知识带来的知识噪声问题，模型还引入软定位和可见矩阵来限制不同位置的词对不同知识的可见性。

为验证以上方案的可行性以及方案在本文系统领域图谱的性能表现，本文使用人工标注的 BIO 领域数据集进行了相关实验，实验结果如表 4.1 所示。

表 4.1　多个模型实验结果对比

Tab. 4.1　Comparison of experimental results of multiple models

	精确率（%）	召回率（%）	F1 值(%)
K-BERT	84.37	81.65	82.99
ALBERT-CRF	84.81	80.77	82.74
ALBERT-BiLSTM-CRF	86.11	82.92	84.25

通过实验结果分析得出，ALBERT-CRF 虽然可以通过去除 BiLSTM 层削减部分参数量，但模型 F1 值也会有所下降；K-BERT 也会因专家级图谱的质量好坏影响到自身性能，因此为保证对问句中的候选实体提取精度，系统最终采用 ALBERT-Bi LSTM-CRF 模型作为实体提取模型。

（2）词典匹配

系统会维护一个基于知识图谱的实体词典库，用于问句关键词文本匹配，采用最长匹配规则提取候选实体提及。同时为减少文本匹配结果的噪声文本，系统设置了停用词

机制，将比如“什么”“哪个”“谁”“的”“人”“为什么”“是”“有”等词设置为停用词，匹配时会忽略掉这些词的匹配，匹配流程示例如图 4.4 所示。

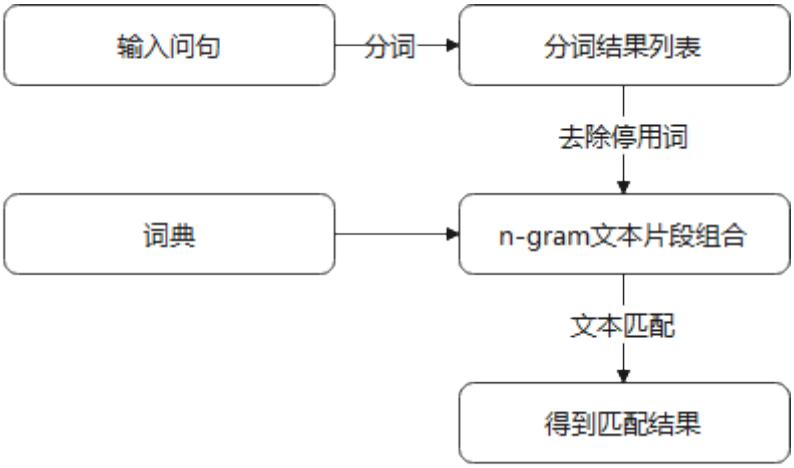

图 4.4　词典匹配流程示例

Fig. 4.4　Dictionary matching process example

（3）启发式规则

针对一些数字、日期、书名文章名等特殊实体和属性值，系统使用正则匹配提取其中的关键词，书名文章名等通过判别“《》”符号提取其中的文字，数值单位提取规则如下：

unit_regex = '([0-9]+.?[0-9]+)(%s)+' % '|'.join(unit_wds)

其中 unit_wds 为提前设置的单位值索引字典。

日期提取规则如下：

time_regex = '[0-9]{4}年[0-9]{0,4}月?[0-9]{0,4}日?'

4.1.2　实体链接

实体链接即将上一步提取的实体提及关联到系统知识图谱数据库的任务，本阶段任务主要解决候选提及的同义词问题。同义词基于系统维护的同义词库进行扩展，例如将“CNN”一词映射到图谱中存储的“卷积神经网络”实体。本文系统的实体链接分为以下三个步骤：

（1）候选实体扩展。将系统维护的同义词库通过倒排索引的方式与得到的候选实体集合相匹配，得到扩展后的候选实体集合。

（2）候选实体召回。针对上一步骤得到的候选实体集合中的每个实体，构建查询语句链接数据库，将能链接到的所有实体作为候选实体集合。

4.1.3 问句分类和候选路径生成

首先通过实体链接的候选实体集合中包含实体的数量判断问句是单/双/三实体类型问句，再通过提前训练的基于 BERT 的分类模型判断是一跳还是多跳类型的问句，最后基于以上得到的信息，以识别到的实体为中心向周围扩散取值。

通过对系统领域图谱数据的观察，将常见问句分为以下六个种类：

（1）一跳单实体查询

作为头实体：<实体><关系><?x>

作为尾实体：<?x><关系><实体>

（2）二跳单实体查询

作为头实体：<实体><关系 1><?x><关系 2><?x>

作为尾实体：<?x><关系 1><实体><关系 2><?x>

（3）一跳双实体查询

共同节点：<实体 1><关系 1><?x><关系 2><实体 2>

顺序节点：<实体 1><关系 1><实体 2><关系 2><?x>

（4）二跳双实体查询

共同节点：(<实体 1><关系 1><?x><关系 2><实体 2>）→<实体 x><关系><?y>

顺序节点：<实体 1><关系 1><实体 2><关系 2><?x><?关系><?y>

（5）一跳三实体查询

对这一种类的问句查询策略为对一跳双实体查询的复合，先将三实体两两组合，取得它们之间的所有共同节点和顺序节点路径后，再取交集作为候选路径。

（6）二跳三实体查询

对这一种类的问句查询策略为对二跳双实体查询的复合，先将三实体两两组合，取得他们之间的所有共同节点和顺序节点路径后，再取交集作为候选路径。

4.1.4 候选路径排序

加入本模块的目的是在众多候选路径中选出与问句匹配度更高的路径集合，提高对问句回答的准确率，通过本步骤对之前获取的候选路径进行排序，确定最终的答案查询路径，具体排序流程如图 4.5 所示。

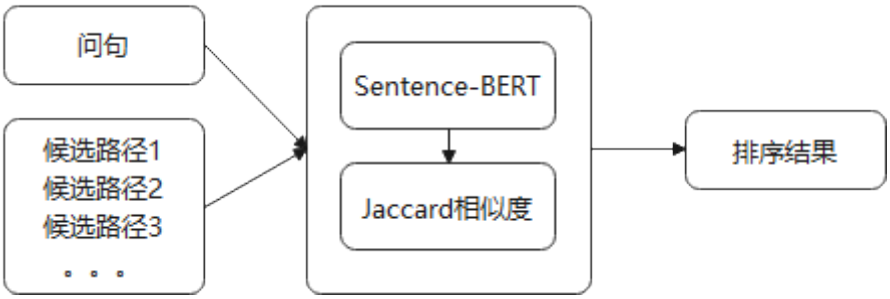

图 4.5　候选路径排序

Fig. 4.5　Candidate path sorting

本文系统原定使用 BERT 对候选路径进行语义排序，但调研后发现 BERT 模型因为自身的模型结构导致不能应用于对实时性交互较强且数据处理较大的系统中。BERT 模型自提出后，以非常强的姿态横扫 NLP 领域多项任务，在语义相似度识别（semantic textual similarity）任务的表现同样如此，但是由于 BERT 模型的结构，每次判断两个句子的语义相似度时必须将两个句子拼接，使二者同时进入模型才能进行运算，这造成了极大地计算性能开销。举例来说，如果目前我们有一个找出 10000 个句子中语义最相似的句子对任务，通过 BERT 模型进行计算，需要进行（10000*9999）/2 次运算，即使是在一块现代 V00 型号的 GPU 上进行运算，也需要较长的计算时间才能得出最终结果，因此 BERT 的结构不适合这种有实时性要求类型的语义相似度匹配任务。

在一些基于模板的检索式实际问答任务中，通常会人为地提前设置许多对于当前系统常用且描述清晰的问句和问句对应的答案，这些提前设置的问题一般被称为“标准问”。在实际使用时，将用户输入的问句与这些提前设置的标准问题进行语义相似度匹配，将匹配到的与输入问句最相似的标准问对应的答案作为最终结果返回给用户查看，如果在这种场景下应用 BERT 模型，就要把问句与每个标准问拼接后输入模型进行相似度计算，这样做耗时耗性能，是不能在一个实时交互的系统落地的。

Sentence-BERT[29]针对以上提出的 BERT 模型的不足之处进行了改进，通过借鉴双塔网络模型的结构，提前准备两个共享参数的模型，将两个句子分别输入两个模型中，通过模型计算获取每个句子的表征向量，用于计算两个句子的语义相似度，其结构如图 4.6 所示。

对于图中两个参数共享的模型可以理解为同一个模型用在两个地方。这种做法的好处在于对于不同的问句，同样的候选答案有相同的计算表征向量，因此可以提前缓存各候选答案的计算后表征向量，这样以后每次使用仅需计算输入问句的表征向量后再与缓存的候选答案表征 u 和 v 使用余弦相似度做相似度计算即可，这个相似度计算的计算量很小，相比仅使用 BERT 模型进行计算，借助双塔模型结构能极大地提高匹配效率。

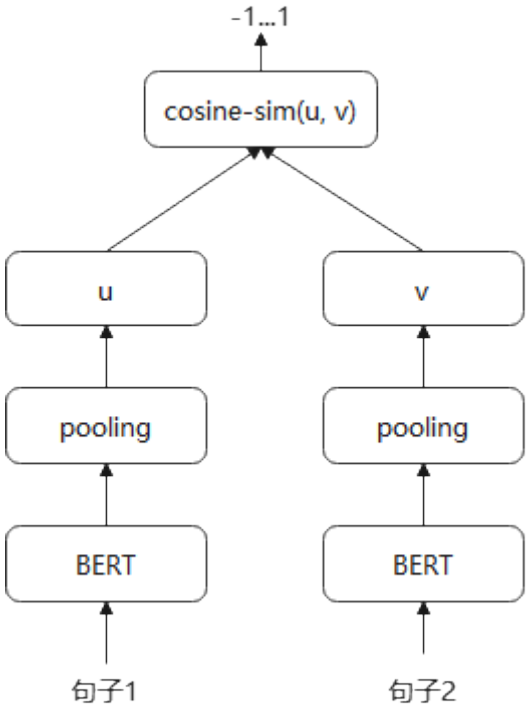

图 4.6　双塔模型结构图

Fig. 4.6　Structure diagram of double tower model

因此本文设计的系统采用 SBERT 对候选路径集合进行筛选，计算每条候选路径与问句的语义相似度，一方面确定当前候选实体的正确性，另一方面判断候选路径的准确性，最终筛选出得分高于预设阈值的候选路径集合。该方法的关键是以图谱中存储的各实体为中心，提取其关联关系，将二者拼接为字符串后计算其向量表示并提前存储，这样做能极大节省路径排序过程的计算量，提升问句解析的整体性能。

