|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类号 |  | |  | | | 密 级 | |  |
| U D C |  | |  | | | 编 号 | | 10486 |
|  | | | | | | | | |
| **硕 士 专 业 学 位 论 文** | | | | | | | | |
|  | | | | | | | | |
| 基于Neo4j的图数据查询分析系统  ——以里耶秦简数据为例 | | | | | | | | |
|  | | 研究生姓名 | | ： | 陆咏 | |  | |
| 学号 | | ： | 2019282110211 | |
| 指导教师姓名、职称 | | ： | 祝园园 副教授 | |
| 专业类别（领域） | | ： | 软件工程 | |
|  | |  |  | |
|  | |  |  | |
|  | | | | | | | | |
| 二〇二一年五月 | | | | | | | | |

Graph Data Query and Analysis System Based on Neo4j——Take Liye Qin Bamboo Slips as an Example

By

Lu Yong

Supervised by

Associate Professor ZHU Yuanyuan

Wuhan University

May 2021

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。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。本声明的法律结果由本人承担。

学位论文作者（签名）：

年 月 日

# 摘要

近年来，随着社交网络、零售电商、金融风控、生命科学等行业的快速发展，需要处理的数据之间的关系随数据量呈几何级数增长，越来越多的数据以图数据的形式进行保存和处理。

由于图数据的复杂性和理解难度较其他类型的数据显著提高，不管是国内还是国外，图数据的可视化查询分析都有着广泛的应用需求。目前，图数据可视化查询分析平台层出不穷、功能日渐强大，但也存在一定的局限。这些平台大部分偏向于工业应用，对学术方面的支持较少，部分平台局仅针对特定领域，使用门槛较高。因此，亟需一个易于使用的图可视化系统，帮助用户直观、快速、灵活地查询和分析图数据，从而发现其中蕴含的价值。针对这一问题，本文构建了一个通用的图数据查询分析系统。

文章基于图数据在各个领域广泛存在的现实情况和当前我国古籍保护与数字化的需求，以里耶秦简数据为切入点，构建了系统使用的里耶秦简知识图谱。对里耶秦简原始数据，使用BiLSTM-CRF模型进行命名实体识别，使用FP-Growth算法和BiGRU-Attention模型抽取实体间的关系，使用SimRank和SimRank++算法与统计分析聚类进行命名实体消歧、实现人物分组。然后，在此基础上，借助Neo4j图数据库、Neovis.js、Vue等工具，构建图数据查询分析系统。根据图数据的特点，设计实现了图数据可视化展示、顶点与边的查询、图数据算法调用、图数据上传等模块，包括图结构可视化展示、顶点信息展示、边信息展示、顶点分类查询、顶点id查询、顶点名称查询、时间轴展示、自定义查询、运行指定算法、自定义调用、图数据上传等功能，满足了用户多样化的查询和深度分析的需求。

**关键词：**图可视化；SimRank；Neo4j；里耶秦简

# Abstract

In recent years, with the rapid development of social networking, retail ecommerce, financial risk control, life science and other industries, the relationship between data to be processed increases geometrically with the amount of data, and more and more data are saved and processed in the form of graph.

Because of the complexity of graph and the difficulty of understanding graph structure, visualized query and analysis of graph data has a wide range of application requirements both at home and abroad. At present, graph data visualization query and analysis platform emerge in endlessly, the function is increasingly powerful, but there are some limitations. Most of these platforms are inclined to industrial applications, with less academic support and some platforms are only for specific areas, with a high threshold of use. Therefore, an easy-to-use graph visualization system is urgently needed to help users query and analyze graph data intuitively, quickly and flexibly, so as to discover its value. To solve this problem, this paper constructs a general graph visualization query and analysis system.

Based on the reality that graph data widely exists in various fields and the current needs of ancient books protection and digitization in China, this paper constructs the knowledge graph of Liye Qin bamboo slips. Based on the original data of Liye Qin bamboo slips, this paper uses BiLSTM-CRF model for named entity recognition, uses FP-Growth algorithm and BiGRU-Attention model to extract the relationship between entities, and uses SimRank and SimRank++ algorithm and statistical analysis clustering for named entity disambiguation and character grouping. Then, on this basis, with the help of Neo4j graph database, Neovis.js and Vue to build the graph visualization query and analysis system. According to the characteristics of graph data, the system designs and implements four modules — graph data visualization, vertex and edge query, graph data algorithms call and graph data upload, including graph structure visualization, vertices information display, edges information display, vertices classification query, vertices ID query, vertices name query, timeline display, customized query, call specified algorithm, call customized algorithm, graph data upload and other functions, with which perfectly meets users' varied query and deep analysis needs.

**Keywords:** Graph Visualization; SimRank; Neo4j; Liye Qin Bamboo Slips

# 目录

[摘要 I](#_Toc68555945)

[Abstract II](#_Toc68555946)

[目录 III](#_Toc68555947)

[1 绪论 1](#_Toc68555948)

[1.1 研究背景 1](#_Toc68555949)

[1.2 国内外研究现状 2](#_Toc68555950)

[1.2.1 知识图谱研究现状 2](#_Toc68555951)

[1.2.2 图可视化系统研究现状 2](#_Toc68555952)

[1.3 研究内容 4](#_Toc68555953)

[1.4 论文组织结构 4](#_Toc68555954)

[2 相关概念与技术 6](#_Toc68555955)

[2.1 主流图数据库对比 6](#_Toc68555956)

[2.2 Neo4j图数据科学库 8](#_Toc68555957)

[2.3 知识图谱 9](#_Toc68555958)

[2.4 古籍保护与数字化 11](#_Toc68555959)

[2.5 本章小结 12](#_Toc68555960)

[3 里耶秦简知识图谱构建 13](#_Toc68555961)

[3.1 实验数据 13](#_Toc68555962)

[3.2 命名实体识别 14](#_Toc68555963)

[3.3 关系抽取 16](#_Toc68555964)

[3.4 命名实体消歧 19](#_Toc68555965)

[3.5 本章小结 20](#_Toc68555966)

[4 图数据查询分析系统设计与实现 22](#_Toc68555967)

[4.1 系统需求分析 22](#_Toc68555968)

[4.2 系统详细设计 23](#_Toc68555969)

[4.3 图数据可视化展示模块 25](#_Toc68555970)

[4.3.1 图结构可视化展示 26](#_Toc68555971)

[4.3.2 顶点信息展示 27](#_Toc68555972)

[4.3.3 边信息展示 30](#_Toc68555973)

[4.4 顶点与边的查询模块 32](#_Toc68555974)

[4.4.1 顶点分类查询 32](#_Toc68555975)

[4.4.2 顶点id查询 33](#_Toc68555976)

[4.4.3 顶点名称查询 34](#_Toc68555977)

[4.4.4 自定义查询 38](#_Toc68555978)

[4.5 图数据算法调用模块 38](#_Toc68555979)

[4.5.1 运行指定算法 39](#_Toc68555980)

[4.5.2 自定义调用 41](#_Toc68555981)

[4.6 图数据上传模块 41](#_Toc68555982)

[4.7 本章小结 42](#_Toc68555983)

[5 实验与结果分析 43](#_Toc68555984)

[5.1 实验环境 43](#_Toc68555985)

[5.2 实验与结果分析 43](#_Toc68555986)

[5.2.1 里耶秦简命名实体消歧结果分析 43](#_Toc68555987)

[5.2.2 系统功能测试与结果分析 45](#_Toc68555988)

[5.2.3 系统性能测试与结果分析 46](#_Toc68555989)

[5.3 本章小结 47](#_Toc68555990)

[6 总结与展望 48](#_Toc68555991)

[6.1 本文总结 48](#_Toc68555992)

[6.2 未来工作展望 48](#_Toc68555993)

[6.2.1 优化与改进 48](#_Toc68555994)

[6.2.2 应用场景 49](#_Toc68555995)

[参考文献 50](#_Toc68555996)

[致谢 55](#_Toc68555997)

# 1 绪论

本章简要叙述了课题研究的主要背景、目前图数据查询分析系统的国内外研究现状、论文的研究内容和组织结构。

## 1.1 研究背景

近年来，随着社交网络、零售电商、金融风控、生命科学等行业的快速发展，需要处理的数据之间的关系随数据量呈几何级数增长，并且，越来越多的数据以庞大而复杂的关系网的形式出现，不得不以图数据（Graph Data）的形式进行保存和处理。

图是描述现实世界中离散个体之间关系的有用工具。图是一种复杂的非线性结构，由顶点及顶点间的关系组成。在离散数学中，图G是由两个集合V和E组成，记做G=（V，E），其中，V是vertex的缩写，在这里代表结点的集合，E是edge的缩写，在这里代表边的集合。线性表中的元素是“一对一”的关系，树中的元素是“一对多”的关系。图是树的推广，在图结构中，元素是“多对多”的关系，每个元素都可以有零个或多个前驱，也可以有零个或多个后继，元素之间的关系是任意的[1]。

由于图的复杂性，与其他数据结构相比，理解图的难度显著提高。因此，亟需一个图可视化系统，帮助用户直观、快速、灵活地查询和分析图数据，从而发现其中蕴含的价值。

传统的关系型数据库检索时需要进行多个表之间的连接，写入数据时也需要考虑外键约束，以保证数据的完整性和一致性。随着数据量的增多和关联深度的增加，传统的关系型数据库会产生较大的额外开销，也很有可能无法在规定的时间内计算出查询结果，导致严重的性能问题。因此，图数据查询分析系统有必要采用其他类型的数据库——图数据库。图数据库是非关系型数据库中的一种，它使用灵活的图形数据模型，将关系信息存储为实体，并具备固有的数据索引结构。使用图数据库能够显著提高图数据的存储、操作和访问速度，也能进行比实时查询更为复杂的分析操作，进一步挖掘图数据中的潜在价值。

## 1.2 国内外研究现状

### 1.2.1 知识图谱研究现状

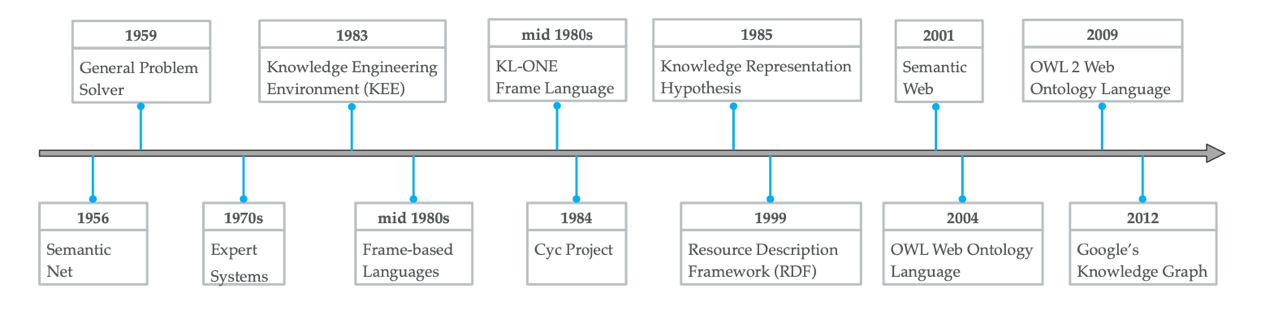
知识图谱（Knowledge Graph）的概念最初由Google公司在2012年正式提出[2]，是指Google用于提升其搜索引擎性能的知识库（Knowledge Base）。但是，知识图谱的发展历程却可以追溯到1956年的语义网络（Semantic Net），中间经历了一系列的演变，如图1-1[3]所示，最终形成了今天的知识图谱。现在，知识图谱泛指利用图结构的数据模型或拓扑结构来集成数据的知识库。

图1-1 知识图谱发展历程

关于知识图谱的研究包括很多方面。2015年，Maximilian Nickel等人研究了知识图谱的统计关系学习模型，将其用于大型知识图谱审查，并尝试了将统计关系学习与机器阅读和信息提取方法结合起来，自动构建知识图谱[4]。2017至2018年，Paulheim Heiko针对知识图谱精化问题，调研了各种知识图谱自动补全和错误检测方法[5]；王泉等人研究了知识图谱嵌入的各项方法和应用[6]；Yankai Lin等人对知识表征学习（Knowledge Representation Learning）的方法进行了评估[7]；Tianxing Wu等人介绍了中文知识图构造及其应用[8]。2020年，Xiaojun Chen等人综述了知识图谱推理的相关方法和应用[9]；Shaoxiong Ji等人深入研究了知识表征学习、知识获取（Knowledge Acquisition）、知识感知应用（Knowledge-aware Applications）等问题[3]。

当前，知识图谱广泛应用于语义搜索、问答系统、推荐系统、辅助决策等领域。

### 1.2.2 图可视化系统研究现状

在图分析中，系统最前端的人机交互的技术方案就是图可视化技术。系统中的图数据通过计算机图形学和图像处理技术，转化成图像或图像，完成信息展示、交互等功能。用户借助图可视化技术，获取直观的图数据结构图像，进而可以实现快速发现趋势和异常值、获取更深入的信息、发现更多问题。

当前，不管是国内还是国外，图数据的可视化查询分析都有着广泛的应用需求。

Linkurio于2019年在其官网发布《Graph visualization software and web applications》，总结了国外的一百多个图产品[10]，包括本地软件、Web应用程序和Saas（Software-as-a-Service），涵盖了通用领域和金融风控、医疗健康、生物信息、社交网络、知识图谱、网络安全等多个特定领域。

Cambridge Intelligence就是其中的一个典型。2011年成立的Cambridge Intelligence公司，专注于为企业提供专业的图可视化技术工具，通过旗下的三款图分析工具：JavaScript图可视化库Keylines、React图可视化组件库ReGraph和时序分析组件库KronoGraph，完成多个业务领域的各种分析任务[11]。

国内也陆续推出了一些图数据可视化查询分析平台。

2006年，清华大学唐杰教授率领团队推出科技情报大数据挖掘与服务系统平台AMiner，并持续更新。AMiner利用数据挖掘和社会网络分析与挖掘技术，提供研究者语义信息抽取、面向话题的专家搜索、权威机构搜索、话题发现和趋势分析、基于话题的社会影响力分析、研究者社会网络关系识别、即时社会关系图搜索、研究者能力图谱、审稿人推荐在内的众多功能[12]，提供了溯源树、AI发展河流图、机器学习趋势图等多种图数据可视化实现，具有很高的学术价值和实用性。

2018年，香港中文大学与香港浸会大学的学者搭建了社区搜索可视化系统VizCS （Visualization system for Community Search）[13]，使用TCP（Triangle Connectivity Preserved）索引维护动态图，支持运行k-truss、k-core、quasi-clique、稠密子图等社区搜索算法，并且用户可以通过上传文件更新图数据，同时借助JavaScript库D3.js对运行结果进行可视化展示。

2020年11月，蚂蚁集团重磅推出AntV图可视分析解决方案。AntV拥有强大的产品矩阵：G2可视化引擎、G2Plot图表库、G6图可视化引擎、Graphin组件库、F2移动可视化方案、L7地理空间数据可视化、ChartCube图表魔方等，实现了统计数据、关系数据、地理数据等可视化技术解决方案，涵盖云安全、知识图谱、企业风控、图数据库等多个子领域[14]。

目前，图数据可视化查询分析平台层出不穷、功能日渐强大，但也存在一定的局限。上述平台大部分偏向于工业应用，对学术方面的支持较少，部分平台局仅针对特定领域，使用门槛较高。

## 1.3 研究内容

本课题的目的在于构建一个通用的图数据查询分析系统，实现图数据的可视化，并支持用户的自定义查询，同时加入图数据相关算法，以满足用户的深度分析需求。

本课题主要工作如下：

（1）介绍了图数据查询分析系统的研究背景和国内外研究现状，包括图数据、图可视化系统和图数据库等内容。

（2）介绍了图数据查询分析系统的相关概念与技术，包括主流图数据库对比、Neo4j图数据科学库、知识图谱以及古籍保护与数字化的相关内容。

（3）基于图数据在各个领域广泛存在的现实情况和当前古籍数字化的需求，以里耶秦简数据为切入点，对里耶秦简原始数据进行了命名实体识别、关系抽取和命名实体消歧等处理，构建知识图谱。