比如对于问句“哈利波特一书的作者是谁？”，候选实体提及将“哈利波特”作为问题的中心实体，通过实体链接取得与中心实体相关的子图，由子图构建得到候选路径“哈利波特作者 J.K.罗琳”、“哈利波特导演克里斯·哥伦布”、“哈利波特主演丹尼尔”等，其中含有作者这一关系的候选路径与问句语义相似度最高，从而得出候选路径的排序序列。具体排序过程实现代码如下：

```
class PathSort(object):
    def __init__(self):
        #加载模型
        self.model = SentenceTransformer(Config.Sentence_BERT_model)
        #加载提前存储的路径向量化预运算结果
```

```
        self.path2tensor = {}
        if os.path.getsize(Config.path2tensor) > 0:
            with open(Config.path2tensor, "rb") as fIn:
                stored_data = pickle.load(fIn)
                self.path2tensor = stored_data['sentences2emb']
    #路径排序
    def predict(self, q_text, sim_texts, sim_paths):
        q_list = [q_text]
        embeddings1 = self.model.encode(q_list) #将问句向量化
        cos_path = []   #初始化存储候选路径向量化的容器
        for i in range(len(sim_texts)):
            if sim_texts[i] in self.path2tensor: #向量在预运算集合中存在
                path_tensor = np.asarray(self.path2tensor[sim_texts[i]])
            else:
                path_tensor = self.model.encode(sim_texts[i])
                self.path2tensor[sim_texts[i]] = path_tensor
        #计算候选路径与问句的相似度
        cosine_score = util.cos_sim(embeddings1, path_tensor)[0]
        cos_path.append((sim_paths[i], sim_texts[i], cosine_score))
        cos_path.sort(key=lambda x: x[2], reverse=True) #根据得分由高到低排序
        return cos_path[:10]   #返回排序得分前 10 名
```

为进一步提高问句答案检索结果的准确度，针对语义相似度筛选后得出的排序集合，系统引入 Jaccard 相似度对候选答案做最终排序，选取与问句字符重复度最高的候选路径为最终答案检索路径。最后系统根据最终路径构建检索语句，返回答案。

4.1.5 结果分析

本章节设计的方法验证需要有当前领域的 BIO 语料标注数据集、同义词库和标准问句及对应答案的数据集，由于目前没有人工智能领域的中文标准问答语料，因此本文人工设计了 300 个针对当前领域数据集的相关问答对测试集对问答模块进行测试，以验证本文设计的问句解析模块的正确性，问句的类型已经由 4.1.3 节列出，且各类型的问答对数量相同。

对于用户的提问，其命名实体识别正确，根据路径排序后查询返回的结果准确就可以认为模块返回了正确答案。从实验结果可以看出，问答模块可以正确回答 78%的本文人工构建问句。尽管一部分问句采用了不同的方式进行提问，但通过 Sentence-BERT 进

行语义排序仍能获取相同的准确答案，基于模板的解析方式虽然也能对相同语义的不同问法做出正确解析，但是需要用穷举的方式列出相同语义的所有问句模板，不是解决此类办法的最优解，本模块设计之初搭建的基于模板解析的方式因未能全部列出当前领域的问句模板，正确率只有 61%，远低于当前模块的解析正确率。由于本文设计的模块在路径排序阶段需要计算大量子图路径与问句的相似度，在相同的环境中与基于模板的方式相比平均响应时间多了 0.7 秒，但依然能满足在较短的时间内做出响应的基本要求。另外因模块设计时主要考虑人们平时的口语化提问习惯，当前对于包含三实体以上及多于两跳的问句类型支持性较差，对于这些类型问句的支持也是当前模块需要改进的方向。

现有实验证明本文设计的问句解析模块能较好地应用于人工智能领域的知识图谱，同时当前模块也可以分为更细粒度的结构，后续可以针对每个子模块进行进一步设计，以提升模块整体性能。

4.2 系统体系架构分析

系统采用基于 C/S 模式改进的 B/S 架构，即浏览器/服务器架构。该架构模式只有少量逻辑在客户端实现，主要业务逻辑都被放在后端，系统由浏览器端、服务器端及数据库端构成三层架构，B/S 架构具体如图 4.7 所示，这种架构模式无须像 C/S 架构一样安装客户端程序，只要有浏览器即可使用系统，并且它可以被部署在广义网，通过使用不同的控制策略使得不同角色的用户都能使用，交互性较好，最主要的是它不像 C/S 模式升级时要更新多个客户端，B/S 架构仅需升级服务端即可。

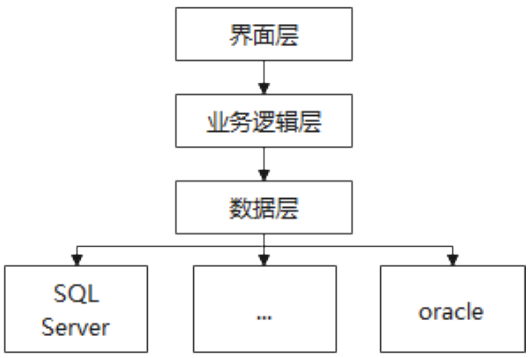

图 4.7 B/S 架构图

Fig. 4.7 B/S architecture diagram

基于知识图谱的领域性知识问答系统的构建框架如图 4.8 所示，因本文设计的系统采用的 B/S 系统架构，因此主体部分同样由界面层、业务逻辑层和数据层构成，下面将对各层给出详细介绍。

界面层包含用户问答页面、结果可视化页面、知识图谱管理页面、同义词管理页面及用户管理页面。用户问答页面主要和用户交互，用于接收用户的问题；结果可视化页面主要展示查询结果；知识图谱管理页面主要用于图谱的数据更新和维护；同义词维护页面主要用于对同义词的管理；用户管理页面主要用于系统管理员管理系统用户。结果可视化界面除了基本的文本结果展现，还提供了图谱可视化效果，进一步提升用户体验。

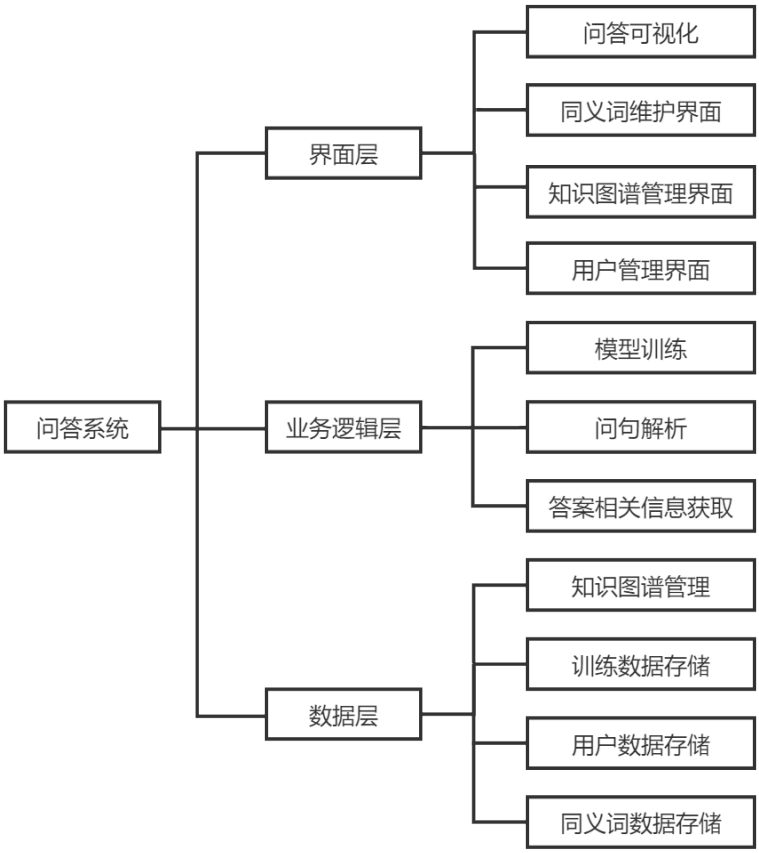

图 4.8　问答系统架构图

Fig. 4.8　Q & a system architecture

业务逻辑层是问答系统的关键部分，不仅是用户和系统交互的桥梁，也是系统构建的核心。系统接收到用户输入的问句后，经过规则拦截、问句类型识别、命名实体识别、实体链接和若干排序剪枝操作后，构建查询语句返回查询结果，以上每个模块都相对独立，通过模块串接完成任务。此外，业务逻辑层还包含用户管理相关功能，比如用户权限的修改等。

数据层是系统的数据核心部分，其中最为关键的部分是领域知识图谱的更新和维护，除此之外数据层还包含系统运行过程产生的数据，如蕴含知识的文本、用户数据和错误日志等，文本信息使用 Elasticsearch 存储，同义词存储、用户数据和错误日志等使用 MySQL 进行存储。

图 4.9 为系统的开发框架，主要展示了系统构建主要用到的技术。

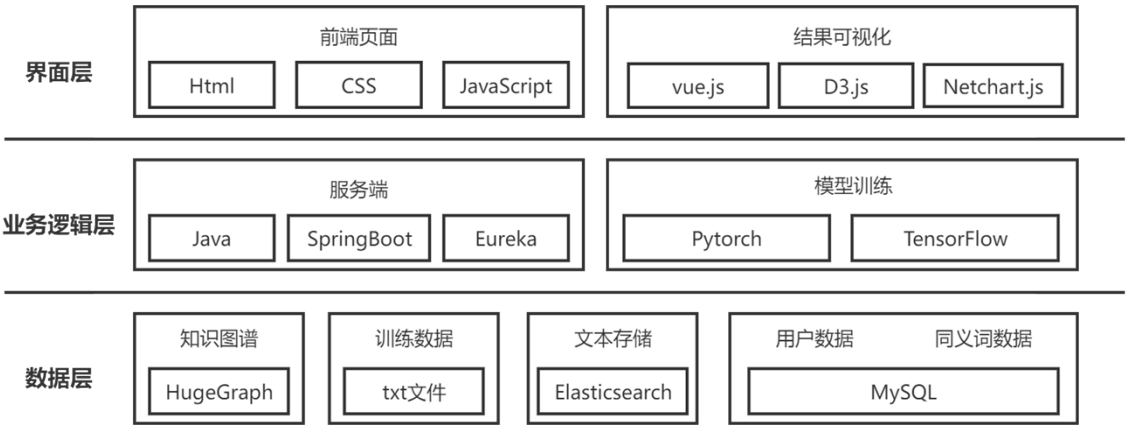

图 4. 9　系统开发框架图

Fig. 4.9　System development framework

首先，Html、CSS 和 JavaScript 是前端构建的三驾马车，本系统的界面层构建也是基于这三项主要技术，同时为了提高开发效率，提供更好的项目后期维护更迭环境，系统使用 vue 作为主要开发框架，结合 Netchart.js 和 D3.js 做项目的可视化部分，提高开发质量的同时提供更优质的用户体验。

业务逻辑层使用 SpringBoot 作为项目主要开发项，它不仅能独立运行 spring 项目，而且内嵌 servlet 容器，提供 starter 且自动装配 Spring，极大的方便项目的配置和开发过程，服务使用基于 Netflix 开发的 Eureka 服务框架，基于此服务平台开发的系统如果在运行过程中宕机，系统会自动切换到可用的 Eureka 服务，待到宕机服务恢复后，服务平台会自动切换恢复并重新注册服务，这样能极大的保证系统的稳定性。

数据层使用 HugeGraph 作为存储容器，该数据库由百度团队开源，支持百亿以上的边和结点的快速导入，并提供极快的查询响应速度，能保证系统查询的响应速度。模型训练使用的带标签文本数据使用本地 txt 文件存储，同时将模型处理过后的与实体或关系相关联的文本存储到 Elasticsearch 中，供用户检索查询使用。最后，系统管理模块所产生的的用户关联数据和同义词数据，通过 MySQL 数据库存储。

4.3 功能模块详细设计

根据第三章的需求分析，同时结合上一节对问答系统核心模块的设计，从实际系统的落地应用角度，本文对系统进行详细设计，系统共分为问句答案检索、图表构建、结果可视化、图谱数据管理、同义词维护、用户管理几大模块，具体功能结构如图 4.10 所示

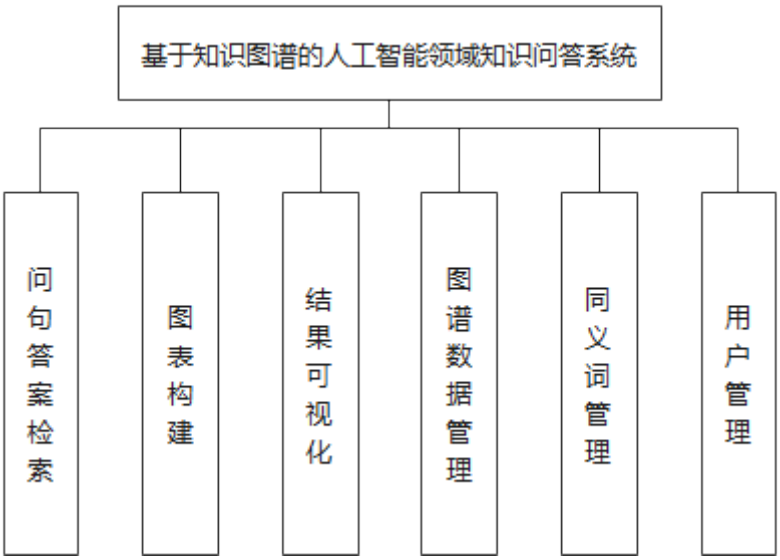

图 4.10 系统功能结构图

Fig. 4.10 Function structure diagram of system

问句答案检索是本文系统的核心，主要用于对用户输入问句的语义解析和答案检索，图表构建针对问句检索对于范围性问句检索结果的不足之处，可以更好地展示检索结果，结果可视化除了提供直接的问句答案，还将答案相关的图谱结构、答案知识来源及与问句相关的文本通过合理的页面布局展现给用户；另外，同义词库能提高候选实体识别的准确率，图谱数据管理模块保证图谱知识库的知识更新，用户管理机制能够保证入库数据的可靠性。下面会分别对这些模块进行详细的设计阐述。

4.3.1 问句答案检索

问句检索的详细方案已经在本章第一小节进行了详细的阐述，该功能模块主要完成对问句的语义解析和答案检索。

首先识别问句中的候选实体提及，简单的字符匹配并不能很好的提取全部有用信息，比如“上海”也称作“沪”，“大连理工大学”也被称作“大工”，同时还有错输漏输的问题，如果图谱中仅存储了某一系列词的代表词，使用基于图谱的词典进行字符匹配

就不能很好地取出问句中的信息，所以系统使用了语义模型配合增强当前模块在词语层面的语言泛化能力，增强系统对问句语义的挖掘，能更为准确的提取出关键信息；相比使用 BERT 模型必须将问句与候选路径同时输入才能计算语义相似度的结构，使用基于双塔结构的 SBERT 模型进行候选路径排序也很好的节省了系统的计算时间，提升了系统的响应速度。

4.3.2 图表构建

问句答案检索模块已经实现了问答系统的基本功能，通过文本形式对问句给出了直接的答案，但是对于答案是一系列集合的情况，仅通过文字展现不能突出其中蕴含的重点信息，为方便用户获取更多相关信息，我们选用图表形式对问题答案进行拓展展示，图表可以展示数据的分布和聚合情况，适合展示集合式数据集，具体流程如图 4.11 所示。

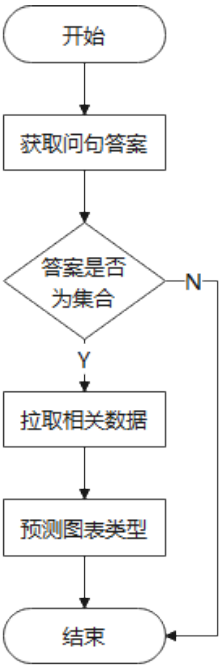

图 4.11 图表构建流程图

Fig. 4.11 Flowchart of Diagram Construction

为能较为准确的展示检索到数据的展现形式，模块进一步结合使用了 data2vis 模型[30]作为转换工具进行表格形式预测。该模型将可视化问题映射为一个序列到序列的问题，其输入序列是一个 json 格式的数据集，输出序列是一个 Vega-lite 格式的有效可视化格式数据集。因为 json 格式的数据严格意义上来讲并不是一个序列性格式，为了能将 seq2seq 模型更好地运用到非序列的模型中，data2vis 使用了双向 RNN 和注意力机制进

行处理。模型的基础架构是一个添加了注意力机制的编码器-解码器结构，将双向 RNN 作为编码器，解码通过计算编码器的输出结果输出目标序列的概率。

4.3.3 结果可视化

结果可视化共分为答案展示、答案来源展示、相关文本展示、图谱展示及图表展示五个模块，具体结构图如 4.12 所示。

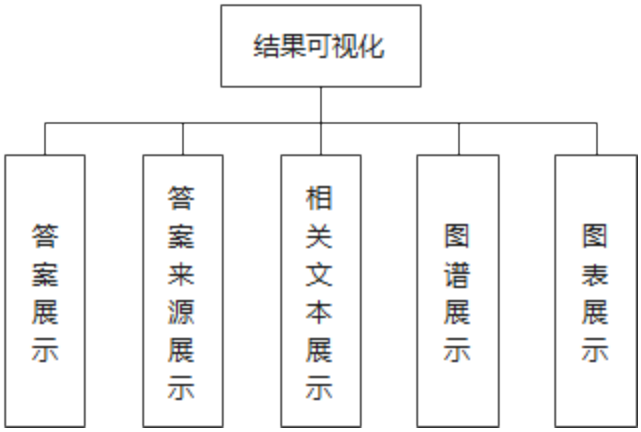

图 4.12　结果可视化功能结构图

Fig. 4.12　Result Visualization Functional Structure Diagram

（1）答案展示

本文设计的系统与传统检索系统给出一系列与查询问句相关的链接不同，直接给出问题的答案是本系统设计的主要特点，对于展示形式，本文共设计了三种直观的检索结果展示方案：对实体查询的结果展示、对尾实体/头实体查询的结果展示、对集合查询的结果展示。

对于单个词语的查询，系统使用实体查询结果展示方案，包含当前实体的信息展示、一跳关系展示，其中每个一跳关系展示区域点击后展现当前关系对应的三元组、三元组的学习来源及针对当前三元组的进一步检索推荐方案，点击每个一跳关系后的跳转按钮可对当前三元组尾实体进行进一步查询。

对于包含对尾实体/头实体或查询的问句，系统对问句解析查询后会直接给出问句对应的答案，只是给一个答案虽然能达到回答问题的目的，但用户并不能对于答案对应的其他隐藏信息进一步挖掘，所以系统还设计了针对答案列出当前检索的图谱路径、答案句子级来源及针对当前检索路径的进一步检索推荐。

对于检索答案为集合的问句，系统会将检索得到的集合子元素依次列出，同样的，仅仅列出一些文本虽然直观的给出了答案，但是对于集合中每个子元素的其他隐含的与

答案关联的关系也是分析问题的一部分潜在价值的信息，所以系统还设计了针对每个子元素的信息展开功能，点击每个子元素会展示当前子元素的学习来源和针对当前子元素的其他关联信息的推荐检索。

（2）答案来源展示

仅提供答案及答案相关联的三元组不能充分展示当前问句相关的信息，因此系统设计了提供与答案三元组相关连的学习来源，在信息抽取时，除了提取句子中的实体、关系与属性，同时还会将提取数据相关的句子、段落、文章，将这些信息一起存入数据库，待到这些信息被检索后，通过结构化查询将信息一起返回页面进行展示，用户仅需点击右侧的知识来源标题或答案下方的来源句子，即可查看整篇文章，获取更多与检索目标相关的信息，系统将包含答案的句子在文章中以红色标记，方便用户定位知识来源在文章中的具体位置。另外，为了缓解系统压力，系统使用懒加载机制对整篇文章进行加载，用户每次查看文章时，系统通过监听用户鼠标滑动来判断是否需要加载更多文章接下来的部分。

（3）相关文本展示

除直接相关文本外，与问句中的一些词语片段相关的其它文本也很可能含有和检索结果相关的信息，因此本文设计系统系统使用 Elasticsearch 结合 IK 分词器对知识文本进行存储，方便问句与相关文本的匹配。IK 分词器分为 ik_max_word 和 ik_smart 两种模式。ik_max_word 会将文本做最细粒度的拆分，例如会将“大连理工大学软件学院”拆分为“大连理工大学”、“大连理工”、“大连”、“理工”、“连理”、“工大”、“大学”、“软件学院”、“软件”、“学院”；ik_smart 会将文本做最粗粒度的拆分，例如会将“大连理工大学软件学院”拆分为“大连理工大学”、“软件学院”。系统为更细致的匹配问句中的相关文本信息，采用 ik_max_word 模式对文本进行分词存储，使用时直接将问句输入 Elasticsearch 进行分词后匹配文本信息，将查询后的结果集合返回展示页面。

（4）图谱展示

对于答案的展示方式除了使用文本直接展示外，本文还设计了使用知识图谱结构展示的方式更加直观的展示答案及答案相关的其他节点，系统基于 NetChart.js 构建前端可视化界面，提供以当前答案节点为中心的 1-6 跳扩展查询，对于用户想要查询的节点，直接点击即可进行跳转查询其具体信息。

（5）图表展示

系统通过问句解析模块获取答案后，再拉取与答案相关的一跳关联数据，为了能以最适合的形式展现获取的数据，通过 data2vis 模型生成图表数据和预测图表类型。系统

目前支持生成 area、bar、circle、line、point、tick 共 6 种类型的图表，后端通过以上流程返回数据后，前端使用基于 ant/g2 框架将数据在页面上进行渲染。

4.3.4 图谱数据管理

随着科研进程的发展，人工智能领域的知识也不断被扩充，同时在数据库建立之初系统对于知识的收集也可能存在遗漏的情况，为了不断完善并扩充知识库，本文为系统设计了图谱数据管理模块，其操作流程如图 4.13 所示。