（4）借助Neo4j图数据库、Neovis.js、Vue等工具，根据图数据的特点，实现了图数据查询分析系统，包含图数据可视化展示、顶点与边的查询、图数据算法调用、图数据上传等功能，满足用户多样化的查询和深度分析的需求。

## 1.4 论文组织结构

本文组织结构如下：

第一章为绪论，简要叙述了课题研究的主要背景、目前图数据查询分析系统的国内外研究现状、论文的研究内容和组织结构。

第二章为相关概念与技术，主要介绍了课题研究的相关概念与技术，包括主流图数据库对比、Neo4j图数据科学库、知识图谱以及古籍保护与数字化的相关内容。

第三章为里耶秦简知识图谱构建，详细叙述了构建里耶秦简知识图谱的主要工作内容和使用的技术方法，包括实验数据介绍和研究框架确定、命名实体识别、关系抽取和命名实体消歧等。

第四章为图数据查询分析系统设计与实现，详细说明了图数据查询分析系统具体的设计与实现过程，包括系统需求分析、系统整体框架设计、图数据可视化展示模块、顶点与边的查询模块、图数据算法调用模块和图数据上传模块等。

第五章为实验与结果分析，简要说明了课题针对图数据查询分析系统的构建目标，设计、进行了相关测试实验，并对实验结果进行了分析。

第六章为总结与展望，对本文的研究内容进行了总结，指出了图数据查询分析系统存在的不足，思考了未来改进方向和工作展望。

# 2 相关概念与技术

本章主要介绍了课题研究的相关概念与技术，包括主流图数据库对比、Neo4j图数据科学库、知识图谱以及古籍保护与数字化的相关内容。

## 2.1 主流图数据库对比

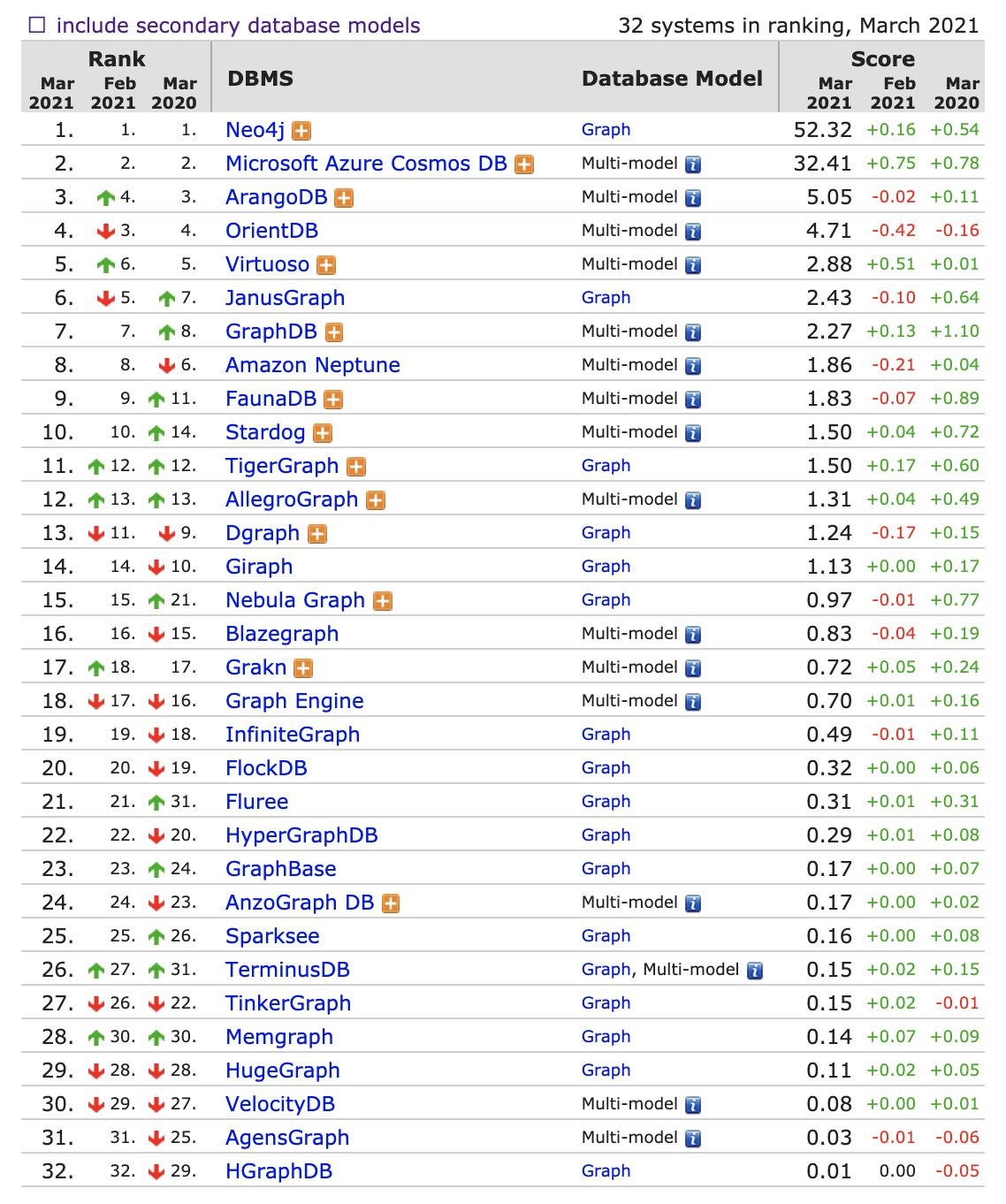
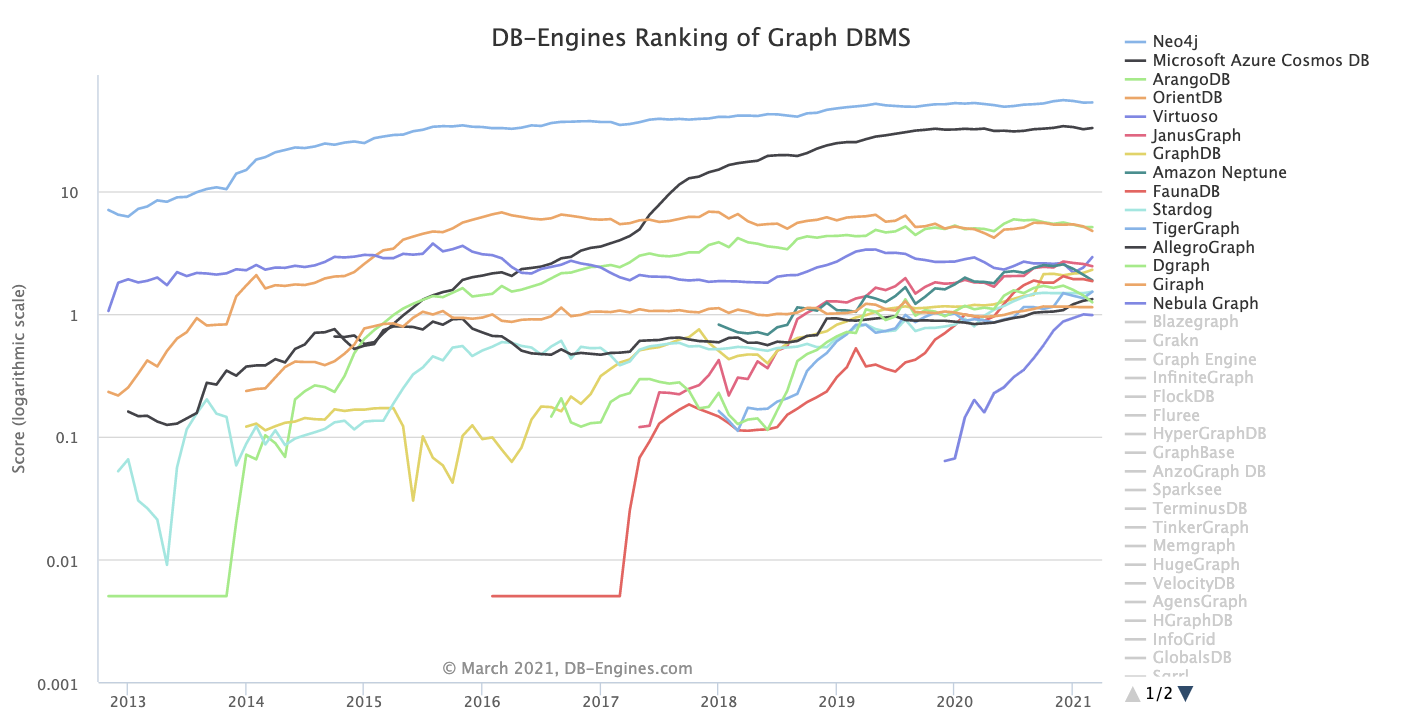
当前市场上的图数据库种类繁多，根据2021年3月DB-Engine的图数据库排名，当前市场占有率较高的图数据库有Neo4j、JanusGraph、TigerGraph、Dgraph、Giraph等，如图2-1[15]、图2-2[16]所示。

图2-1 2021年3月DB-Engine图数据库排名

图2-2 2021年3月DB-Engine图数据库排名折线图

在当前市场上的众多图数据库中，Neo4j长期稳居图数据库市场占有率第一。Neo4j是高性能的原生图图数据库，它功能强大、服务稳定、社群活跃，主要有以下几个优点：

（1）使用原生图（Native Graph）存储和处理数据，能够提供最优化的关系遍历执行效率，比关系数据库的表连接快上千倍；

（2）实现了完全ACID（Atomicity，Consistency，Isolation，Durability）兼容，保证数据一致性，适用于事务型（OLTP）和分析型（OLAP）应用；

（3）使用基于因果集群（Causal Clustering）的分布式数据库，提供高可用性、故障切换、数据冗余和可扩展的吞吐量；

（4）拥有丰富的语言驱动，包括Java, JavaScript, Python, .Net, GO, C/C++, R等语言，各类工具齐全[17]；

（5）内置Neo4j图数据科学库（Graph Data Science Library），能够直接执行部分基础的图数据算法。

JanusGraph是Linux基金会下的一个开源分布式图数据库，采用第三方存储（Apache Cassandra、Apache HBase、Bigtable、Berkeley DB等）作为底层存储；通过与大数据平台（Apache Spark，Apache Giraph，Apache Hadoop等）集成，支持全局图数据的分析、报告和ETL；通过外部索引存储（Elasticsearch，Solr，Lucene等）支持地理、数字范围和全文搜索。JanusGraph可以配合多种工具一起使用，具有很高的扩展性，但是由于需要开发者自行组合，使用时的技术门槛较高，图数据可视化也需要借助外部系统才能实现。

TigerGraph使用原生并行图系统，将图表示为一个计算模型，实现了大规模并行和快速计算，可以进行实时图分析和深度关联多步分析任务。

Dgraph是基于GO语言开发的开源分布式图数据库，使用原生图存储数据，性能优异；可以自动组成集群，运维部署非常简单；原生支持全文检索、正则表达式、地理位置检索、可视化等功能。但是存在不支持多重边、与大数据生态兼容不足、一个集群只支持一个图等问题。

Giraph是以Apache Hadoop为基础开发的迭代图处理框架，由Pregel模型[18]改进而来，其系统架构和计算模型与Pregel模型一致，主要用于大型图的分布式计算问题，例如，目前Facebook使用Giraph分析由用户及其关系形成的社交网络。

本课题旨在构建一个通用的图数据查询分析系统，实现图数据可视化展示、顶点与边的查询、图数据算法调用、图数据上传等功能，以图的遍历、图的查询、算法调用和图可视化为主要功能，图数据的更新迭代较少。综合考虑以上几种图数据库的特点，使用Neo4j图数据库最为合适。

## 2.2 Neo4j图数据科学库

Neo4j图数据科学库（Graph Data Science Library）是Neo4j官方推出的图数据算法库，它为Neo4j图数据库提供了通用、高效、并行的图数据算法。Neo4j图数据科学库使用了一种专门的内存图格式来表示图数据，以优化算法运行效率。

如图2-3[19]所示，Neo4j图数据科学库目前提供了六大类图数据算法：

（1）社区检测算法（Community Detection Algorithms）：社区检测算法基于关系对图进行聚类，以查找成员联系紧密的社区。社区检测有助于预测相似的行为，找到重复的实体，或者为其他分析准备数据。包括Louvain算法、标签传播算法、弱连通分量算法、三角计数算法、局部聚类系数算法等。

（2）中心性算法（Centrality Algorithms）：中心性算法根据图的拓扑结构确定网络中不同结点的重要性，具有广泛的用途。包括PageRank算法、中介中心性算法等。

（3）相似性算法（Similarity Algorithms）：相似性算法使用集合比较，根据结点的邻居或属性，计算单个结点之间的相似性，常用于个性化推荐等方面。包括结点相似性算法、KNN（K-Nearest Neighbor）算法等。

（4）链接预测算法（Link Prediction Algorithms）：链接预测算法考虑图中结点的接近度和结构关系，例如结点之间可能的三角形，以预测未来形成新链接或存在未记录链接的可能。包括Adamic Adar算法、共同邻居算法、优先依附算法、资源分配算法等。

（5）路径搜索算法（Path Finding Algorithms）：路径搜索算法可以在结点之间找到最有效或最短的路径，应用于IP路由计算、最低成本计算等方面。包括Dijkstra算法、A\*算法、随机游走算法、最小权重生成树、最短路径等。

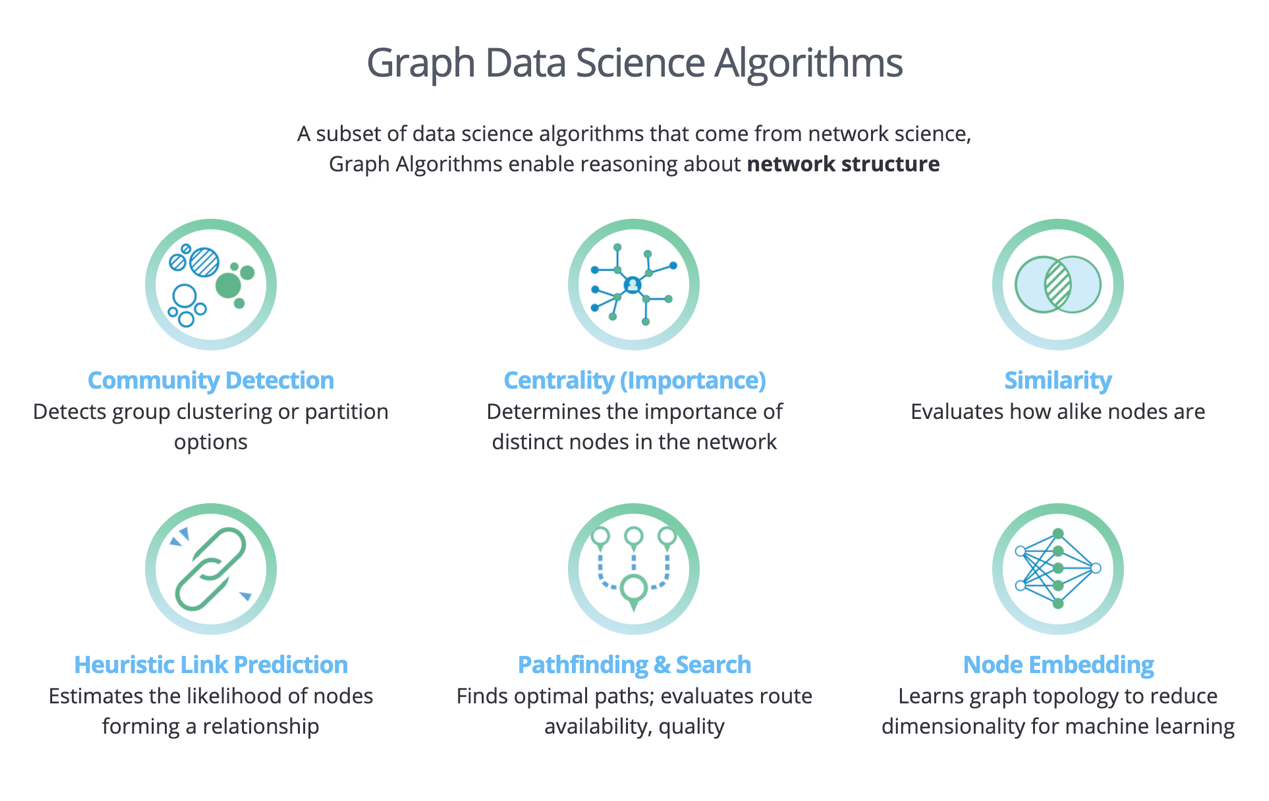
（6）结点嵌入算法（Node Embedding Algorithms）：结点嵌入算法将图的拓扑和特征转换为唯一的表示每个结点的固定长度向量。图嵌入保留了图的关键特征，并且以可解码的方式降低了维数，将其转换为各种机器学习任务。包括Node2Vec算法、FastRP算法、GraphSAGE算法等。