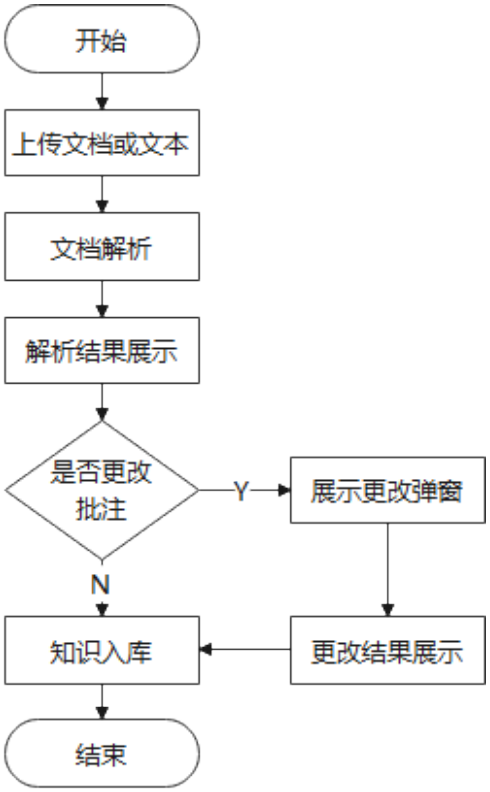

图 4.13 图谱数据管理流程图

Fig. 4.13 Flowchart of Graph Data Management

为方便用户操作，系统采用接收上传文档的方式接收外界传入的文本资料，并以句子为单位自动解析每个句子中蕴含的三元组关系，图谱具体存储结构已经在第三章进行了详细阐述，解析完成后会将结果展示给用户供用户查看或修改。在数据导入前会校验导入的数据与图谱中已存数据的重复性和关联性，校验完后执行数据入库，完成对数据库的知识更新。

4.3.5 同义词维护

同义词是指一个词可能有多种称谓，比如“大连理工大学”被称作“大工”、“理工大学”，“上海”被称作“沪”等，本文设计了同义词模块协助问句解析的候选实体提及模块更好地抓取问句中的同义词信息。该模块共分为实体类、同义词组、代表词三项，通过标记代表词的实体类别可有效区分同词不同义问题，如电子产品类别的“苹果”和水果类别的“苹果”，同义词组是一个词不同称谓的集合，代表词是这一系列集合的代表，也是这一系列词组在图谱上节点的名称代表。同义词维护的流程如图 4.14 所示。

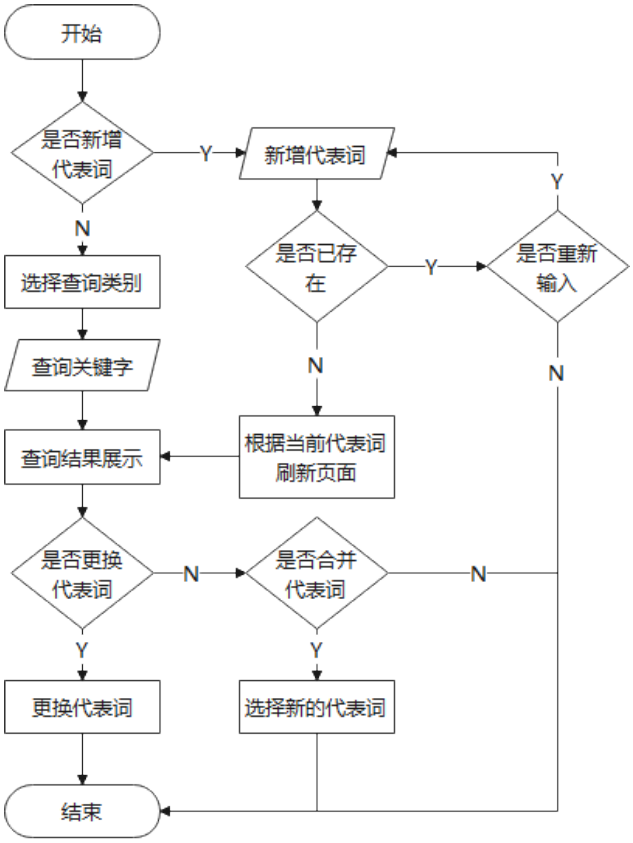

图 4.14 同义词维护流程图

Fig. 4.14 Flowchart of Synonym Maintenance

同义词维护分为新增代表词、更换代表词、合并代表词、同义词信息查询四项主要功能。新增代表词即在选择的实体类下添加一个新的实体代表词汇；更换代表词即在当前选中的同义词组中选择一个词汇代替当前同义词组的代表词；合并代表词为多组同义词组的合并，在合并是要选出合并后同义词组的代表词；同义词信息查询分为根据实体类查询和模糊查询两种模式，若选择实体类查询，系统会将与输入的实体类相匹配的所

有关联同义词组返回，若选择模糊查询，系统会把所有与输入文本模糊相关的同义词组信息全部返回。

4.3.6 用户管理

为保证系统后续录入数据的正确性，本文设计了角色管理模块，管理员或拥有其指定权限的用户角色才可以对知识库进行数据更新操作。用户管理模块共分为用户注册、用户登录、用户权限管理三项功能，本文设计的系统无需登录也可使用，但如果要参与到知识库的更新与维护工作中，就需要一个有权限的身份进行操作。

4.4 数据库设计

4.4.1 图数据库设计

本文系统的图数据库设计使用 HugeGraph，其存储结构由四种基本元数据构成：数据索引 IndexLabel、节点 VertexLabel、边 EdgeLabel 和属性 PropertyKey。数据索引主要为了方便查询，存储名称、类型、在哪些属性建立索引等关于索引的约束信息。节点主要存储名称、值等描述节点的具体信息。边主要存储边的类型、具体描述等关键信息。属性主要描述边和节点本身的属性信息。

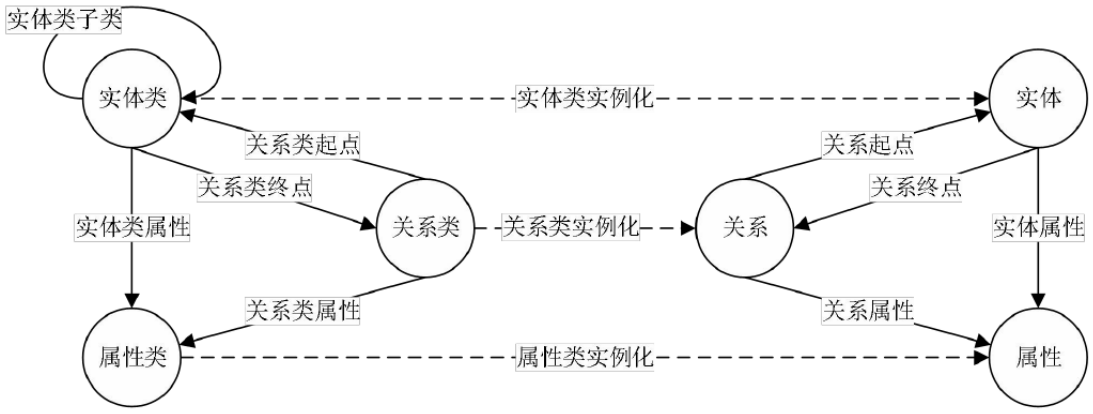

图 4.15 HugeGraph 存储设计

Fig. 4.15 Storage design of HugeGraph

图 4.15 为本文设计的图数据库存储结构，定义了存储结构中实体类、属性类、关系类、实体、属性、关系之间的连接关系。系统会根据设计的结构创建数据对应的本体，再由本体实例化出相应的业务数据。

4.4.2 关系型数据库设计

系统主要使用 MySQL 数据数对同义词维护模块及用户管理模块进行数据存储，因用户管理模块仅涉及到用户权限的存储，所以本节只对同义词维护模块作具体介绍。同义词维护模块的数据库结构逻辑图如图 4.16 所示。

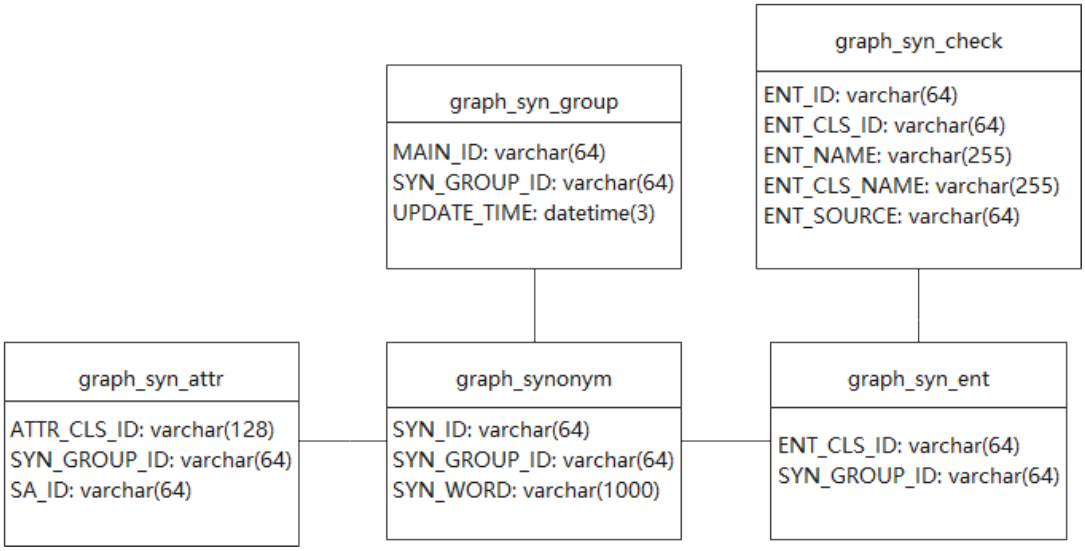

图 4.16　同义词数据库逻辑结构图

Fig. 4.16　Synonym Database Logical Structure Diagram

表 4.2 为同义词存储表，ID 为每个同义词的主键，通过 ID 可以快速定位到每个同义词，同时与同义词组 ID 的关联实现了同义词的归属群组。

表 4.2　同义词存储表

Tab. 4.2　Synonym storage table

序号	字段	数据类型	非空	描述
1	SYN_ID	varchar(64)	是	同义词 ID，主键
2	SYN_GROUP_ID	varchar(64)	是	同义词组 ID
3	SYN_WORD	varchar(1000)	是	同义词

表 4.3 为同义词组存储表，主键为同义词组 ID，该表实现了同义词组与代表词的关联，代表词为同义词集合的子集，因此 MAIN_ID 为上方表 4.2 中 SYN_ID 的子集，这里将代表词单独提出来赋予 ID 方便系统的数据查询。

表 4.3　同义词组存储表

Tab. 4.3　Synonym group storage table

序号	字段	数据类型	非空	描述
1	SYN_GROUP_ID	varchar(64)	是	同义词组 ID，主键
2	MAIN_ID	varchar(64)		代表词 ID
3	UPDATE_TIME	datetime(3)		修改时间