图2-3 Neo4j图数据科学库算法

## 2.3 知识图谱

在本文的1.2.1节中已经提到，知识图谱的概念最初由Google公司在2012年正式提出，但是，它的发展历程却可以追溯到1956年的语义网络（Semantic Net），中间经历了一系列的演变，最终形成了今天的知识图谱。

知识图谱（Knowledge Graph）是事实的结构化表示，包括实体（Entities）、关系（Relationships）以及语义描述（Semantic Descriptions）。实体可以是真实世界中的物体，也可以是抽象的概念。关系表示了实体间的关系。关于实体和实体间关系的语义描述包含了预先定义好的类型和属性[3]。知识图谱的基本组成单位是“实体—关系—实体”三元组，以及实体及其相关属性—值对，实体间通过关系相互联结，构成网状的知识结构。

知识图谱是一种特殊的图数据。具体来说，知识图谱是一种带标记的有向属性图。知识图谱中每个结点都有若干个属性和属性值，实体与实体之间的边表示的是结点之间的关系，边的指向方向表示了关系的方向，而边上的标记表示了关系的类型。

通常,可以把知识分为领域知识、百科知识、场景知识、语言知识以及常识知识等。根据所存储的知识的类型，将知识图谱划分为通用知识图谱和领域知识图谱。通用知识图谱的知识覆盖面广，包罗万象，构建难度较大；领域知识图谱的知识集中于某一个特定的领域，构建起来相对容易。

知识图谱由数据层（Data Layer）和模式层（Schema Layer）构成。模式层构建在数据层之上，是知识图谱的核心，通常采用本体库来管理知识图谱的模式层。本体是结构化知识库的概念模板，通过本体库而形成的知识库不仅层次结构较强，并且冗余程度较小。数据层主要是由一系列的事实组成，而知识将以事实为单位进行存储,通常以“实体—关系—实体”或“实体—属性—属性值”的三元组存储，形成一个图结构的知识库。

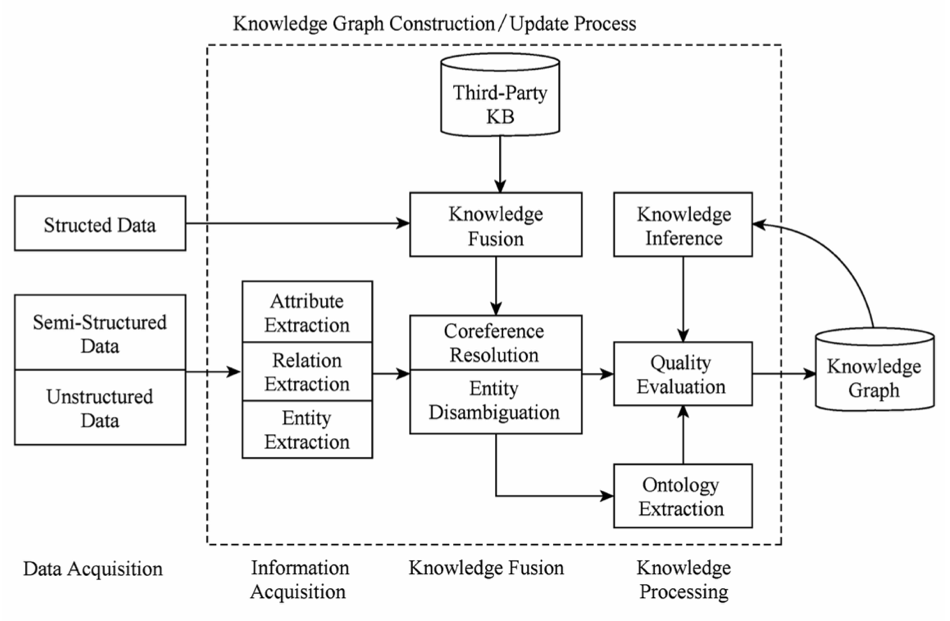
知识图谱的构建方法主要有两种：一种是自底而上的构建方法，一种是自顶而下的构建方法。自底向上的构建方法流程如图2-4所示[20]，从开放链接的数据源中提取实体、属性和关系，加入到知识图谱的数据层；然后将这些知识要素进行归纳组织，逐步往上抽象为概念，最后形成模式层。自顶而下的方法正好相反。

图2-4 知识图谱的技术架构

具体来说，知识图谱自底而上的构建方法主要有四个步骤：

（1）数据获取（Data Acquisition）：知识图谱数据源根据数据结构的不同，可分为三种：结构化数据、半结构化数据和非结构化数据。不同的数据类型，采用不同的方法进行处理。

（2）信息抽取（Information Extraction）：信息抽取从各种类型的数据源中提取出实体、属性以及实体间的相互关系，在此基础上形成本体化的知识表达。信息抽取的难点主要在从半结构化数据和非结构化数据中提取出结构化信息，涉及的关键技术包括实体抽取、关系抽取和属性抽取等。

（3）知识融合（Knowledge Fusion）：知识融合旨在解决如何将关于同一个实体或概念的多源描述信息融合起来。在获得新知识之后，需要对其进行整合，以消除矛盾和歧义，比如某些实体可能有多种表达，某个特定称谓也许对应于多个不同的实体等。在这个过程中，主要关键技术包含指代消解、实体消歧等。

（4）知识加工（Knowledge Processing）：海量数据在经信息抽取、知识融合之后得到一系列基本的事实表达，对于经过融合的新知识，需要经过质量评估之后（部分需要人工参与甄别），才能将合格的部分加入到知识库中，以确保知识库的质量。知识加工主要包括本体抽取、知识推理和质量评估等方面的内容。

本课题将以里耶秦简数据为切入点，对里耶秦简原始数据进行了实体识别、关系抽取和实体消歧等处理，构建知识图谱，获得图数据。

## 2.4 古籍保护与数字化

我国是历史悠久的文明古国，拥有卷帙浩繁的古代文献典籍，这些古籍是中华民族的宝贵精神财富。我国古代文献典籍是中华民族在数千年历史发展过程中创造的重要文明成果，但是在现代古籍却面临着老化破损严重、修复保护不足、缺乏学者解读、鲜少有人阅读等问题。

2007年，国务院办公厅提出“中华古籍保护计划”，自此，我国大力开展古籍普查登记、古籍数字化、古籍资源共享、古籍出版和研究、古籍修复和保护、古籍保护人才培养、古籍保护学科体系建设等工作，并取得大量成果。其中，古籍数字化更是承上启下的重要一环[21]。

古籍数字化是以利用和保护古籍为目的，制作古籍文献书目数据库和古籍全文数据库，用以揭示古籍文献信息资源，是古籍整理工作的一部分[22]。我国古籍时间跨度和空间跨度都很大，大量文献使用不同的字体和语法记录，必须由专家分析解读，这就导致了当前领域专家们工作量巨大，且大量古籍仍处于无人问津的状态。因此，很有必要借助机器学习和数据挖掘技术，对海量的古籍数据进行快速的知识提炼和挖掘分析，构建古籍知识图谱，以减少领域专家手动分析的工作量，同时也能更规范地整理古籍资源，方便民众查阅。

熊晶等人在2015年针对甲骨文专家知识的共享及重用程度低的问题，提出了构建甲骨文知识图谱的解决方案，借助科学知识图谱MKD（Mapping Knowledge Domains），发现甲骨文知识实体及实体之间的关系[23]。2017年，于彤等人也进行了中医临床知识图谱的构建，将古代医学典籍和现代科研成果纳入其中[24]。

里耶为战国秦代古城，现位于湖南省境内，里耶秦简则是当时的地方档案，涵括户口、土地开垦、物产、田租赋税、劳役徭役、仓储钱粮、兵甲物资、道路里程、邮驿津渡管理、奴隶买卖、刑徒管理、祭祀先农以及教育、医药等相关政令和文书，是继秦始皇兵马俑之后秦代考古的又一重大发现，其研究成果将大大填补史料的缺佚。本课题将对里耶秦简数据进行知识图谱的构建。

## 2.5 本章小结

本章主要介绍了论文相关的概念与技术。首先对比了当前市场主流的图数据库，分析了各个数据库的特点，论述了课题选用Neo4j图数据库的原因；其次介绍了Neo4j图数据库图数据科学库，展示了其强大的内置算法库；再次简述了知识图谱的相关概念及其构建步骤，说明了本课题需要进行的知识图谱构建工作；最后介绍了我国古籍保护及其数字化工作，本课题将以里耶秦简数据为切入点，构建其知识图谱。

# 3 里耶秦简知识图谱构建

本章详细叙述了构建里耶秦简知识图谱的主要工作内容和使用的技术方法，包括实验数据介绍和研究框架确定、命名实体识别、关系抽取和命名实体消歧等。

## 3.1 实验数据

本课题使用的里耶秦简数据主要来自《里耶秦簡分類舉例》《稟食記錄簡》《里耶秦簡補充語料》和《〈里耶秦簡（貳）〉所見人名統計表》[25]。

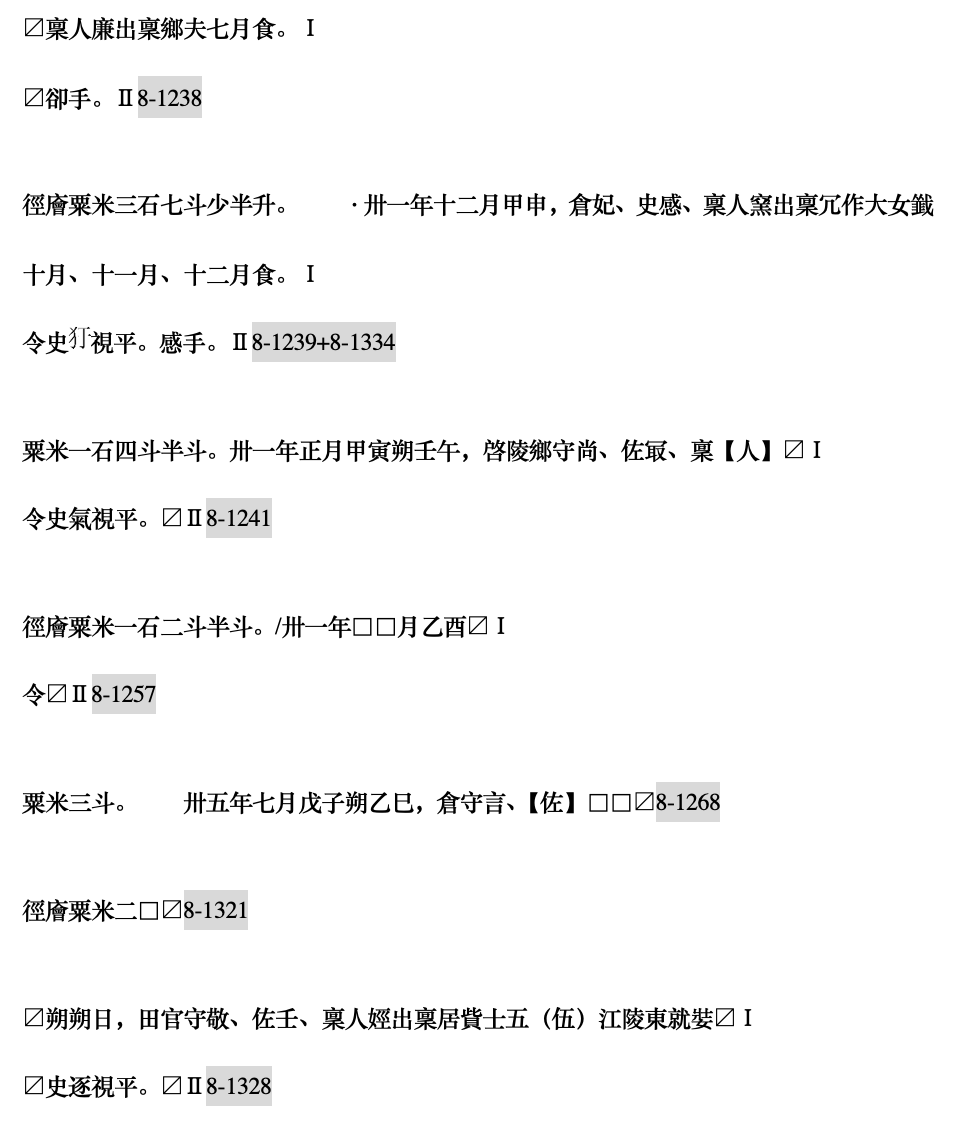
里耶秦简语料以简号为单位进行组织，如图3-1所示：繁体中文为语料正文；灰色底色部分表示本段文字所属的简号，如“8-1238”、“8-1239+8-1334”；“Ⅰ”和“Ⅱ”分别表示一支简的正面和背面。

图3-1 里耶秦简语料示例

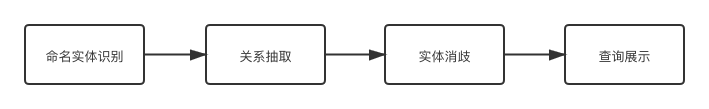
为了实现将复杂晦涩的简帛文本转换为系统规范且便于分析的知识的研究目标，首先，需要抽取人物、属性、事件、发生时间等关键信息；其次，分析上述关键信息之间的关联关系并建立关系图；再次，在关系图中进行实体消歧，并构建知识图谱；然后，实现人物的生平事迹的有效查询和组织；最后，进行可视化展示以便于人工判读与二次分析。具体研究框架如图3-2所示。

图3-2 研究框架

## 3.2 命名实体识别

命名实体识别（Named Entity Recognition，NER），又称作“专名识别”，是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。命名实体识别分为两步，第一步识别出实体词边界，也就是实体的开始位置和结束位置；第二步确定实体的类型，也就是上文所说的人名、地名等具体的实体类型。

当前命名实体识别的主要技术方法有基于规则和词典的方法、基于传统机器学习的方法和基于神经网络的方法等。

基于规则和词典的方法是命名实体识别中最早使用的方法，通常采用由语言学家通过人工方式，依据数据集特征构建的特定规则模板或者特殊词典，选用特征包括统计信息、标点符号、关键字、指示词和方向词、位置词(如尾字)、中心词等。这类方法大多依赖于知识库和词典的建立，使用匹配的方式对文本进行处理以实现命名实体识别。基于规则和词典的方法需要耗费大量人力，而且局限性较高，受限于具体语言、领域和文本风格[26]。

基于传统机器学习的方法主要包括隐马尔可夫模型（Hidden Markov Model， HMM）[27]、最大熵(Maximum Entropy，ME)[28]、最大熵马尔可夫模型(Maximum Entropy Markov Model，MEMM)[29]、支持向量机(Support Vector Machine，SVM)[30]、条件随机场(Conditional Random Fields，CRF)[31]等。基于传统机器学习的方法将命名实体识别当作序列标注问题处理，对特征选取的要求较高，需要从文本中选择对该项任务有影响的各种特征，并将这些特征加入到特征向量中。依据特定命名实体识别所面临的主要困难和所表现出的特性，考虑选择能有效反映该类实体特性的特征集合。主要是通过对训练语料所包含的语言信息进行统计和分析，从训练语料中挖掘出特征。基于传统机器学习的方法对语料库的依赖较大，因而局限性也较高。