表 4.4、表 4.5 及表 4.6 建立了同义词与图谱的关联关系。

表 4.4 为同义词实体类关系表，表明当前同义词组的所属实体类类别。同义词组和实体类都是集合的代表，但都代表同一类，同义词组为实体类的子集。

表 4.4　同义词实体类关系表

Tab. 4.4　Association table between synonyms and entity classes

序号	字段	数据类型	非空	描述
1	SYN_GROUP_ID	varchar(64)	是	同义词组 ID，主键
2	ENT_CLS_ID	varchar(64)	是	实体类 ID

表 4.5 为同义词属性类关系表，表示同义词组与属性类的关联关系。其中 SA_ID 仅做主键标识，无其他含义。

表 4.5　同义词属性类关系表

Tab. 4.5　Association table between synonyms and Attribute classes

序号	字段	数据类型	非空	描述
1	SA_ID	varchar(64)	是	主键
2	SYN_GROUP_ID	varchar(64)	是	同义词组 ID
3	ATTR_CLS_ID	varchar(128)	是	属性类 ID

表 4.6 为同义词校对表，表示实体的具体信息，主要包括其所属实体类 ID、所属实体类名称、实体名称及录入来源。

表 4.6　同义词校对表

Tab. 4.6　Synonym alignment table

序号	字段	数据类型	非空	描述
1	ENT_ID	varchar(64)	是	实体 ID
2	ENT_CLS_ID	varchar(64)	是	实体类 ID

3	ENT_NAME	varchar(255)	实体名称
4	ENT_CLS_NAME	varchar(255)	实体类名称
5	ENT_SOURCE	varchar(64)	实体录入来源

4.4.3 Elasticsearch 设计

为保证本文设计的问答系统在提供与答案相关的学习来源时有较快的数据查询速度，系统将与知识图谱中节点相关联的文本使用 Elasticsearch 做了分词存储，存储分为文章级、段落级、句子级三种粒度。Elasticsearch 的配置要求使用 json 格式的数据，这里为了便于展示，将配置的关键部分以表格方式列出。

表 4.7 为 Elasticsearch 文档存储配置，主要针对文章级粒度的信息存储，id 为录入文件的唯一代表符号，以后的工作可通过 id 查找对应的文件信息，除此之外还记录了文件的标题、后缀、文件所属组及文件的更新内容等信息。

表 4.7 Elasticsearch 文档存储配置

Tab. 4.7 Elasticsearch document storage configuration

序号	字段	数据类型	描述
1	id	String	主键
2	docTitle	String	文件标题
3	docType	String	文件后缀
4	docOldName	String	原始文件名称
5	docFileName	String	新文件名称
6	createDate	Date	创建时间
7	updateDate	Date	更新时间
8	remark	String	介绍
9	docGroup	String	文件组
10	docText	String	文件文本
11	createUser	String	文档创建用户
12	isDelete	Integer	是否删除

表 4.8 为 Elasticsearch 段落存储配置，主要针对段落级粒度的信息存储。paraText 字段采用 ik_max_word 模式进行分词存储，同时根据当前段落的来源记录到相关的文件组，方便系统实现针对文章的段落级懒加载。

表 4.8 Elasticsearch 段落存储配置

Tab. 4.8　Elasticsearch paragraph storage configuration

序号	字段	数据类型	描述
1	id	String	主键
2	documented	String	文章 ID
3	paraTitle	String	段落标题
4	paraIndex	Integer	段落序号
5	paraText	String	段落文本
6	createDate	Date	创建时间
7	updateDate	Date	更新时间
8	createUser	String	文档创建用户

表 4.9　Elasticsearch 句子存储配置

Tab. 4.9　Elasticsearch sentence storage configuration

序号	字段	数据类型	描述
1	id	String	句子 ID
2	paraId	String	段落 ID
3	documented	String	文章 ID
4	sentenceTitle	String	句子标题
5	sentenceIndex	Integer	句子序号
6	paraIndex	Integer	段落序号
7	sentenceText	String	句子文本
8	createDate	Date	创建时间
9	updateDate	Date	更新时间
10	createUser	String	文档创建用户
11	startDocIndex	Long	句子在文档的起始位置
12	endDocIndex	Long	句子在文档的终止位置

表 4.9 为 Elasticsearch 句子存储配置，主要针对句子级粒度的信息存储。将 sentenceText 字段通过 ik_max_word 模式进行分词存储，在文本匹配功能中能很好地根据问句分词匹配到相关句子，实现相关文本功能的检索。

5　系统实现

本章节为系统实现的展示及具体功能描述。系统使用 vue 实现前端页面、Java 构建后端服务，python 进行数据模型的训练，开发工具基于 vscode、IDEA 和 Pycharm。开发过程中，前端依照 ESlint 规范，基于 npm 进行包管理，后端依照 Java Web 的技术规范，基于 Maven 管理项目依赖。协同开发工具前端使用腾讯 coding 平台，后端使用基于实验室局域网搭建的 GitLab 平台进行代码托管。

图 5.1　系统首页

Fig. 5.1　System Home Page

系统的首页如图 5.1 所示，右上角的同义词管理和知识导入功能只对管理员和管理员指定权限的用户开放，系统的问答功能无需登录也可使用。

5.1　可视化问答交互

5.1.1　问句答案检索

问答功能是本文设计系统的核心模块，以直接回答接收的用户输入问句为主要目标，本文根据答案类型将问答模块的可视化部分细分为实体查询、单实体答案展示、多实体集合答案展示三种展示模块，下面对这三个模块的实现进行详细展示。

（1）实体查询

系统接收一个实体名称，并将该实体的相关信息检索返回并展示，图 5.2 为实体查询的系统实现。

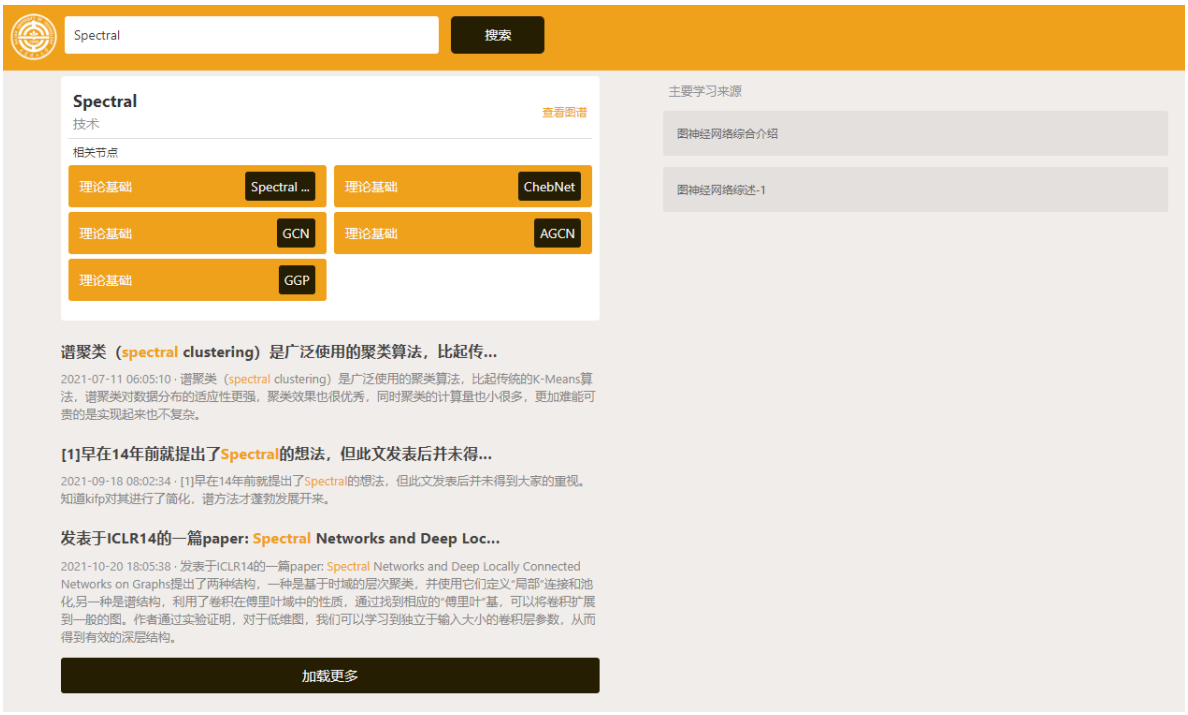

图 5. 2　实体查询

Fig. 5.2　Entity query

页面展示的主要内容包括：当前查询实体的实体类、相关节点、以查询实体为中心的图谱展示、知识主要学习来源和与查询文本相关的其他文本片段。

（2）单实体答案展示

针对答案为单实体的情况，系统设计了单独的 UI 形式做只直观展示，具体展示如图 5.3 所示。

该模块主要包含对问句的直接回答、蕴含答案的关系路径展示、答案学习来源的句子级展示、针对当前关系路径的进一步推荐检索实体及与问句文本相关的其它文本列表。用户可点击下方的进一步搜索查看实体的更多关联关系，答案学习来源及与问句文本相关的文本列表点击后的展示方式与图 5.7 学习来源文章展示相同。

页面设计采用了对答案醒目展示的方式，更直观的突出问句对应的答案，使用户能一目了然的看到问句的直观答案，系统同时将与答案相关的三元组在答案下方标出，使得答案指向更加明确。

图 5.3　单实体答案展示

Fig. 5.3　Single entity answer display

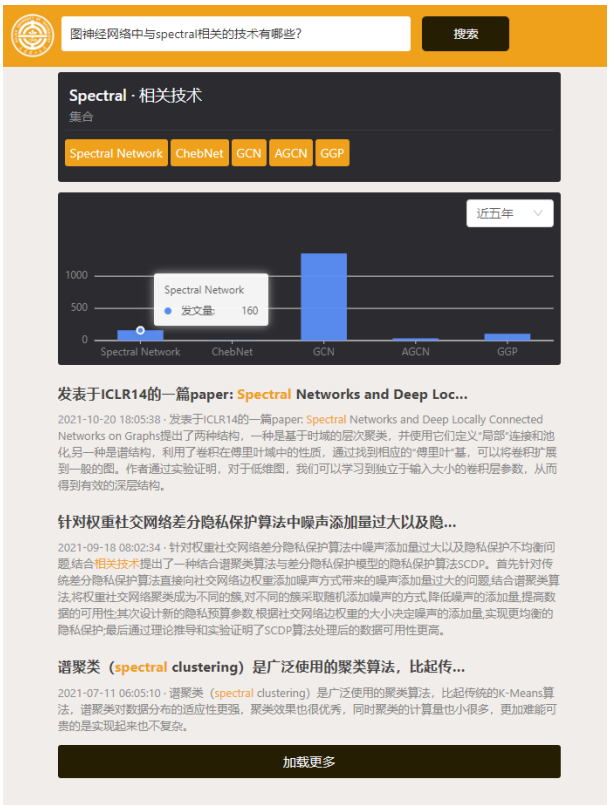

图 5.4　多实体答案展示

Fig. 5.4　Multi entity answer display

（3）多实体集合答案展示

由于多实体数目的不确定性，若使用多个单实体 UI 的方式展示多实体，不仅会造成展示界面的混乱，还会降低用户获取答案的效率，因此本文采用图 5.4 的方式对多实体答案集合作展示。

5.1.2 图表构建

以集合的形式罗列出对问句的回答，虽然基本解决了问题，但是并没有给出更多的有用信息，因此本文设计了图表对比模块辅助用户分析当前集合数据间的关系，方便用户通过对比发现检索目标的发展趋势等。以图 5.4 为例，通过对比我们发现 GCN 近五年的发文量极高，进一步点击 GCN 的柱状图切换为图 5.5 进一步观察选中目标的其他趋势或特征等信息。

图 5.5 图表分析

Fig. 5.5 Chart analysis

5.1.3 结果可视化

本模块是与用户交互的功能汇总，除以上两个小节介绍的功能外，针对当前系统需求，本文还拓展设计了答案关联关系可视化展示的方式，将答案知识条目的出处、相关实体及数据库中包含问句文字信息的相关语句汇总列出。

图 5.6 展示了关联节点的具体展示功能。点击相关节点的文本展示区域，系统会针对当前点击的关联节点展开更多的信息，主要包括查询节点与点击节点的三元组文本展示，当前三元组的学习文本来源，并以句子作展示，根据当前三元组的进一步查询推荐，用户只需点击即可查询，该功能免去了用户打开新页面再输入查询文本的繁琐步骤。

图 5.6 的下半部分展示了与查询文本相关的其它文本片段的检索结果。该模块的检索结果并不是与检索目标绝对一致，本文试图通过检索文本中的片段与数据库中的数据做匹配，返回有文本重合的部分以列表形式供用户查阅，方便用户获取到更多与检索目标相关的其他信息。用户可点击文本标题打开整篇文章进行阅读，其展示形式与图 5.7 相同。系统对此部分同样采用懒加载的形式返回数据，每次只返回检索结果的一部分，用户通过点击加载更多获取更多的文本相关文章。

页面右侧为每个关联节点的学习来源，为方便用户分辨多个实体关联关系的不同学习来源，本文设计采用短线连接的方式指出学习来源与三元组的关系，用户只需将鼠标放到右侧的学习来源文章标题，系统就会自动与当前文章关联的三元组连线。与每个关联节点展开后展示的学习来源不同的是，前者为文章级学习来源展示，后者为句子级来源展示。点击二者都可以打开对应的文章供用户查阅，文章的具体展示页面如图 5.7 所示。

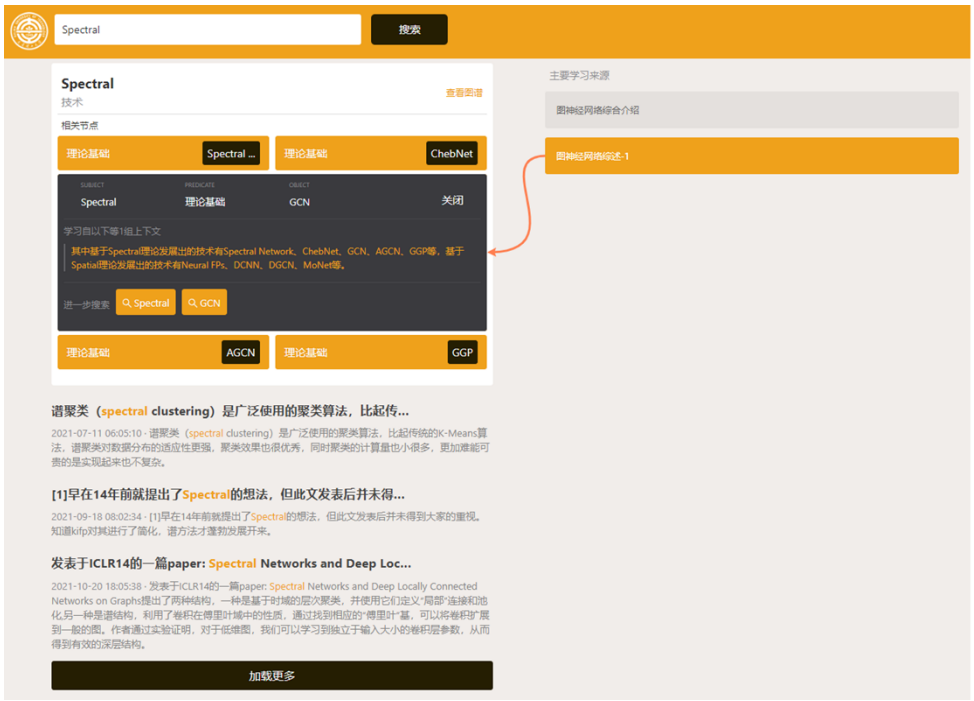

图 5.6　关联节点展开功能

Fig. 5.6　Associated node expansion function

为了使用户能很快定位到文章中与当前查询的知识相关联的部分，系统设计了关键词使用醒目颜色的方式显示，若用户通过点击右侧标题打开文章阅读，则高亮标记与左侧三元组相关的所有文本，若用户通过点击左侧关联节点展开后的学习来源中的句子打开文章阅读，则高亮显示这个句子。

一次性加载全部文章内容不仅不利于用户查看关心的部分，而且还极大的浪费了服务器资源。为了节省系统资源，系统采用懒加载的形式加载文章，每次只加载一个段落，前端通过监听用户鼠标的滑动判断是否需要加载更多文章内容，在判断用户需要阅读更多文章内容时才会调用加载接口，通过服务器请求更多的文章资源，并通过返回数据的句子 id 进行段落拼接，将文章渲染在屏幕上。

图 5.7　学习来源文章展示

Fig. 5.7　Learning source article display

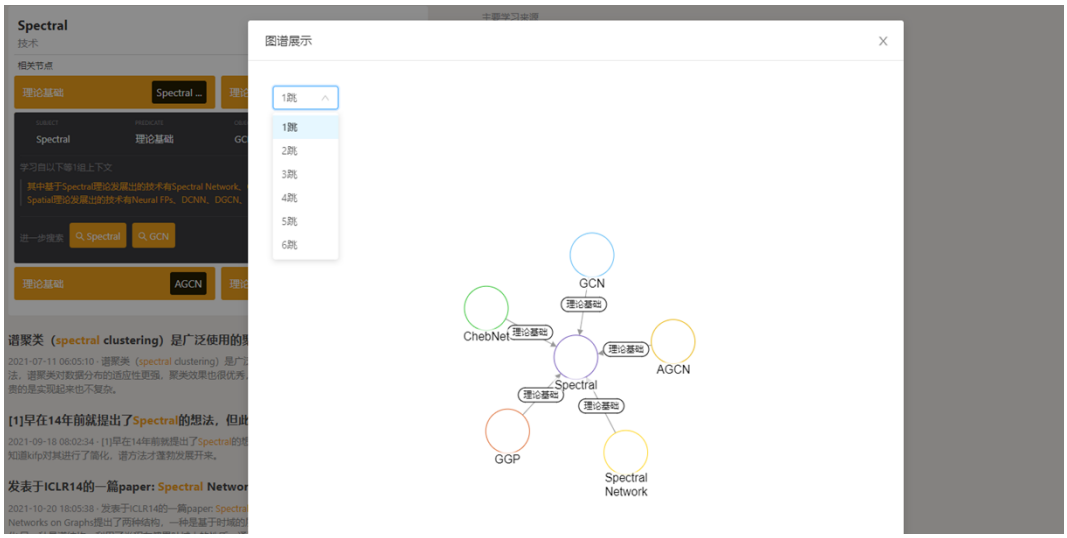

图 5.8　知识图谱展示

Fig. 5.8　Knowledge graph display

图谱展示功能如图 5.8 所示。采取知识图谱的结构展示能够使用户对当前节点的相关信息有更明了的认识，同时系统设计了以当前查询节点为中心向外扩散最多 6 跳的图谱展示方便用户能查看到更多与当前查询节点相关联的其他节点。

5.2 数据管理与维护

5.2.1 图谱数据管理

一个高质量的知识库是系统能准确回答用户问题的基本保障，因此不断更新知识库中的信息是非常有必要的。本文设计了数据导入模块的实现，管理员或有权限的用户点击首页的知识导入按钮后页面会出现接收文本文档的组件接收上传的信息，具体如图 5.9 所示。

本文为方便文本上传的过程，设计了既可以通过纯文本又可以通过 word 文档上传的功能，在文本准备工作就绪后，点击确定即可对文本进行解析。

图 5.9 文本上传

Fig. 5.9 Text upload

图 5.10 为文本解析页面，系统将上传的文本通过预设工具及预设规则解析文本，并将解析结果展现给管理员，管理员可以检查解析结果的正误，系统提供了结果修改功能，管理员通过双击文本标签或关系标签修改对应的信息。

图 5.11 为文本标签修改的基本功能展现，管理员可通过双击被标注错误的实体修改其实体类类型，关系修改和实体类标注操作相同。系统还设计了为当前选中的实体提供外网知识链接查询，方便用户查询当前标注信息的更多相关知识。