随着深度学习的不断发展，命名实体识别的研究重点逐渐转向深层神经网络，该技术几乎不需要特征工程和领域知识[32]。基于传统机器学习的方法主要包括卷积神经网络(Convolutional Neural Networks, CNN)[33]、循环神经网络(Recurrent Neural Network, RNN)[34]、长短期记忆网络(Long Short Term Memory，LSTM)[35]、双向长短期记忆网络-条件随机场（Bi-directional Long Short Term Memory - Conditional Random Fields，BiLSTM-CRF）[36]等。

里耶秦简数据属于古汉语，古汉语的命名实体识别主要面临两个难点，一是分词界限不明确，与现代汉语有较大差异；二是汉语分词和命名实体识别互相影响。因此，综合上文各种方法的特点，本课题选用BiLSTM-CRF方法对里耶秦简数据进行命名实体识别处理。

里耶秦简数据命名实体识别主要分为以下三个步骤：

（1）数据预处理

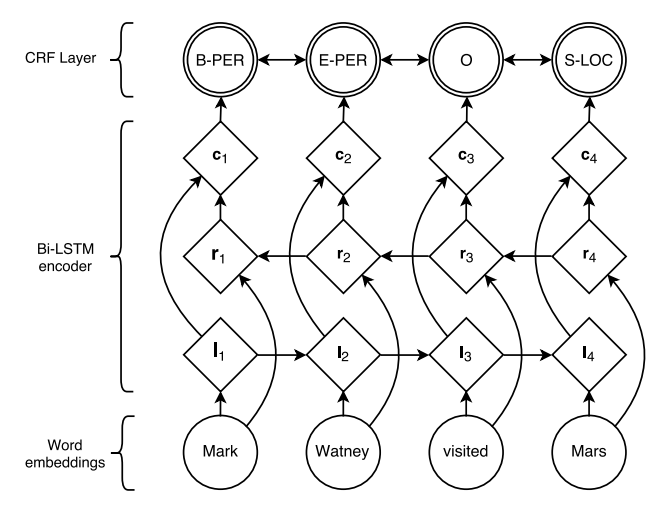
首先，将语料中的“Ⅰ”和“Ⅱ”等字符所对应的内容合并，与简号一一对应；其次，处理简号中的“+”、“、”、“至”等表示并列的字符，拆分、复制简帛文本内容；再次，处理语料中以图片表示的生僻字，手动检索所有图片表示的字并用十六进制编码；然后解决句子中的注释编号与简号混淆的问题，查找并删除在原文中的注释数字；最后，处理身份中的“/”字符，拆分、复制其内容。

（2）实体标注

以简号为索引，按照《〈里耶秦簡（貳）〉所見人名統計表》中每个简号对应的人名和身份在《里耶秦簡分類舉例》《稟食記錄簡》《里耶秦簡補充語料》中逐字标注。采用BIO标注模式，总五种实体标签。人名用B-PER、I-PER表示，身份用B-PRO、I-PRO表示。其中，B表示人名或身份首字，I表示人名或身份的后续所有字。对于既不是人名、又不是身份的字或标点符号，用O标注。

（3）使用BiLSTM-CRF模型进行命名实体识别

BiLSTM-CRF模型结构如图3-3[32]所示，第一层是Embedding层，利用预训练或随机初始化的embedding矩阵将语料句子中每个字由one-hot向量映射为低维稠密的字向量；第二层是BiLSTM层，从输入的句子的embedding序列中自动提取特征，输出维度为标签个数的分值向量；第三层是CRF层，输出得分最高的标签序列，判断出每个字的实体标签。

图3-3 BiLSTM-CRF模型结构

实体标注结果示例如图3-4所示。

图3-4 实体标注结果示例

## 3.3 关系抽取

信息抽取（Information Extraction, IE）旨在从大规模非结构或半结构的自然语言文本中抽取结构化信息。关系抽取（Relation Extraction, RE）是其中的重要子任务之一，主要目的是从文本中识别实体并抽取实体之间的语义关系。

目前，常用的关系抽取方法有五类，分别是基于模式匹配的关系抽取、基于词典驱动的关系抽取、基于机器学习的关系抽取、基于Ontology的关系抽取以及混合抽取方法[37]。基于模式匹配和词典驱动的方法依靠人工制定规则，耗时耗力，而且可移植性较差，基于本体的方法构造比较复杂，理论尚不成熟。基于机器学习的方法以自然语言处理技术为基础，结合统计语言模型进行关系抽取，方法相对简单，并具有不错的性能，成为当下关系抽取的主流方法。基于机器学习的关系抽取方法，又可以根据训练数据的标记程度分为有监督的关系抽取、半监督的关系抽取和无监督的关系抽取三类。

有监督的关系抽取（Supervised Relation Extraction），将关系抽取任务当做分类问题，处理的基本单位是包含特定实体对的句子，每一个句子都有类别标注，根据训练数据设计有效的特征，从而学习各种分类模型，然后使用训练好的分类器预测关系。这类方法能够有效利用样本的标记信息，准确率和召回率都比较高，但是需要大量的人工标记训练语料，费时费力。有监督的关系抽取方法包括借助最大熵、SVM、CRF等分类器的基于特征的方法[38-40]、基于核函数的方法、基于概率图模型的方法等。

半监督的关系抽取（Semi-Supervised Relation Extraction），以句子作为训练数据的基本单位，只有部分是有类别标注的。对于要抽取的关系首先人工设定若干种子实例，然后迭代地从数据从抽取关系对应的关系模板和更多的实例。这类方法让学习器不依赖外界交互，自动地利用未标记样本来提升学习性能。半监督的关系抽取方法包括使用种子和模式匹配的Bootstrapping方法[41]、基于图的标签传播方法[42]等。

无监督的关系抽取（Unsupervised Relation Extraction），利用有相同语义关系的实体对进行关系抽取，假设拥有相同语义关系的实体对拥有相似的上下文信息，因此可以利用每个实体对对应上下文信息来代表该实体对的语义关系，选用特征包括统计信息、标点符号、关键字、指示词和方向词、位置词(如尾字)、中心词等，并对所有实体对的语义关系进行聚类。这类方法法包含实体对标记、关系聚类和关系词选择三个过程，完全不需要对训练数据进行标注，但是大多依赖于知识库和词典的建立，系统可移植性不好。无监督的关系抽取方法包括使用特征的聚类方法和非聚类方法[43]。

本课题选用有监督的关系抽取和无监督的关系抽取两种方法。

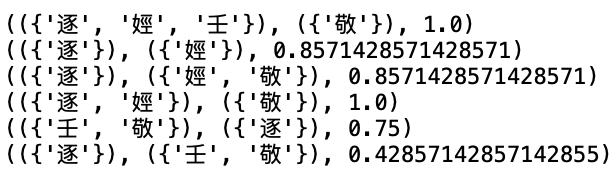
一方面，进行无监督的关系抽取，使用FP-Growth（Frequent Pattern Growth）算法[44]挖掘频繁项集。首先统计每个人名出现的总次数，找出频繁项实体集合L，并按照支持度计数降序排列；其次基于频繁项L，遍历人名实体，构造与频繁项关联的FP-Tree（Frequent Pattern Tree）；然后计算二者出现的条件概率，在FP-Tree上递归寻找频繁项集；最后推算出具有强关联的规则关系的二元组。无监督的关系抽取结果示例如图3-5所示。

图3-5 无监督的关系抽取结果示例

另一方面，进行有监督的关系抽取。第一步，根据已知的关系对命名实体识别中已有的官职进行两两间关系标注；第二步，将实体、关系和语句转化为词向量；第三步，运用BiGRU-Attention模型（Bidirectional Gated Recurrent Unity - Attention，双向门控循环-注意力机制）[45]提取两两标注实体间关系。

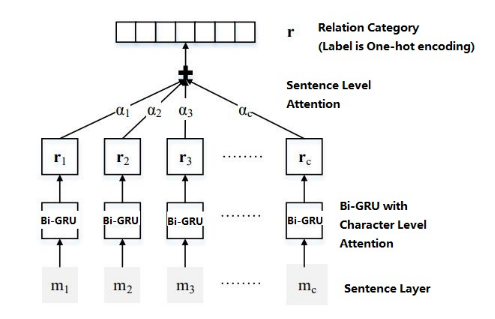
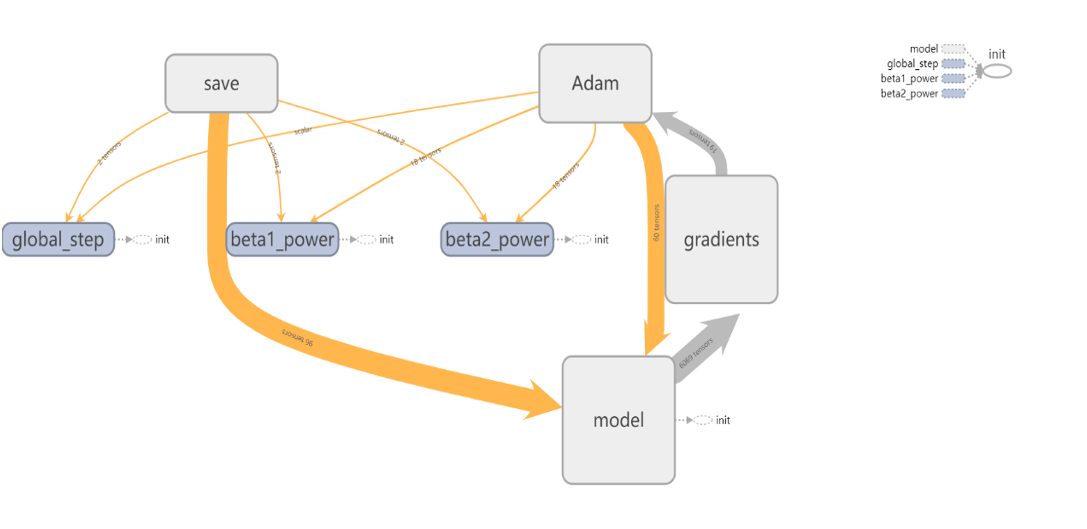
BiGRU-Attention模型结构如图3-6[46]所示。先输入句子，使用Embedding层和BiGRU层将每个词变换到低维向量中并提取高层特征；再通过Attention层创造权值向量，通过各个时间结点权值向量与字向量相乘再相加获得语句级特征；最后将语句级特征用于关系分类，输出关系分类。

图3-6 BiGRU-Attention模型结构

BiGRU-Attention模型训练过程如图3-7所示，词向量输入维数：585\*100，训练总轮数：300，每次训练样本数：50；训练结果如图3-8所示。

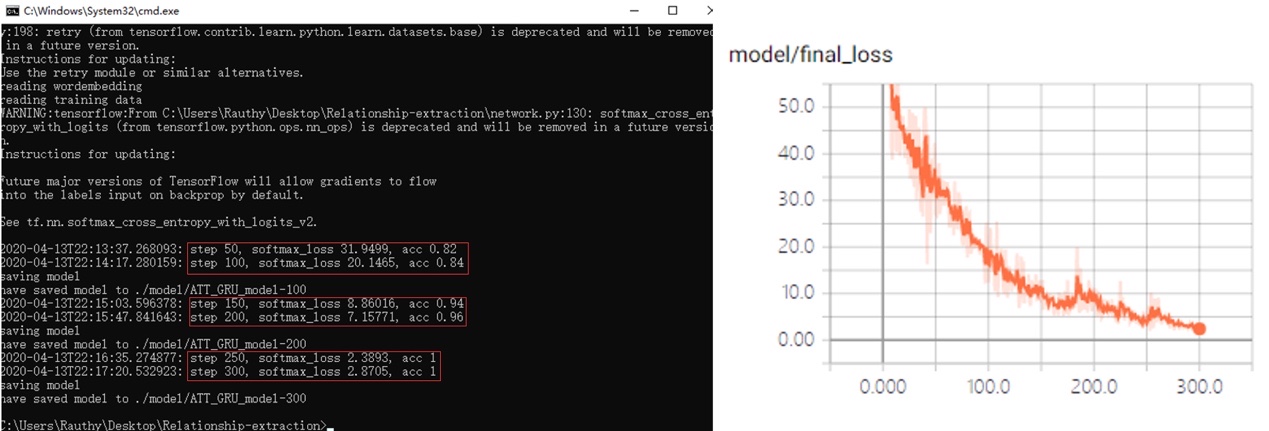
图3-7 BiGRU-Attention模型训练过程

图3-8 BiGRU-Attention模型训练结果

## 3.4 命名实体消歧

命名实体消歧（Named Entity Disambiguation，NED），也称语义消歧，是专门用于解决同名实体产生歧义问题的技术。命名实体的歧义现象可以概括为多样性和歧义性两类，即多名现象（Synonymy）和重名现象（Polysemy）。多名现象指一个实体可能会有多种命名，如全名、缩写、别名等。重名现象指多个实体拥有相同的命名，即一词多义现象[47]。

命名实体消歧方法主要按照目标列表是否给定分为基于聚类的命名实体消歧方法和基于实体链接的命名实体消歧方法。

基于聚类的命名实体消歧方法不给定目标列表，通过选取特征对指称项进行表示，比较各个实体的相似程度，将相似度高的聚集到一起。常用的方法有基于词袋模型的聚类方法[48]、基于语义特征的聚类方法[49]、基于社会化网络的聚类方法[50]、基于百科知识的聚类方法[51]、基于多源异构语义知识融合的聚类方法[52]等。

基于实体链接的命名实体消歧方法给定目标列表，其目标是将给定实体指称项链接到目标知识库中的相应实体上，主要分为候选实体的生成和候选实体的链接两步。实体链接又分为基于知识库的实体链接和基于知识图谱的实体链接[53]。

在里耶秦简数据中，命名实体的歧义现象主要是重名现象，有许多人名在不同的简上出现了多次，但是由于古汉语的特殊性，很难判断这些人名是否指代的是同一个人物。由于本课题的目标是构建里耶秦简知识图谱，并且经过命名实体识别和关系抽取工作，目前已经获得了顶点数据和边数据，因此，本课题选择了基于社会化网络的聚类的命名实体消歧方法，借助SimRank和SimRank++算法，计算出各个人名顶点之间的相似度，再进行进一步的人物分组聚类。

SimRank算法由MIT实验室的Glen Jeh和Jennifer Widom教授在2002年提出，它是一种基于图的拓扑结构信息来衡量任意两个对象间相似程度的模型，其核心思想为：如果两个对象和被其相似的对象所引用（即它们有相似的入邻边结构），那么这两个对象也相似[54]。

SimRank算法基于以下递归思想定义两个结点的相似度：如果指向结点和指向结点的结点相似，那么认为结点和结点也是相似的。这个递归定义的初始条件是：每个结点与它自身最相似。

如果用记号表示所有指向结点的结点集合（即入邻点集合），用表示两个对象间的SimRank相似度，那么SimRank的数学定义式可以以式（3-1）表示：

（3-1）

其中， 是阻尼系数常数，通常取0.6至0.8。

近年来，SimRank算法已引起广泛关注，应用于网页排名、协同过滤、孤立点检测、网络图聚类、近似查询处理等领域。但是，SimRank算法也存在以下两点问题：

（1）结点的连接结点集合越大，则与这个结点相似的结点相似度会越小。

（2）没有考虑边的差异，即没有考虑边的权重。

为了解决这些问题，Antonellis等人在2008年提出了SimRank++算法，在SimRank的基础上考虑了结点相似度证据并加入了边的权重[55]，数学定义式可以以式（3-2）表示：

（3-2）

其中， 为补偿项， 是图中每条边的权重。

本课题先借助SimRank和SimRank++算法，计算出各个人名顶点之间的相似度，筛选出每个顶点的相似度前五百的结果并去除无关顶点；然后对数据进行统计分析，分别以每组结果的相似度的均值、上四分位数、中位数和下四分位数为度量指标，对结果进行筛选，并匹配其他信息进行聚类分析。综合考虑两种算法的分组合理性和四项度量指标的筛选后数量效果，最终选用SimRank++算法和均值聚类结果。由于古汉语的特殊性，分组结果不具有百分之百的准确性，需要结合其他上下文信息进一步判断才能确定不同的人名顶点是否为同一人，课题选择不合并人名顶点，而是将分组结果作为人名顶点属性添加。至此，里耶秦简知识图谱数据准备完成。