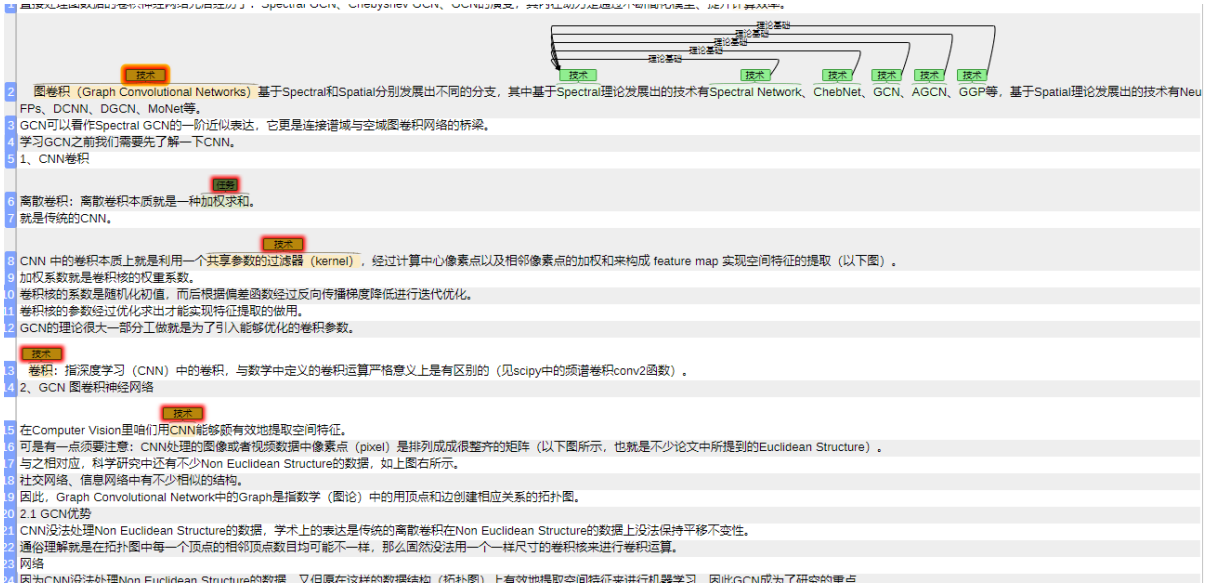

图 5.10　知识解析

Fig. 5.10　Knowledge analysis

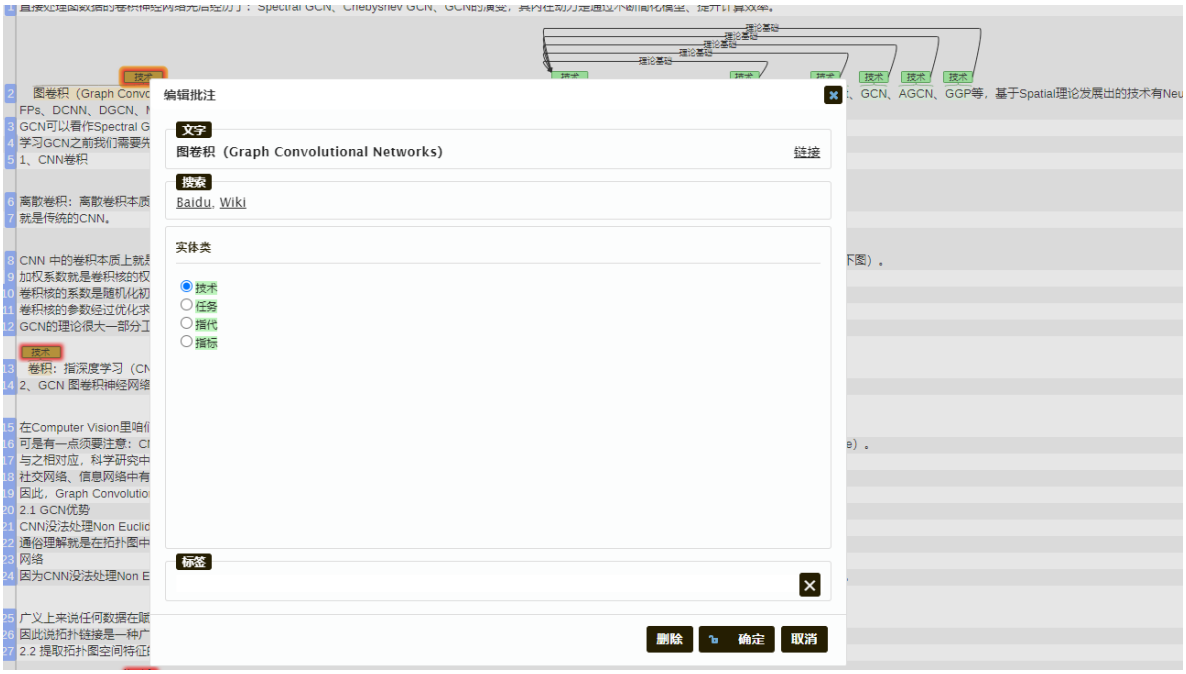

图 5.11　标注信息修改

Fig. 5.11　Annotation information modification

5.2.2 同义词维护

为解决同义词问题，本文设计了同义词库协助问句解析模块更准确地提取问句中的关键词，同义词维护页面如图 5.12 所示。

页面主要由实体类、代表词、同义词组和操作四项构成，检索功能下，管理员可以选择模糊搜索关键词或者直接检索某个实体类下的所有代表词；代表词为当前一行同义词组的统称，同义词组是与代表词有相同表示的词汇集合。

为防止用户误操作，系统设置了将合并和新增代表词两个功能按钮置灰的预操作，只有提前选中某个实体类才能新增代表词，只有选择了两条及以上相同实体类的行才能进行合并操作。

合并前要选择合并后的代表词。为简化页面，同时更好地执行当前功能，系统设置了只将多个代表词供用户选择，而不是提供所有的同义词组集合，具体如图 5.13 所示。

图 5.12 同义词维护页面

Fig. 5.12 Synonym maintenance page

图 5.14 为同义词组的编辑功能，管理员可点击每行的编辑按钮对该行的同义词组进行编辑，功能包括切换主词、在当前同义词组下新增同义词及批量删除同义词。

系统将当前同义词组的代表词置灰，用户仅能选择除当前代表词以外的其他词汇对当前词组的代表词进行更新。另外系统还设置了批量删除词汇功能，用户仅需将想要删

除的词汇选中并点击向右的箭头，就可将选中目标导入待删除列表，点击确定即可删除；在确认删除之前，选中不想删除的词汇点击向左的箭头，或者直接点击关闭键，即可撤销当前的批量词汇删除操作。

图 5.13　合并代表词

Fig. 5.13　Merge representative words

图 5.14　编辑同义词组

Fig. 5.14　Edit synonyms

5.2.3　用户管理

为保证数据入库的可靠性，本系统的数据维护功能只有管理员或者通过管理员指定权限的用户才能使用，图 5.15 为用户管理页面。

管理员可以通过输入用户名称或用户 ID 进行用户查找，点击每一行的赋予数据管理权限/取消权限即可控制普通用户对数据库维护功能的使用。

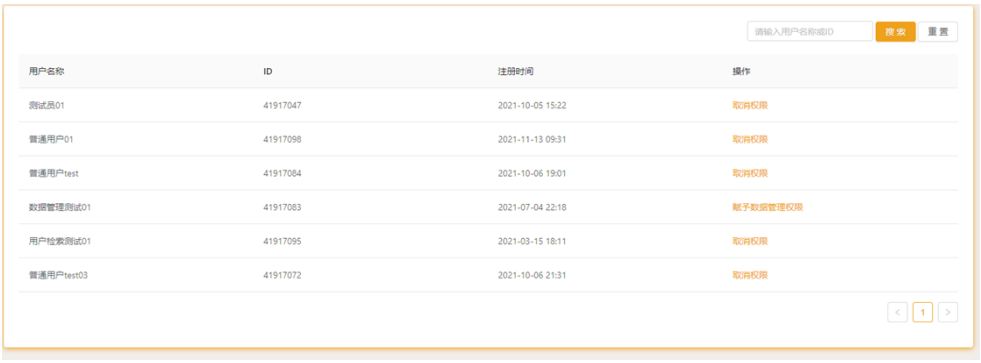

图 5.15　用户管理页面

Fig. 5.15　User management page

6 系统测试

系统测试主要为验证系统的执行结果是否符合预期，以保证系统能正常运转并提供服务，同时系统测试还能协助测试人员识别软件产品是否符合需求、功能是否缺失。对系统进行测试的目标是通过提供高质量的测试用例，尽可能多的发现系统中的错误，减少生产环境可能产生的损失。

本文采用黑盒测试的方法，在忽视软件内部结构及特性的情况下针对系统的多个功能点构建测试用例，测试系统各功能点输入输出的正确性。

表 6.1 为问答功能的测试用例。

表 6.1　问答功能测试用例表

Tab. 6.1　Q & a function test case table

序号	测试步骤	预期结果	测试结果
1	输入实体查询问句	展示与查询实体相关的节点、知识来源及相关文本	与预期结果一致
2	输入一跳单实体问句	展示正确答案	与预期结果一致
3	输入二跳单实体问句	展示正确答案	与预期结果一致
4	输入一跳双实体问句	展示正确答案	与预期结果一致
5	输入二跳双实体问句	展示正确答案	与预期结果一致
6	输入一跳三实体问句	展示正确答案	与预期结果一致
7	输入二跳三实体问句	展示正确答案	与预期结果一致
8	点击单实体答案下方对应的学习来源	弹窗加载点击文本的来源文章	与预期结果一致
9	点击进一步搜索推荐的实体	浏览器打开新的页面并展示对当前点击实体的查询结果	与预期结果一致
10	点击相关实体中的关联节点	向下展开展示当前点击节点的三元组、学习来源及推荐实体	与预期结果一致
11	点击关联节点中的学习来源	弹窗加载与点击文本的来源文章	与预期结果一致
12	点击关联节点推荐搜索实体	浏览器打开新的页面并展示对当前点击实体的查询结果	与预期结果一致
13	点击相关节点查看图谱按钮	弹窗展示以当前节点为中心的图谱展示	与预期结果一致
14	选择图谱展示中的跳数	根据选择的跳数重新渲染图谱并显示对应的数据	与预期结果一致

表 6.1　续

Tab. 6.1　Cont

序号	测试步骤	预期结果	测试结果
15	输入实体查询问句后查看右侧学习来源	学习来源会自动连接左侧与其相关的实体文本	与预期结果一致
16	点击实体集合答案下方的单个实体	浏览器打开新的页面并展示对当前点击实体的查询结果	与预期结果一致
17	查看实体集合答案下方的推荐图表	正确展示与当前集合相关的数据	与预期结果一致
18	点击右侧学习来源的标题	弹窗加载与点击标题的来源文章	与预期结果一致
19	点击下方相关文本列表标题	弹窗加载点击文本的来源文章	与预期结果一致
20	点击下方相关文本加载更多	加载更多相关文本列表	与预期结果一致
21	点击文章阅读下方加载更多	加载文章的下一段内容	与预期结果一致

表 6.2 为图谱数据维护功能的测试用例。主要测试数据上传的流程及导入结果的正确性。

表 6.2　图谱数据维护测试用例表

Tab. 6.2　Atlas data maintenance test case table

序号	测试步骤	预期结果	测试结果
1	点击首页知识导入	打开知识导入弹窗	与预期结果一致
2	点击上传文档再点击确定	打开知识解析页面并展示解析结果	与预期结果一致
3	输入文本再点击确定	打开知识解析页面并展示解析结果	与预期结果一致
4	选中文本	弹出批注编辑框	与预期结果一致
5	双击文本	弹出批注编辑框	与预期结果一致
6	鼠标点击连接两个实体	弹出批注编辑框	与预期结果一致
7	双击关系连接线	弹出批注编辑框	与预期结果一致
8	点击批注弹窗中的搜索	浏览器打开选中的搜索引擎页面	与预期结果一致
9	点击批注更换其所属类别	更换成功	与预期结果一致

表 6.3 为同义词维护功能的测试用例。主要测试同义词的检索、新增、合并及删除等主要功能。在代表次新增功能中，还要注意测试当前新增的代表词是否已存在，系统对于已存在的代表词应给出正确的反馈。