## 3.5 本章小结

本章主要介绍了里耶秦简知识图谱的构建。首先介绍了实验所用的里耶秦简数据的来源、基本情况和示例，提出了知识图谱构建的研究框架；其次介绍了命名实体识别的基本概念和当前命名实体识别的主要技术方法，针对课题所用的里耶秦简数据的特点，选取了合适的处理方法，进行了数据预处理、实体标注并使用BiLSTM-CRF模型进行命名实体识别；再次简述了关系抽取的基本概念及其主要方法，说明了本课题使用了FP-Growth算法进行无监督的关系抽取和运用BiGRU-Attention模型进行有监督的关系抽取；最后介绍了命名实体消歧的基本概念、命名实体消歧的两大类方法，以及SimRank和SimRank++算法，课题使用SimRank和SimRank++算法与统计分析聚类，实现了人物分组，里耶秦简知识图谱数据准备完成。

# 4 图数据查询分析系统设计与实现

本章详细说明了图数据查询分析系统具体的设计与实现过程，包括系统需求分析、系统整体框架设计、图数据可视化展示模块、顶点与边的查询模块、图数据算法调用模块和图数据上传模块等。

## 4.1 系统需求分析

近年来，随着社交网络、零售电商、金融风控、生命科学等行业的快速发展，需要处理的数据之间的关系随数据量呈几何级数增长，并且，越来越多的数据以庞大而复杂的关系网的形式出现，不得不以图数据（Graph Data）的形式进行保存和处理。由于图的复杂性，与其他数据结构相比，理解图的难度显著提高。因此，亟需一个图可视化系统，帮助用户直观、快速、灵活地查询和分析图数据，从而发现其中蕴含的价值。

不管是国内还是国外，图数据的可视化查询分析都有着广泛的应用需求。目前，图数据可视化查询分析平台层出不穷、功能日渐强大，但也存在一定的局限。这些平台大部分偏向于工业应用，对学术方面的支持较少，部分平台局仅针对特定领域，使用门槛较高。

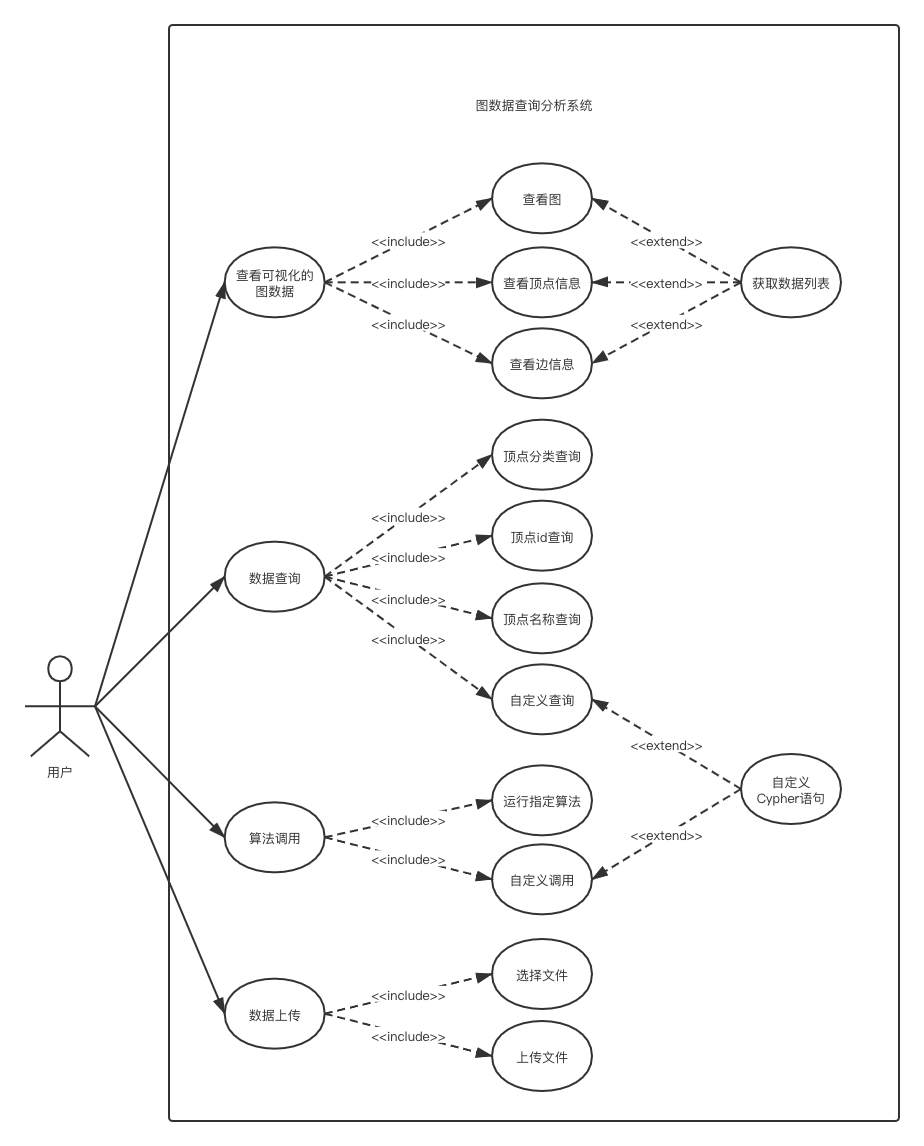
本课题的目标是构建一个通用的图数据查询分析系统，实现图数据的可视化，支持用户的自定义查询，同时加入图数据相关算法，满足用户的深度分析需求。综合考虑当前市场上的主流图数据库的特点，本课题将使用Neo4j图数据库。

根据课题目标，确定了如表4-1所示的系统总体功能需求。

表4-1 系统总体功能需求

|  |  |  |
| --- | --- | --- |
| 功能序号 | 功能名称 | 功能描述 |
| 1 | 可视化展示 | 实现图数据的可视化展示 |
| 2 | 数据查询 | 根据用户输入的信息查询相关图数据 |
| 3 | 算法调用 | 由用户选择算法、设置参数，并在图上运行 |
| 4 | 数据上传 | 用户上传自定义的图数据 |

根据课题目标和系统总体功能需求，确定了系统用例，如图4-1所示。

图4-1 系统用例图

## 4.2 系统详细设计

本课题的目标是构建一个通用的图数据查询分析系统，实现图数据的可视化，支持用户的自定义查询，同时加入图数据相关算法，满足用户的深度分析需求。综合课题目标和系统总体功能需求考虑，系统将分为图数据可视化展示、顶点与边的查询、图数据算法调用和图数据上传四个模块。

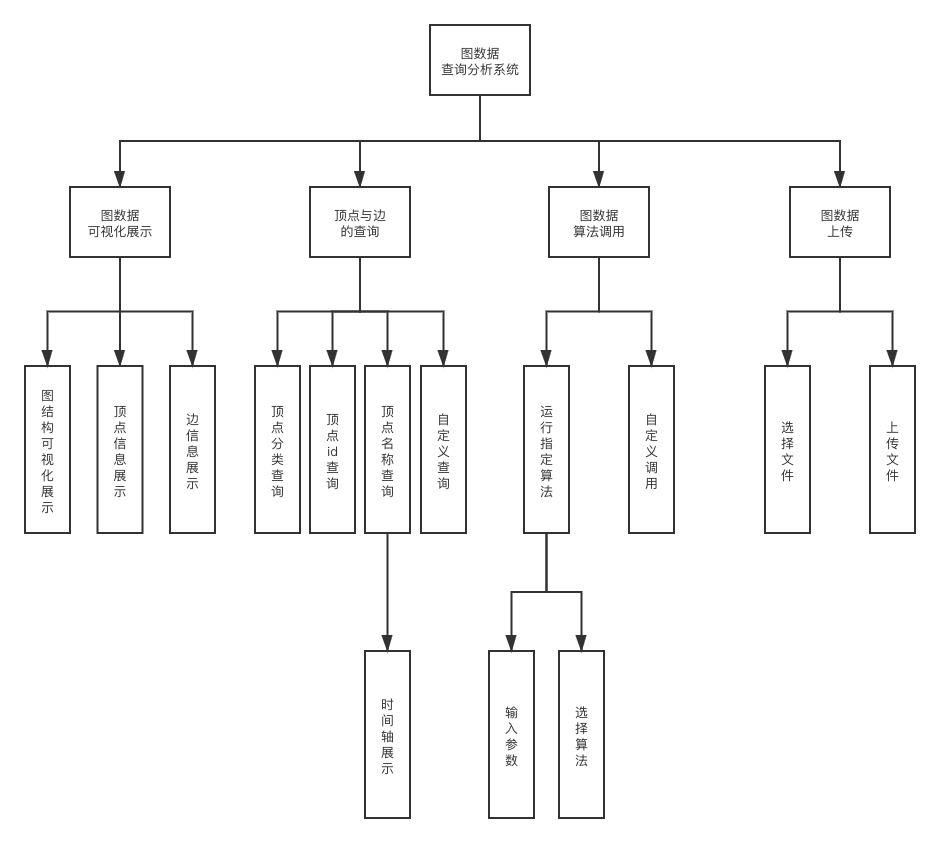
系统总体功能模块如图4-2所示。

图4-2 系统总体功能模块图

系统具体功能如表4-2所示。

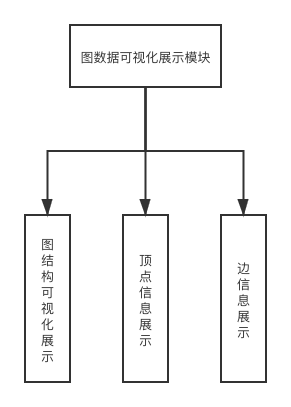
表4-2 系统具体功能

|  |  |  |
| --- | --- | --- |
| 功能序号 | 功能名称 | 功能描述 |
| 1.1 | 图结构可视化展示 | 实现图结构的可视化展示，包括拖动、缩放、点击等效果 |
| 1.2 | 顶点信息展示 | 显示当前图中所有顶点的具体信息 |
| 1.3 | 边信息展示 | 显示当前图中所有边的具体信息 |
| 2.1 | 顶点分类查询 | 在图中显示指定类别的顶点 |
| 2.2 | 顶点id查询 | 根据顶点id查询指定顶点及其相关边 |
|  |  |  |
| 续表 |  |  |
| 2.3 | 顶点名称查询 | 根据顶点名称查询指定顶点及其相关边 |
| 2.3.1 | 时间轴展示 | 根据当前图中顶点的时间信息生成时间轴 |
| 2.4 | 自定义查询 | 自定义顶点和边的查询 |
| 3.1 | 运行指定算法 | 在图上运行指定算法 |
| 3.1.1 | 输入参数 | 输入算法运行所需参数 |
| 3.1.2 | 选择算法 | 选择要在图上运行的算法 |
| 3.2 | 自定义调用 | 自定义要调用的算法及参数 |
| 4.1 | 选择文件 | 选取要上传的图数据文件 |
| 4.2 | 上传文件 | 上传已选择的图数据文件 |

## 4.3 图数据可视化展示模块

图数据可视化展示模块是实现图数据查询分析系统的第一步，系统对图数据的查询与分析都是建立在可视化的图数据的基础之上。图数据可视化展示模块负责完成图结构的可视化展示，同时对点击指定顶点或边时做出响应，显示其具体信息，并实现显示当前图中所有顶点的具体信息和显示当前图中所有边的具体信息功能。

图数据可视化展示模块主要包括图结构可视化展示、顶点信息展示和边信息展示三个部分。图数据可视化展示模块结构如图4-3所示。

图4-3 图数据可视化展示模块结构

### 4.3.1 图结构可视化展示

图结构可视化展示部分是图数据可视化展示模块的核心组成部分，也是整个系统的基础，它决定了用户是否能够轻松地理解系统所呈现的图数据。

随着近年来图数据的兴起，很多针对图结构的可视化的工具和方法也随之出现，例如Neovis.js、D3.js、Vis.js、Sigma.js等JavaScript库和GraphXR、yFiles、Linkurious Enterprise等独立的图数据可视化平台。本课题构建的图数据查询分析系统，需要在系统的前端实现图结构的可视化，因此，需要借助JavaScript库工具实现。

在JavaScript库图数据可视化工具之中，最著名、最流行的莫过于D3.js（Data-Driven Documents）。D3.js将强大的可视化和交互技术与数据驱动的DOM操作方法结合在一起，使得用户能够自由地为其数据设计合适的可视化界面。D3.js功能非常强大，并且异常灵活，能够满足各种个性化的需求，但是它的实现也较为复杂，而且必须将数据转换为D3.js所用的格式。

另一类JavaScript库图数据可视化工具则以Neovis.js为代表。Neovis.js是基于Vis.js和Neo4j数据模型构建的JavaScript可视化工具。由于Neovis.js是在Neo4j的属性图模型的基础上构建的，所以Neovis.js实现了将JavaScript可视化和Neo4j数据库无缝集成在一起。它与Neo4j的连接非常简洁，所用的数据格式与Neo4j数据库保持一致，并且能够轻易设置图结构可视化的标签、特性、结点、关系等相关细节。

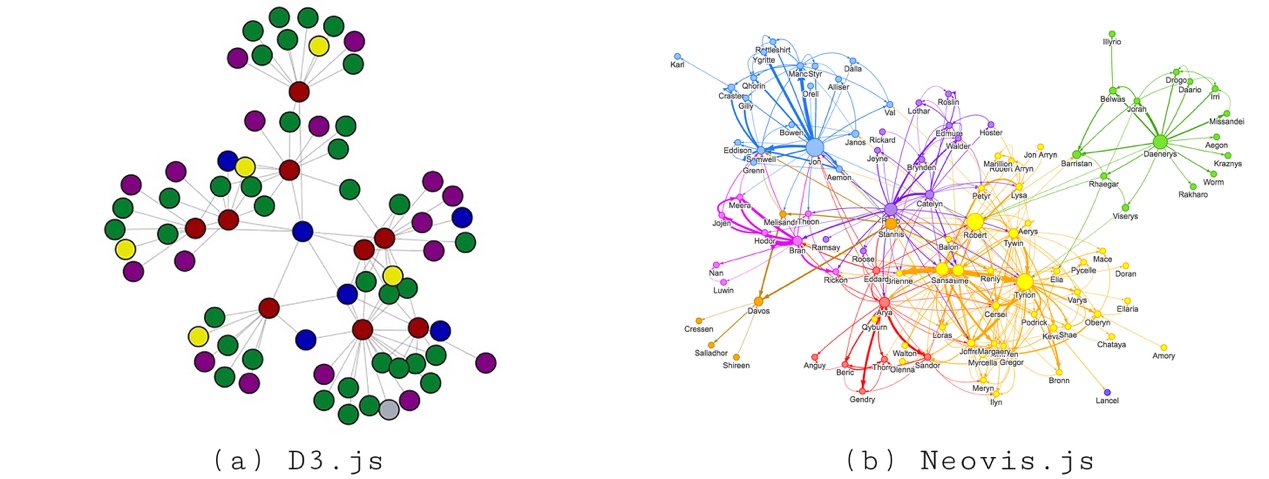
D3.js和Neovis.js的图结构可视化效果图如图4-4[56]所示。

图4-4 D3.js和Neovis.js的图结构可视化效果图

本系统选用Neo4j图数据库存储图数据，对图数据的可视化需求主要在于展示顶点和边的结构关系以及显示顶点和边的属性信息，并无额外附加的特殊需求。综合以上信息考虑，课题选用Neovis.js实现图结构可视化展示的功能。