表 6.3　同义词维护测试用例表

Tab. 6.3　Synonym maintenance test case table

序号	测试步骤	预期结果	测试结果
1	点击首页同义词管理	打开同义词维护页面	与预期结果一致
2	选择模糊搜索输入关键词	展示正确的搜索结果	与预期结果一致
3	选择实体类输入关键字	展示正确的搜索结果	与预期结果一致
4	搜索后点击重置按钮	清空搜索框并重新加载所有数据	与预期结果一致
5	选择实体类后点击新增按钮	弹出代表词新增框	与预期结果一致
6	在新增代表词功能输入已存在的词	提示新建代表词是白	与预期结果一致
7	直接点击合并按钮	按钮无效不能点击	与预期结果一致
8	选择一行后点击合并按钮	按钮无效不能点击	与预期结果一致
9	选择两行及以上后点击合并按钮	弹出选择代表词弹窗	与预期结果一致
10	在合并弹窗中不选择代表词直接点击确定按钮	弹出提示选择代表词的弹窗	与预期结果一致
11	在合并弹窗中选择代表词后点击确定按钮	将选中的代表词及其同义词组合并并重新加载同义词列表	与预期结果一致
12	在合并弹窗中直接点击取消按钮	合并代表词弹窗消失	与预期结果一致
13	任选一行点击编辑按钮	弹出编辑同义词弹窗	与预期结果一致
14	在编辑同义词组弹窗直接点击切换主词	弹出先选择一个词作为主词的提示	与预期结果一致
15	在编辑同义词组弹窗选择主词后点击切换主词	弹出切换主词弹窗	与预期结果一致
16	在编辑同义词组弹窗点击新增主词	弹出新增代表词弹窗	与预期结果一致
17	在编辑同义词组弹窗批量选择同义词后点击确认	批量选择的词组被删除	与预期结果一致

表 6.4 为用户管理功能的测试用例。主要测试该模块的注册、登录及用户权限控制模块的功能。在权限控制模块，系统对不同权限的用户有不同的页面展现形式，比如首页的同义词管理和图谱数据导入功能，确保无权限用户能正常使用系统的问答功能但不能管理系统的数据模块。

表 6.4　用户管理功能测试用例表

Tab. 6.4　User management function test case table

序号	测试步骤	预期结果	测试结果
1	点击首页用户登录	跳转到登录页面	与预期结果一致
2	登录页面点击用户注册	弹出注册弹窗	与预期结果一致
3	注册页面输入账号密码进行用户注册	注册成功	与预期结果一致
4	登录页面输入账号密码进行登录	登陆成功	与预期结果一致
5	用户管理页面输入用户名称或 ID 点击搜索	检索出对应的用户	与预期结果一致
6	用户管理页面检索后点击重置	清空检索条件并重新加载用户列表	与预期结果一致
7	用户管理页面点击赋予用户数据管理权限	赋予权限成功	与预期结果一致
8	用户管理页面点击取消用户权限	取消成功	与预期结果一致
9	用户管理页面点击分页加载	展示对应页数的数据	与预期结果一致

结　论

本文以直接回答用户的问题为目标，通过将深度学习与传统问答系统流程相结合的方式，设计并实现了基于知识图谱的领域型知识问答系统。除了直接返回问句答案，同时也将答案的文本学习来源一同返回，便于用户获取更多与查询目标相关的信息。本文还设计了对不同类型答案的不同展示方式，针对实体查询将其相关节点一同返回，方便用户观察其图谱关系；针对单实体答案直接展示检索结果，同时返回学习来源；针对多实体集合答案，为给出更多与答案有关的信息，系统会在分析集合共有属性字段后给出图表分析。同时为给出更多与问句相关的信息，系统还设计了采用文本碎片匹配的方式，在答案下方列出与答案有文本重合的列表供用户查阅。

本文详细介绍了问答系统的搭建实现过程。通过调研基于知识图谱的相关问答技术，结合系统的功能需求和用例图，将系统实现分为问句答案检索、图表构建、结果可视化、图谱数据管理、同义词维护、用户管理六个模块。其中问句答案检索为系统核心模块，主要用于解析问句的语义，定位检索目标；图表构建为对于集合结果的进一步分析；结果可视化以问句检索目标为中心，展示与其相关的三元组、知识来源、相关文本、先关文章等更多信息；图谱数据管理模块与同义词维护保证了问答系统数据的更新；用户管理模块保证了系统导入数据的可靠性。系统实现后，通过系统测试模块对系统进行了测试工作，保证了系统线上运行的稳定性。

本文成功设计并实现了针对问句给出直接答案的领域型问答系统，后续会对继续系统进行一些优化维护工作，比如问句解析模块的实体链接模块会添加能提高解析效率的排序功能，针对问句答案添加多模态功能等。

参 考 文 献

[1] Do P, Phan T H V. Developing a BERT based triple classification model using knowledge graph embedding for question answering system[J]. Applied Intelligence, 2022, 52(1): 636-651.

[2] 刘峤,李杨,段宏,等.知识图谱构建技术综述[J].计算机研究与发展,2016,53(3):582-600.

[3] 田莉霞. 知识图谱研究综述[J]. 软件, 2020, 41(4):67-71.

[4] Lehmann J, Isele R, Jakob M, et al. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia[J]. Semantic Web, 2015, 6(2):167-195.

[5] Suchanek F M, Kasneci G, Weikum G. YAGO: A Large Ontology from Wikipedia and WordNet[J]. Journal of Web Semantics, 2008, 6(3):203-217.

[6] Bollacker K, Cook R, Tufts P. Freebase: a shared database of structured general human knowledge[C]. The AAAI Conference on Artificial Intelligence. Vancouver, Canada, 2007: 1962-1963.

[7] Welbl J, Stenetorp P, Riedel S. Constructing datasets for multi-hop reading comprehension across documents[J]. Transactions of the Association for Computational Linguistics, 2018, 6(3):287-302.

[8] 吴天星,漆桂林,高桓. 知识图谱研究进展[J]. 情报工程, 2017, 3(1):4-25.

[9] 岳世峰,林政,王伟平,等. 智能回复系统研究综述[J]. 信息安全学报, 2020, 5(1):20-34.

[10] Green Jr B F, Wolf A K, Chomsky C, et al. Baseball: an automatic question-answerer[C]. western joint IRE-AIEE-ACM computer conference. USA, 1961:219-224.

[11] Liu Y, Li S, Cao Y, et al. Understanding and summarizing answers in community-based question answering services[C].the 22nd International Conference on Computational Linguistics (COLING 2008), Manchester, UK, 2008: 497-504 .

[12] Liu Y, Agichtein E. On the evolution of the yahoo! answers QA community[C]. the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, 2008: 737-738.

[13] Mishra A, Jain S K. A survey on question answering systems with classification[J]. Journal of King Saud University - Computer and Information Sciences, 2018, 28(3):345-361.

[14] 陈振宇,袁毓林,张秀松,等. 亲属关系的逻辑意义及其自动推理[J]. 计算机工程与应用, 2009, 45(16):43-47.

[15] 张楚婷,常亮,王文凯,等. 基于 BiLSTM-CRF 的细粒度知识图谱问答[J]. 计算机工程, 2020, 46(2):41-47.

[16] 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.

[17] Berant J, Chou A, Frostig R, et al. Semantic parsing on freebase from question-answer pairs[C]. the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, USA, 2013:1533-1544.

[18] Lopez V, Unger C, Cimiano P, et al. Evaluating question answering over linked data[J]. Journal of Web Semantics, 2013, 21(6):3-13.

[19] Yao X, Van Durme B. Information extraction over structured data: Question answering with freebase[C]. the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, 2014: 956-966.

[20] Singh K, Radhakrishna A S, Both A, et al. Why reinvent the wheel: Let's build question answering systems together[C]. the 2018 world wide web conference, Lyon, France, 2018: 1247-1256.

[21] Wu P, Zhang X, Feng Z. A survey of question answering over knowledge base[C]. China Conference on Knowledge Graph and Semantic Computing, Springer, Singapore, 2019: 86-97.

[22] Dong L, Wei F, Zhou M, et al. Question answering over freebase with multi-column convolutional neural networks[C]. the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 2015: 260-269.

[23] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems, Long beach, USA, 2017: 5998-6008.

[24] DEVLIN J，CHANG M W，LEE K，et al. BERT：pre-training of deep bidirectional transformers for language understanding[C]. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Stroudsburg，USA，2019：4171-4186.

[25] Rogers A, Kovaleva O, Rumshisky A. A primer in bertology: What we know about how bert works[J]. Transactions of the Association for Computational Linguistics, 2020, 8(1): 842-866.

[26] C Gormley. Elasticsearch: The Definitive Guide[M]. Sevastopol: Oreilly Media, 2015.

[27] Cai Q. Research on Chinese naming recognition model based on BERT embedding[C]. 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Las Vegas, USA, 2019: 1-4.

[28] Liu W, Zhou P, Zhao Z, et al. K-bert: Enabling language representation with knowledge graph[C]. the AAAI Conference on Artificial Intelligence, New York, USA, 2020: 2901-2908.

[29] Choi H, Kim J, Joe S, et al. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks[C]. 2020 25th International Conference on Pattern Recognition (ICPR), Italy, 2021: 5482-5487.

[30] Dibia V, Demiralp Ç. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks[J]. IEEE computer graphics and applications, 2019, 39(5):33-46.

致　　谢

很幸运能在大连理工大学完成我的研究生学业，回想过去三年学习生活的点点滴滴，一切都还历历在目，研究生学习生涯的第一堂课也依然恍如昨日。三年的学习时光匆匆而过，重返校园之初，我心中感慨万千，工作后又回来继续学业的我对研究生阶段的学习机会倍感珍惜。这段学习经历也很好的磨砺了我的性格和心智，使我由刚入学的浮躁过渡到了现在的沉稳。三年来，我的老师、同学给予我的帮助使我受益良多，这一段经历是我最宝贵的财富。

首先要特别感谢单世民老师为我的研究生学习指明了方向，指导我如何研究和解决问题，对本文系统的设计和修改提出了修改和完善的宝贵建议。同时还要感谢刘宇老师和赵哲焕老师对我研究生学习的精心教导和帮助，老师们严谨的治学态度、渊博的学识和平易近人的生活作风都对我影响深远，是我以后生活学习的榜样。每当我遇到难以解决的问题，老师们都能抽出时间及时地进行指导，帮助我解决问题，衷心的对各位老师表示感谢！另外还要感谢校外实践指导老师和张鑫师兄，在我实现系统的过程中提供了许多技术指导和支持，帮我解决了很多技术上的难题，让我能及时的完成整个系统的开发工作。

感谢我的父母，能让我任性的放弃工作来读研，感谢他们对我的支持和鼓励。

论文定稿，象征着我的学生身份或将结束，但学习之心态会永远存在，不管生活还是学习，都是过程的代表而非简单结果的描述，这篇文章是我之前学习的阶段性谢幕，也是我后续学习历程的开启。

大连理工大学学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于大连理工大学，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以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。

学位论文题目：_____

作 者 签 名 ：_____ 日期：_____年____月日

导 师 签 名 ：_____ 日期：_____年____月日