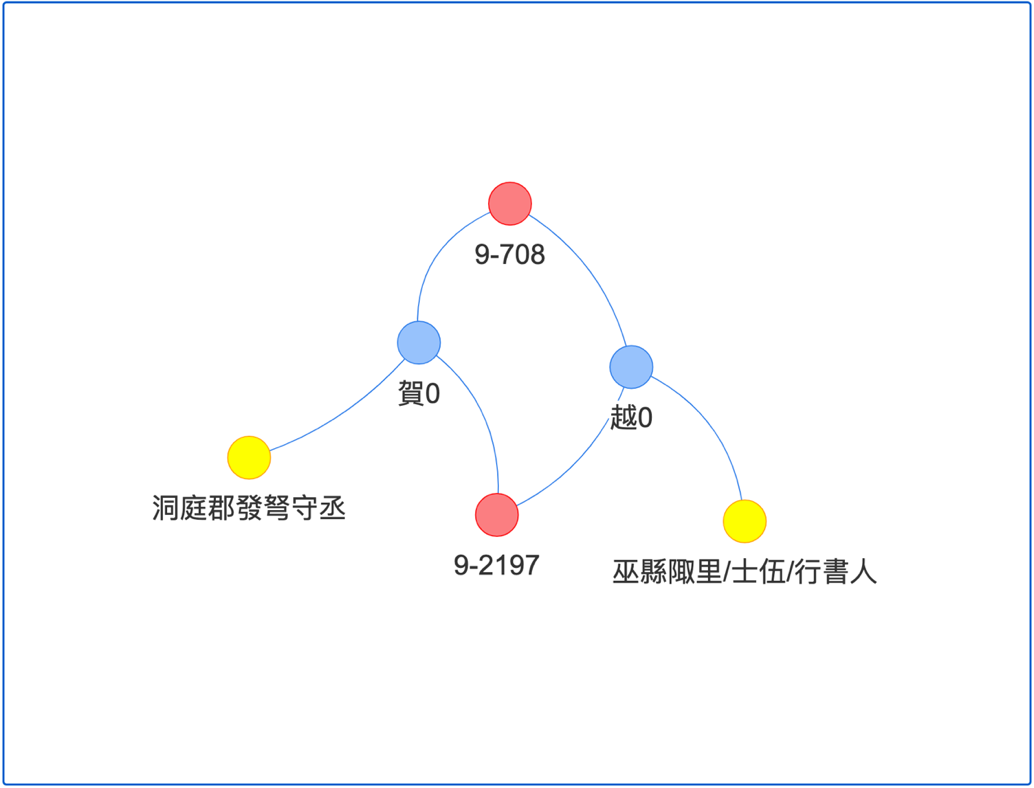
借助Neovis.js实现图结构可视化展示的过程也很简洁：在系统项目中添加Neovis.js库，创建绘制图像的draw函数，并将draw函数设置为随页面加载完成后立即运行，即可完成图结构可视化展示。在draw函数中，首先定义config对象，配置连接Neo4j数据库所需的信息、图中顶点和边的显示规则和默认显示设置；然后根据config创建Neovis实例viz，连接Neo4j数据库并获取数据；最后渲染Neovis实例viz，实现图结构可视化展示，包括拖动、缩放、点击等效果。图结构可视化展示示例如图4-5所示。

图4-5 图结构可视化展示示例

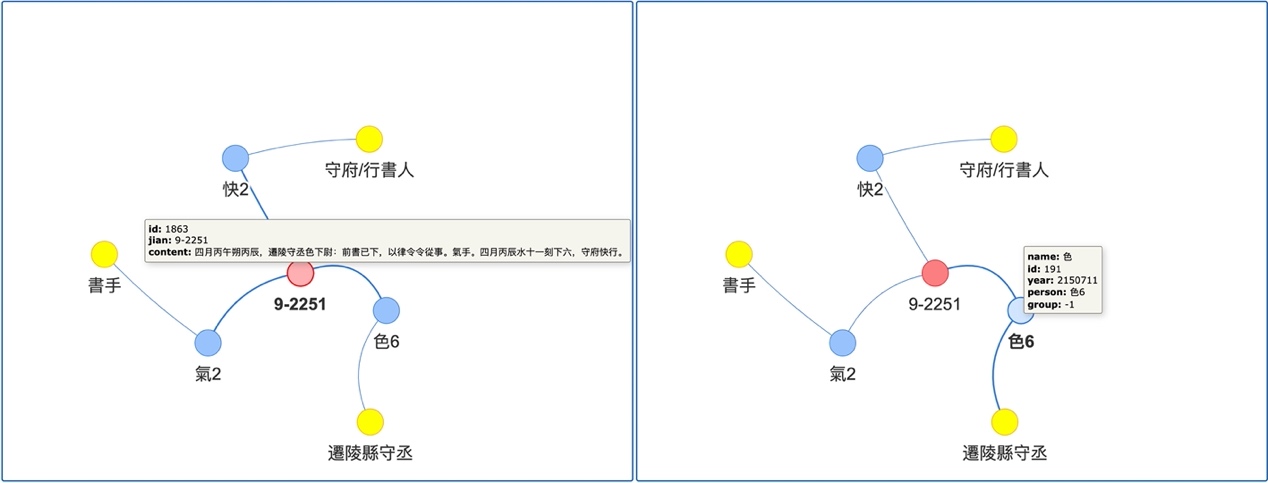
### 4.3.2 顶点信息展示

顶点信息展示是图数据可视化展示模块的重要组成部分，它能够呈现图数据的细节信息，从而帮助用户深入理解数据内容、进一步地进行数据分析工作。

顶点信息展示包括两个部分，一是显示当前图中某一个顶点的具体信息，二是显示当前图中所有顶点的具体信息。

（1）显示当前图中某一个顶点的具体信息

显示当前图中某一个顶点的具体信息通过点击顶点实现。在图结构可视化展示部分完成图的显示后，可以对图进行拖动和缩放。对于想查看具体信息的顶点，可以在图上进行点击操作，点击后，被点击的顶点会改变图标颜色、加粗顶点名称文本，同时图上会出现悬浮文本框，显示该顶点的具体信息。具体效果如图4-6所示。

图4-6 显示当前图中某一个顶点的具体信息

（2）显示当前图中所有顶点的具体信息

实现显示当前图中所有顶点的具体信息功能，需要先获取相应的数据，再对数据进行格式化处理，最后实现显示。

系统使用NeoVis实例viz连接Neo4j数据库，获取的顶点数据保存在viz.\_nodes对象数组中。viz.\_nodes中的对象包含id、title等属性，id为Neo4j数据库赋予顶点的id，title字符串则是顶点的具体信息。title字符串也会用于实现图4-6中的悬浮文本框，它包含了属性名称、属性内容和HTML标签三类信息，其中，属性名称和属性内容与Neo4j数据库中的数据一致。因此，需要对title字符串进行格式化处理，筛选出需要的信息。借助正则表达式，去除HTML标签，提取属性名称和属性内容，一一对应保存，再进行显示。

由于界面显示的图一旦更新就需要重新获取顶点信息，如果每一次更新都重新格式化数据、创建新对象保存数据，那么不可避免地会产生很多额外的时间和空间开销，影响系统性能，所以，课题选用Vue框架下的Vuex工具管理数据。

Vue是一套用于构建用户界面的渐进式JavaScript框架，可以自底向上逐层应用，使用组件实现代码的模块化。Vue 易用、灵活、高效，并且拥有很多与之配套的工具，Vuex就是其中之一。

Vuex是一个专为Vue.js应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。当多个组件共享状态时，使用Vuex可以将组件的共享状态抽取出来，以一个全局单例模式管理，便于各个组件获取状态或者触发行为[57]。

借助Vuex实现显示当前图中所有顶点的具体信息的过程如下：

① 根据Neo4j数据库中的属性名称，为系统预置的里耶简帛数据、DBLP数据和用户上传的自定义数据创建对应的数据类。

② 创建getInfo函数，传入id和title字符串，使用正则表达式去除HTML标签和其他无关信息，提取属性名称和属性内容，保存为info数组并输出。

③ 在Vuex中，创建nodeList数组，用于保存结果；创建布尔值isNewNodes标记当前图中的顶点信息是否提取过，默认值为false；创建Getter方法getIsNewNodes、getNodeList和Mutation方法setIsNewNodes、setNodeList，用于在组件中获取顶点信息和触发函数调用。

④ 组件调用getNodeList从Vuex获取顶点信息：先调用getIsNewNodes获取isNewNodes；若isNewNodes为false，则返回当前nodeList数组；若isNewNodes为true，则调用setNodeList，遍历viz.\_nodes，通过getInfo函数从title中获取info数组，再使用info初始化当前图对应的数据类对象，存入nodeList，遍历结束返回nodeList数组。

⑤ 组件获取到顶点信息后，调用setIsNewNodes将isNewNodes设置为false，并完成顶点具体信息的显示。

⑥ 界面显示的图每次更新组件都将调用setIsNewNodes将isNewNodes设置为true。

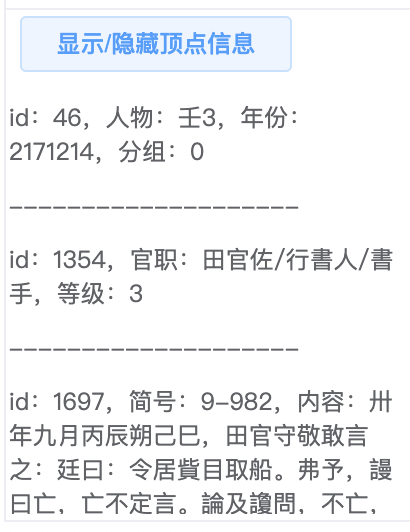
在系统界面添加“显示/隐藏顶点信息”按钮，完成显示当前图中所有顶点的具体信息功能，如图4-7所示。

图4-7 显示当前图中所有顶点的具体信息

### 4.3.3 边信息展示

边信息展示同样是图数据可视化展示模块的重要组成部分，它能够呈现顶点间关系的细节信息，从而帮助用户深入理解数据的结构、进一步地进行数据分析工作。

边信息展示也包括两个部分，一是显示当前图中某一条边的具体信息，二是显示当前图中所有边的具体信息。

（1）显示当前图中某一条边的具体信息

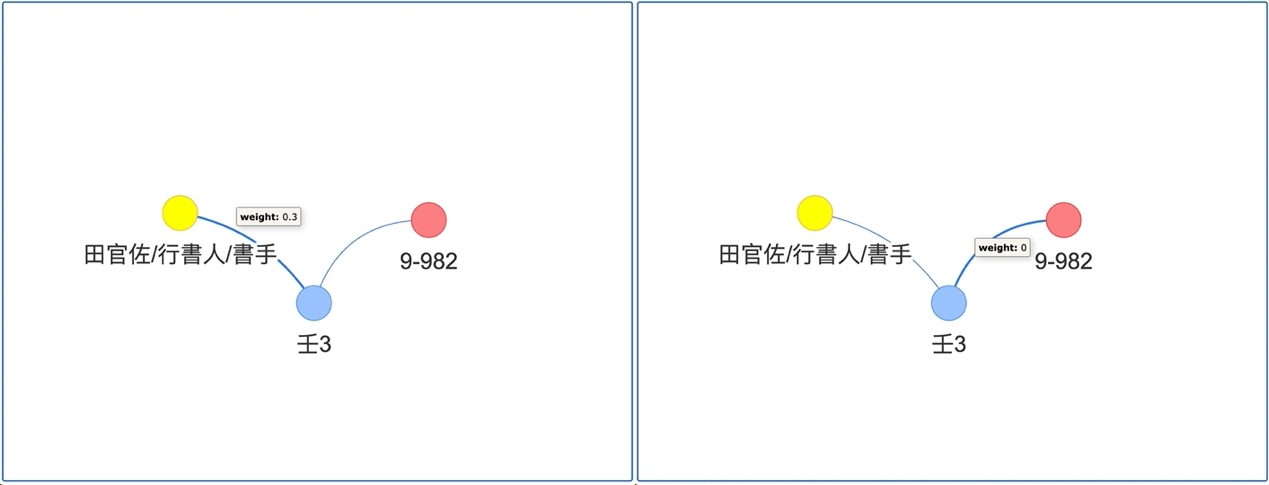
显示当前图中某一条边的具体信息通过点击边实现。在图结构可视化展示部分完成图的显示后，可以对图进行拖动和缩放。对于想查看具体信息的边，可以在图上进行点击操作，点击后，被点击的边会加粗，同时图上会出现悬浮文本框，显示边的具体信息，通常为边的权重信息。具体效果如图4-8所示。

图4-8 显示当前图中某一条边的具体信息

（2）显示当前图中所有边的具体信息

与实现显示当前图中所有顶点的具体信息功能相同，系统将借助Vuex实现实现显示当前图中所有边的具体信息功能。不过，与之不同的是，要实现显示当前图中所有边的具体信息除了需要获取边的数据，还必须先获取顶点数据，否则将因无法匹配边两端顶点的信息而只能显示边本身的信息，只显示边本身的信息显然不利于用户理解、分析数据。

系统使用NeoVis实例viz连接Neo4j数据库，获取的边数据保存在viz.\_edges对象数组中。viz.\_ edges中的对象包含id、from、to、title等属性，id为Neo4j数据库赋予边的id，from为边的起始顶点id，to为边的终止顶点id，title则是边的具体信息。

借助Vuex实现显示当前图中所有边的具体信息的过程如下：

① 在Vuex中，创建edgeList数组，用于保存结果；创建布尔值isNewEdges标记当前图中的边信息是否提取过，默认值为false；创建Getter方法getIsNewEdges、getEdgeList和Mutation方法setIsNewEdges、setEdgeList，用于在组件中获取边信息和触发函数调用。

② 组件调用getEdgeList从Vuex获取顶点信息：先调用getIsNewEdges获取isNewEdges；若isNewEdges为false，则直接返回当前edgeList；若isNewEdges为true，则调用getIsNewNodes获取isNewNodes，若isNewNodes为false，则调用setEdgeList，遍历viz.\_edges，获取from、to和title，使用from和to匹配nodeList中的顶点信息，使用正则表达式去除title中的HTML标签和其他无关信息，将结果保存为info数组存入edgeList，遍历结束返回edgeList数组；若isNewNodes也为true，则先执行本文4.3.2节中获取顶点数据的步骤，再调用setEdgeList。

③ 组件获取到边信息后，调用setIsNewEdges将isNewEdges设置为false，并完成顶点具体信息的显示。

④ 界面显示的图每次更新组件都将调用setIsNewEdges将isNewEdges设置为true。

在系统界面添加“显示/隐藏边信息”按钮，完成显示当前图中所有边的具体信息功能，如图4-9所示。

图4-9 显示当前图中所有顶点的具体信息

## 4.4 顶点与边的查询模块

顶点与边的查询模块是图数据查询分析系统的核心组成部分，构建系统的一大目标就是实现对图的自由查询。顶点与边的查询模块负责接收用户发出的各种查询命令并返回其结果。

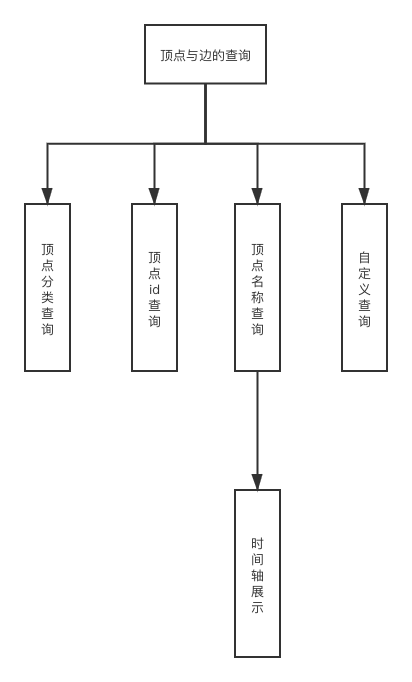
顶点与边的查询模块主要包括顶点分类查询、顶点id查询、顶点名称查询和自定义查询四个部分，顶点名称查询部分还附加了时间轴展示功能。顶点与边的查询模块结构如图4-10所示。

图4-10 顶点与边的查询模块结构

### 4.4.1 顶点分类查询

图数据中经常具有多个类型的顶点，用户也常常需要筛选出某一类型的顶点、对其进行操作。因此，有必要在顶点与边的查询模块中加入顶点分类查询功能。

Neo4j数据库使用声明式查询语言Cypher，系统使用的Neovis.js与Neo4j数据库无缝集成，其实例初始化时即可传入cypher参数，从而实现图的初次加载。需要在图中查询其他内容时，可以使用renderWithCypher函数，传入新的cypher参数即新的查询语句，即可重新查询、显示新的结果。

但是，Cypher语言属于专业语言，使用门槛较高。为了便于用户使用，系统选择将常用的查询语句封装到操作按钮中，用户只需要输入相应参数，就能实现查询操作。

如图4-11所示，系统将顶点分类查询封装到按钮中。对于系统预置的数据，里耶简帛数据具有三类顶点，分别设置“显示人名”、“显示官职”、“显示简”和能够显示图中所有顶点的“显示图”按钮；DBLP数据具有两类顶点，分别设置“显示论文”、“显示作者”和“显示图”按钮。对于用户上传的自定义数据，最多支持三类顶点，设置三个类型查询按钮和“显示图”按钮。

图4-11 顶点分类查询按钮

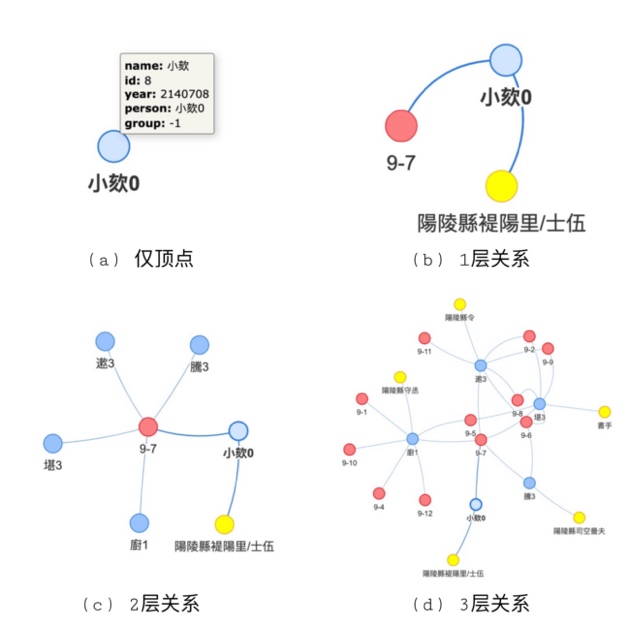
### 4.4.2 顶点id查询

图数据中，每个顶点都有其专属的id，顶点id唯一且不重复，作为顶点的标识，研究图数据时常常使用id指代顶点。因此，顶点与边的查询模块必须具有顶点id查询功能。

系统将顶点id查询也封装到按钮中，用户只需要输入要查询的顶点id，即可获取相应的结果。

我们在研究图数据中的顶点时，除了顶点本身，往往也要考虑它的邻居顶点，查看它在图中的结构关系。因此，对于顶点id查询，系统也将被查询顶点的邻居顶点以及它们之间的关系包含在内，支持查询指定顶点及其一层关系、二层关系和三层关系。

顶点id查询按钮图如4-12所示。以id为8的顶点为例，各个按钮的查询效果示例如图4-13所示。

图4-12 顶点id查询按钮 图4-13 顶点id查询按钮查询示例

### 4.4.3 顶点名称查询

在研究图数据时，除了顶点id，也常常使用顶点的某一个重要属性指代顶点，本文将其称为顶点名称。例如，里耶秦简数据中人名顶点的人名属性，DBLP数据中论文顶点的标题属性。因此，顶点名称查询是顶点与边的查询模块不可或缺的功能。

顶点名称查询也被封装到按钮中，支持查询指定顶点及其一层关系和二层关系。同时，顶点名称查询支持模糊查询，只输入部分信息也能获取相应的查询结果。系统预置里耶秦简数据和DBLP数据，对这两种数据，根据它们的特点，设置了不同的顶点名称查询模式。

本文在第三章构建了里耶秦简知识图谱，里耶秦简知识图谱的核心是人名顶点，其中的一个重要工作就是命名实体消歧，对里耶秦简中的人名顶点进行了判断分组，并将分组结果作为人名顶点属性添加。因此，对于里耶秦简数据的顶点名称查询，系统使用了人名查询和人物分组查询模式，如图4-14(a)所示。原始的里耶秦简中，具有很多同名人物，因此，人名顶点使用人名加序号作为人名属性，以便区分。使用人名查询，可以获取所有人名为输入信息的人名顶点及相应的关系，例如，查询“卬”，将返回“卬0”、“卬1”、“卬2”和“卬3”及相应的关系。使用人物分组查询，可以获取该分组的所有人名顶点及相应的关系。

图 4-14顶点名称查询按钮

DBLP数据包含论文和作者两类顶点，对于DBLP数据的顶点名称查询，系统系统使用了论文查询和作者查询模式，如图4-14(b)所示。使用论文查询，可以获取所有标题包含输入的关键词的论文及相应的关系。使用作者查询，可以获取所有姓名包含输入的关键词的作者及相应的关系。

系统预置的里耶秦简数据和DBLP数据，其顶点除了具有图数据中常见的id、名称等属性外，还共同拥有一个重要属性——时间。在对里耶秦简的研究中，人物与官职和时间直接的关系尤为重要，通过分析人物与其担任某一官职的时间，再结合官职直接的上下级关系，可以协助进行判断不同的人名是否指代同一个人物、不同的人物之间的关系等研究工作。在DBLP数据中，论文之间的时间关系揭示了其作者的研究方向变化。因此，系统针对数据的特点和对数据的分析需求，在顶点名称一层关系查询时添加了时间轴展示功能。

（1）里耶秦简人物职位变迁时间轴

里耶秦简数据具有三类顶点和三类边：人名顶点、官职顶点、简帛顶点和人名-官职边、人名-简帛边、官职-官职边。人名顶点具有时间属性，是在简帛语料记录中该人名所出现的时间；官职顶点具有等级属性，通过人工标注和关系抽取获得，表示官职的等级；简帛顶点具有内容属性，记录了该简帛的具体信息。人名-官职边表示该人名出现时的官职，人名-简帛边表示该人名所在的简帛，官职-官职边则表示官职之间的层级关系。

里耶秦简研究的一项重要内容就是人物判断。借助里耶秦简知识图谱的信息，我们可以通过人名顶点出现的时间、人名顶点出现时担任的官职和官职之间的层级关系，进行人物判断工作。通常情况下，人物的官职等级会随着时间而提升，不考虑人物降职情况。

当进行人物分组一层关系查询时，可以获得疑似同一人物的人名顶点以及与它们关联的官职顶点和简帛顶点。整合以上信息，即可获得人物职位变迁时间轴，具体步骤如下：

① 创建calcPositionChange函数，传入从Vuex中获取当前顶点信息nodeList数组和边信息edgeList数组。

② 从nodeList中提取人名顶点数组并遍历，根据edgeList找到与当前人物连接的两个顶点并选出官职顶点，然后从人名顶点和其对应的官职顶点中提取人名顶点的id、名称、时间和官职顶点的id、官职名称、等级属性，存入groupInfo数组。

③ 去除groupInfo中无时间信息的元素后，按官职等级排序，获取每个官职的起始时间和中止时间，存入positionChangeInfo数组，再按起始时间对positionChangeInfo中相同等级的官职排序。

④ 创建timeline数组，遍历positionChangeInfo，将元素存入与它的起始时间最接近且不冲突的timeline。

⑤ 选取元素最多的timeline作为人物职位变迁时间轴展示，如果timeline中只有一个元素则不展示并提示“信息不足”。

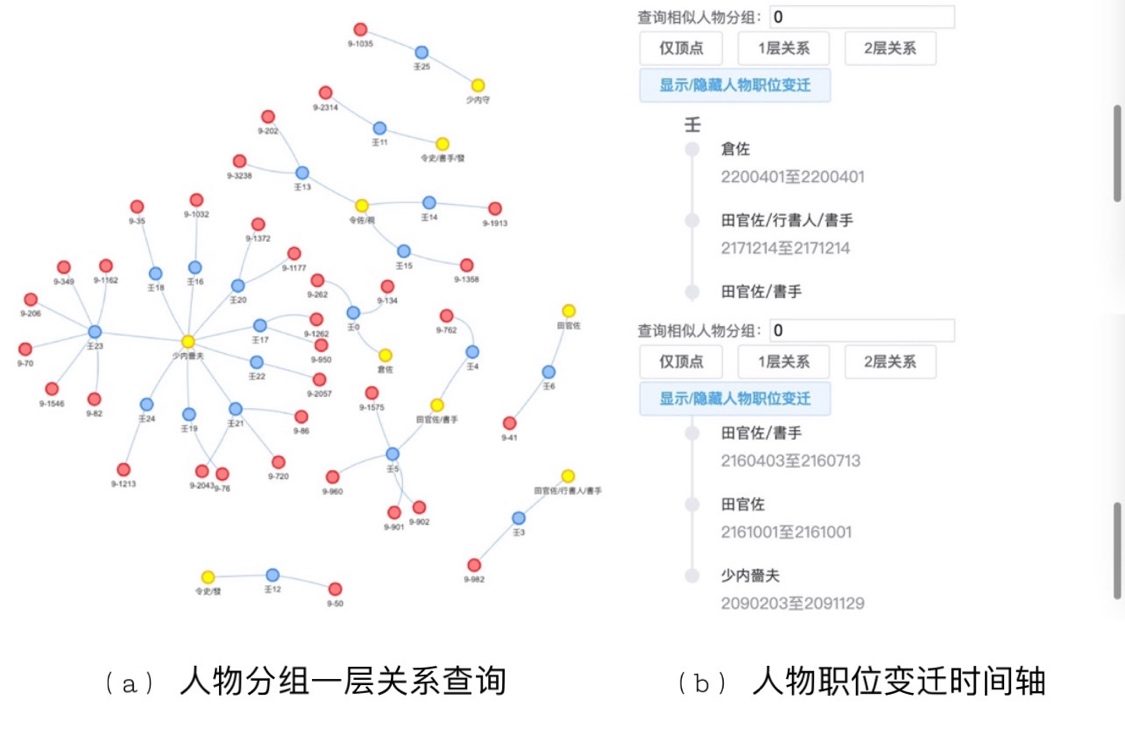
以分组0为例，人物分组一层关系查询和时间轴展示如图4-15所示。图4-15(a)为分组0的一层关系查询结果，显示了若干个疑似同一人物的人名顶点“壬”及其关联的官职顶点和简帛顶点。图4-15(b)为分组0的时间轴展示效果，使用纵向时间轴展示了人物“壬”在不同年份担任的官职，时间轴从早到晚顺序排列。特别的是，里耶秦简中所记录的年份信息均为公元前，因此，年份信息不同时，年份数值大的时间更早；年份信息相同时，月份和日期数值小的时间更早。

图4-15 人物分组一层关系查询和人物职位变迁时间轴展示

（2）DBLP作者学术生涯时间轴

DBLP数据具有两类顶点和两类边：作者顶点、论文顶点和作者-论文边、作者-作者边。作者顶点具有关键词属性，表明其研究领域；论文具有时间属性，表示论文发表的时间。作者-论文边表示论文的作者，一个论文顶点与一个或多个作者顶点相连；作者-作者边表示不同作者的合作关系。

在DBLP数据中，每一个作者都关联至少一篇论文，每一篇论文都具有时间属性，论文之间的时间关系揭示了其作者的研究方向变化。

当进行作者一层关系查询时，可以获得查询的作者顶点以及与它关联的论文顶点和合作作者顶点。对论文顶点按其时间属性进行排序，即可获得作者学术生涯时间轴，具体步骤如下：

① 创建calcAcademicCareer函数，传入从Vuex中获取当前顶点信息nodeList数组。

② 从nodeList中提取论文顶点数组，存入academicCareerInfo数组，并按照时间排序。

③ 将academicCareerInfo作为作者学术生涯时间轴展示，如果academicCareerInfo中只有一个元素则不展示并提示“信息不足”。

以作者“Tibor Kiss”为例，作者一层关系查询和时间轴展示如图4-16所示。图4-16(a)为作者“Tibor Kiss”的一层关系查询结果，显示了作者顶点“Tibor Kiss”及其与他关联的论文顶点和合作作者顶点。图4-16(b)为作者“Tibor Kiss”的时间轴展示效果，使用纵向时间轴展示了作者“Tibor Kiss”发表的论文，时间轴从新到旧逆序排列。

图4-16 作者一层关系查询和作者学术生涯时间轴展示

### 4.4.4 自定义查询

Neo4j数据库的查询语言为Cypher语言，为了降低用户使用门槛，如本文的4.4.1节、4.4.2节和4.4.3节所述，系统封装了多组查询按钮。同时，为了满足专业用户更高、更个性化的查询分析需求，系统也设置了自定义查询功能。

图4-17 自定义查询

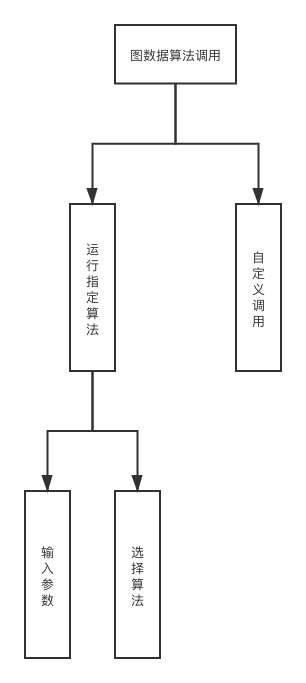
如图4-17所示，顶点查询界面的顶部设置了自定义查询模块，用户在文本框内输入自己编写的Cypher查询语句并提交后，系统将调用Neovis.js实例viz的renderWithCypher函数并将用户输入的内容作为新的cypher参数传入，进行新的查询，进而在页面显示用户定制的查询结果。

自定义查询模块最大程度地保留了系统的灵活性，使得系统能够兼顾普通用户和专业用户的使用需求。

## 4.5 图数据算法调用模块

图数据算法调用模块是图数据查询分析系统的重要功能，在系统中加入图数据相关算法能够满足用户的深度分析需求。图数据算法调用模块负责获取用户输入的参数、执行用户指定的算法并返回其结果。

图数据算法调用模块主要包括运行指定算法和自定义调用两个部分，运行指定算法部分包括输入参数和选择算法两个功能。图数据算法调用模块结构如图4-18所示。

图4-18 图数据算法调用模块结构

### 4.5.1 运行指定算法

系统支持运行的算法包括两类，一类是Neo4j数据库的图数据科学库中内置的算法，本文的2.2节已经对其进行了简单的介绍，图数据科学库可以直接执行部分基础的图数据算法，能够有效地帮助系统功能实现；另一类是课题自行准备的扩展算法，给系统提供了更多的支持。

（1）图数据科学库内置算法

考虑到课题所用数据的特点，系统选用了图数据科学库中的社区检测算法Louvain算法和中心性算法PageRank算法。

Louvain算法,又称为Fast-Unfolding算法,是基于模块度(Modularity）的社区发现算法[58]。模块度是一种评价社区划分好坏的度量标准，它的含义是社区内结点的连边数与随机情况下的边数之差。Louvain算法在大规模图结构上的性能和表现效果都比较好，并且能够发现层次性的社区结构。

PageRank是一种由搜索引擎根据网页之间相互的超链接计算的技术，作为网页排名的要素之一，Google用它来体现网页的相关性和重要性[59]。PageRank算法的核心思想十分简单：如果一个网页被很多其他网页链接到的话说明这个网页比较重要，其PageRank值会相对较高；如果一个PageRank值很高的网页链接到另一个网页，那么被链接到的网页的PageRank值会相应地因此而提高。目前很多重要的链接分析算法都是在PageRank算法基础上衍生出来的。

（2）扩展算法

在图数据科学库的内置算法以外，系统还加入了相似度算法SimRank算法和社区搜索算法CAC搜索算法两个扩展算法。

SimRank算法是一种相似度算法，借助图的拓扑结构信息来衡量任意两个对象之间的相似程度，课题的命名实体消歧工作就借助了SimRank算法，本文的3.4节已经进行了详细介绍。

CAC搜索算法是一种基于内聚属性社区（Cohesive Attributed Community）模型的属性社区搜索算法。其目的是，对于查询顶点和关键词集合，找到包含关键词集合中共享关键词最多的查询顶点的内聚连通社区。CAC搜索算法既能保证社区的结构内聚性，又能保证社区的属性内聚性[60]。

扩展算法的加入分为两个步骤：第一步，使用Java将实现好的算法封装为动态链接库和jar包，算法不一定需要用Java语言实现；第二步，将动态链接库和jar包放入Neo4j数据库指定的位置，并修改Neo4j数据库的配置文件，将算法包作为插件导入。导入完成后，即可和图数据科学库的内置算法一样，使用Cypher语言调用扩展算法。用这种方式加入算法，扩展性极高，可以很方便地添加、修改、移除算法包。

运行指定算法包括输入参数和选择算法两个功能。用户输入运行算法所需的顶点id、阻尼系数、k值等参数后，通过点击按钮选择要运行的算法。系统前端通过neo4j-driver获取Neo4j数据库的算法运行结果，并以表格的形式进行显示。

为了便于理解算法运行结果，系统对Louvain算法和CAC搜索算法这两个与社区相关的算法，添加了显示指定社区的功能。输入要查询的社区id，即可在图数据可视化展示模块显示对应的社区。

运行指定算法示例如图4-19所示。

（待补充）

图4-19 运行指定算法示例

### 4.5.2 自定义调用

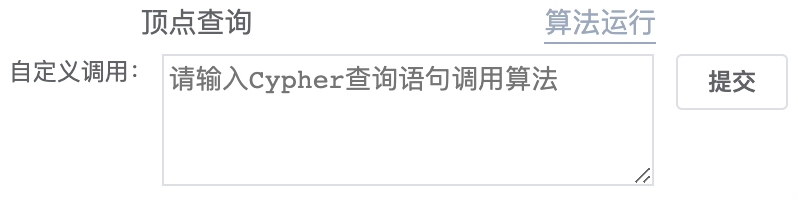
与本文的4.4.4节相同，为了满足专业用户更高、更个性化的查询分析需求，在图数据算法调用模块，系统也设置了自定义调用功能。

图4-20 自定义调用

如图4-20所示，算法运行界面的顶部设置了自定义调用模块，用户在文本框内输入自己编写的Cypher查询语句并提交后，系统将借助neo4j-driver与Neo4j数据库进行通信，调用用户指定的算法，进而在页面显示算法运行结果。

自定义调用模块也体现了系统的灵活性，使得系统能够兼顾普通用户和专业用户的使用需求。

## 4.6 图数据上传模块

图数据上传模块是最能体现图数据查询分析系统的灵活性的功能，图数据上传模块能够满足用户个性化的查询分析需求。

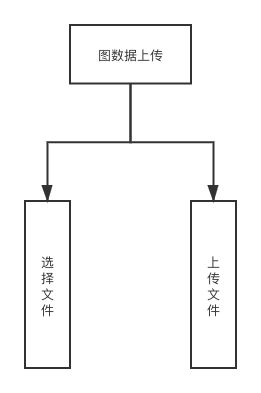
图数据上传模块主要包括选择文件和上传文件两个部分。图数据上传模块结构如图4-21所示。

图4-21 图数据上传模块结构

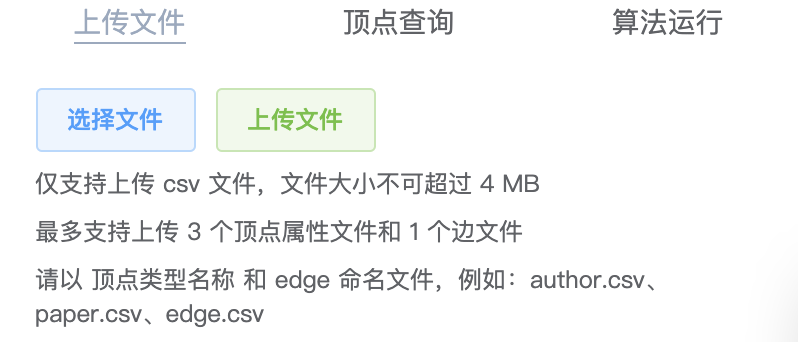
上传图数据首先需要选取要上传的文件，如图4-22所示，点击图数据上传模块中的“选择文件”按钮即可选取文件。文件选择完成后点击“上传文件”，系统就会将文件上传、导入到Neo4j数据库，导入完成系统前端后会弹窗提示，此后便能在图数据可视化展示模块查看上传的图数据的可视化结果，并在顶点与边的查询模块和图数据算法调用模块对其进行查询、分析操作。

图4-22 图数据上传模块

“选择文件”和“上传文件”按钮的下方标明了对上传的文件的类型限制、大小限制、数量限制和命名限制，如果上传的文件不符合限制条件，将会弹窗提示并取消文件选择。

## 4.7 本章小结

本章主要介绍了图数据查询分析系统具体的设计与实现过程。首先分析了系统的需求，确定了系统的总体功能需求和用例；然后综合课题目标和系统总体功能需求考虑，对系统进行了详细设计，将系统分为图数据可视化展示、顶点与边的查询、图数据算法调用和图数据上传四个模块；最后依次介绍了系统各个模块的实现技术。

# 5 实验与结果分析

本章简要说明了课题针对图数据查询分析系统的构建目标，设计、进行了相关测试实验，并对实验结果进行了分析。

## 5.1 实验环境

图数据查询分析系统的实验使用的设备为MacBook Pro(13-inch,2018,Four Thunderbolt 3 Ports)，设备环境如表5-1所示。

表 5-1 实验环境

|  |  |
| --- | --- |
| 环境 | 属性值 |
| 操作系统 | macOS Big Sur 11.2.3 |
| 处理器 | Intel Core i5-8259U CPU @ 2.30GHz（4 CPUs） |
| 内存 | 8 GB 2133 MHz LPDDR3 |
| 硬盘 | SSD固态硬盘512GB |
| 显卡 | Intel Iris Plus Graphics 655 1536 MB |
| 开发环境 | Xcode 12.4，IntelliJ IDEA 2020.3，Neo4j Desktop 1.3.8等 |

## 5.2 实验与结果分析

### 5.2.1 里耶秦简命名实体消歧结果分析

在里耶秦简知识图谱构建的命名实体消歧过程中，本课题先借助SimRank和SimRank++算法，计算出各个人名顶点之间的相似度，筛选出每个顶点的相似度前五百的结果并去除无关顶点；然后对数据进行统计分析，分别以每组结果的相似度的均值、上四分位数、中位数和下四分位数为度量指标，对结果进行筛选，并匹配其他信息进行聚类分析。

（1）SimRank和SimRank++算法运行结果分析

SimRank和SimRank++算法效果对比如图5-1所示。以人名顶点“壬”为例，SimRank将“壬”分为两组，SimRank++将“壬”分为一组；经过以相似度为均值进行筛选，并匹配其他信息进行分析后，可以发现SimRank++的结果更具逻辑，更为合理。

图5-1 SimRank和SimRank++算法效果对比

（2）均值、上四分位数、中位数和下四分位数度量指标分析

使用不同的度量指标筛对SimRank++算法运行结果进行筛选，主要的差异在于最终结果的数量。以相似度的均值、上四分位数、中位数和下四分位数为度量指标分别进行筛选，筛选结果数量如表5-2所示。

表5-2 均值、上四分位数、中位数和下四分位数筛选结果数量

|  |  |
| --- | --- |
| 度量指标 | 结果数量/组 |
| 均值 | 51 |
| 上四分位数 | 44 |
| 中位数 | 70 |
| 下四分位数 | 102 |

分析各个度量指标筛选后的结果数量，可以发现，使用上四分位数筛选后的结果数量较少，使用均值筛选后的结果数量始终，使用中位数和下四分位数筛选后的结果数量较多。

综合考虑两种算法的分组合理性和四项度量指标的筛选后数量效果，最终选用SimRank++算法和均值聚类结果。由于古汉语的特殊性，分组结果不具有百分之百的准确性，需要结合其他上下文信息进一步判断才能确定不同的人名顶点是否为同一人，课题选择不合并人名顶点，而是将分组结果作为人名顶点属性添加。

### 5.2.2 系统功能测试与结果分析

课题将根据本文4.2节的系统详细设计，对图数据查询分析系统的功能进行测试。测试结果如表5-3所示。

表5-3 功能测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能序号 | 功能  名称 | 测试过程 | 测试结果 | 是否符合预期 |
| 1.1 | 图结构可视化展示 | 启动系统，进入初始化界面；执行查询操作 | 实现图结构的可视化展示，包括拖动、缩放、点击等效果 | 是 |
| 1.2 | 顶点信息展示 | 点击“显示/隐藏顶点信息”按钮 | 显示当前图中所有顶点的具体信息 | 是 |
| 1.3 | 边信息展示 | 点击“显示/隐藏边信息”按钮 | 显示当前图中所有边的具体信息 | 是 |
| 2.1 | 顶点分类查询 | 点击顶点分类查询按钮进行查询 | 在图中显示指定类别的顶点 | 是 |
| 2.2 | 顶点id查询 | 输入顶点id，查询指定顶点 | 根据顶点id查询指定顶点及其相关边 | 是 |
| 2.3 | 顶点名称查询 | 输入顶点名称，查询指定顶点 | 根据顶点名称查询指定顶点及其相关边 | 是 |
| 2.3.1 | 时间轴展示 | 在顶点名称查询下，点击“显示/隐藏人物职位变”或“显示/隐藏作者学术生涯”按钮 | 根据当前图中顶点的时间信息生成时间轴 | 是 |
| 2.4 | 自定义查询 | 输入自定义的Cypher查询语句并提交 | 完成自定义顶点和边的查询 | 是 |
|  |  |  |  |  |
| 续表 |  |  |  |  |
| 3.1 | 运行指定算法 | 输入参数并选择算法运行 | 在图上运行指定算法 | 是 |
| 3.1.1 | 输入参数 | 输入运行算法所需参数 | 输入算法运行所需参数 | 是 |
| 3.1.2 | 选择算法 | 点击对应算法的运行按钮 | 选择要在图上运行的算法 | 是 |
| 3.2 | 自定义调用 | 输入自定义的Cypher查询语句并提交 | 执行自定义调用的算法 | 是 |
| 4.1 | 选择文件 | 选取图数据文件 | 选取要上传的图数据文件 | 是 |
| 4.2 | 上传文件 | 上传图数据文件 | 上传已选择的图数据文件 | 是 |

表5-3的测试结果表明，图数据查询分析系统所有功能测试结果均符合预期，系统所有功能正常运行，系统达成了功能测试目标。

### 5.2.3 系统性能测试与结果分析

图数据查询分析系统的性能测试的目的是测试系统的运行效率，主要测试是系统的响应速度。因此，系统性能测试的主要内容为：在里耶秦简数据集上执行系统的各个功能10次，对于包含多种命令的功能，各个命令均执行10次，记录系统从执行各个功能命令到获得执行结果的时间并计算平均执行时间。测试结果如表5-4所示。

表5-4 性能测试结果

|  |  |  |
| --- | --- | --- |
| 功能序号 | 功能名称 | 平均执行时间/秒 |
| 1.1 | 图结构可视化展示 | 2.11 |
| 1.2 | 顶点信息展示 | 0.44 |
| 1.3 | 边信息展示 | 0.55 |
| 2.1 | 顶点分类查询 | 8.89 |
| 2.2 | 顶点id查询 | 3.56 |
|  |  |  |
| 续表 |  |  |
| 2.3 | 顶点名称查询 | 2.34 |
| 2.3.1 | 时间轴展示 | 0.79 |
| 2.4 | 自定义查询 | 2.43 |
| 3.1 | 运行指定算法 | 10.23 |
| 3.1.1 | 输入参数 | 0.05 |
| 3.1.2 | 选择算法 | 2.33 |
| 3.2 | 自定义调用 | 5.67 |
| 4.1 | 选择文件 | 1.22 |
| 4.2 | 上传文件 | 2.75 |

表5-4的测试结果表明，图数据查询分析系统所有功能命令都能够在合理的时间内执行完成，系统达成了预期的性能测试目标。

## 5.3 本章小结

本章主要介绍了图数据查询分析系统的实验与结果分析。首先说明了进行实验的实验环境；其次进行了里耶秦简知识图谱命名实体消歧实验，对比了不同实验的实验结果并阐述了选用SimRank++算法和均值聚类结果的原因；然后根据系统详细设计目标进行了系统功能测试，并分析了测试结果，测试结果符合预期；最后根据系统的构建目标和自身特点进行了性能测试，并分析了测试结果，测试结果符合预期。

# 6 总结与展望

本章对本文的研究内容进行了总结，指出了图数据查询分析系统存在的不足，思考了未来改进方向和工作展望。

## 6.1 本文总结

随着社交网络、零售电商、金融风控、生命科学等行业的快速发展，需要处理的数据之间的关系随数据量呈几何级数增长，越来越多的数据以图数据的形式进行保存和处理。

由于图数据的复杂性和理解难度较其他类型的数据显著提高，本文针对这一问题设计实现了一个通用的图数据查询分析系统，以帮助用户直观、快速、灵活地查询和分析图数据，从而发现其中蕴含的价值。本文主要工作包括以下几个方面：

（1）介绍了图数据查询分析系统的研究背景和国内外研究现状，包括图数据、图可视化系统和图数据库等内容。

（2）介绍了图数据查询分析系统的相关概念与技术，包括主流图数据库对比、Neo4j图数据科学库、知识图谱以及古籍保护与数字化的相关内容。

（3）基于图数据在各个领域广泛存在的现实情况和当前古籍数字化的需求，以里耶秦简数据为切入点，对里耶秦简原始数据进行了命名实体识别、关系抽取和命名实体消歧等处理，构建知识图谱。

（4）借助Neo4j图数据库、Neovis.js、Vue等工具，根据图数据的特点，实现了图数据查询分析系统，包含图数据可视化展示、顶点与边的查询、图数据算法调用、图数据上传等功能，满足用户多样化的查询和深度分析的需求。

## 6.2 未来工作展望

### 6.2.1 优化与改进

课题构建的图数据查询分析系统的测试结果较为理想，达成了课题的构建目标,但是由于受到课题研究时间和实验环境的限制，系统仍然存在诸多不足，有很大的优化和加强的空间，未来可以考虑进行以下改进：

（1）增加对比功能。系统当前只有一个图数据可视化模，可以考虑通过增加图数据可视化模块数量，实现同时显示多个图或同时显示同一个图的不同部分，便于用户进行对比。

（2）提高系统的兼容性。一方面，有必要优化图数据上传模块，减少对上传的图数据文件的限制；另一方面，优化系统界面，增强对不同的硬件环境和软件环境的兼容。

（3）提高系统的扩展性。增加图数据算法包导入接口，实现在系统内运行用户自定义的图数据算法，帮助用户进行专业研究，同时扩展系统图数据算法调用模块的功能。

### 6.2.2 应用场景

图数据查询分析系统达成了课题的构建目标，实现了图数据可视化、自定义查询、图数据相关算法运行、图数据上传等重要功能，能够以满足用户的深度分析需求，可以应用于图数据研究领域和社交网络、零售电商、金融风控、生命科学等相关领域，具有极高的发展潜力和广阔的应用价值。

# 参考文献

1. 陈广. 数据结构（C#语言描述）[M]. 北京: 北京大学出版社, 2008.
2. Singhal A. Introducing the knowledgegraph: things, not strings[J]. Official google blog, 2012.
3. Ji S, Pan S, Cambria E, et al. A survey on knowledge graphs: Representation, acquisition and applications[J]. arXiv preprint arXiv:2002.00388, 2020.
4. Nickel M, Murphy K, Tresp V, et al. A review of relational machine learning for knowledge graphs[J]. Proceedings of the IEEE, 2015, 104(1): 11-33.
5. Paulheim H. Knowledge graph refinement: A survey of approaches and evaluation methods[J]. Semantic web, 2017, 8(3): 489-508.
6. Wang Q, Mao Z, Wang B, et al. Knowledge graph embedding: A survey of approaches and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2724-2743.
7. Lin Y, Han X, Xie R, et al. Knowledge representation learning: A quantitative review[J]. arXiv preprint arXiv:1812.10901, 2018.
8. Wu T, Qi G, Li C, et al. A survey of techniques for constructing Chinese knowledge graphs and their applications[J]. Sustainability, 2018, 10(9): 3245.
9. Chen X, Jia S, Xiang Y. A review: Knowledge reasoning over knowledge graph[J]. Expert Systems with Applications, 2020, 141: 112948.
10. Linkurio. Graph visualization software and web applications[DB/OL]. https://resources.linkurio.us/graph-visualization-software, 2021-03-05.
11. Cambridge Intelligence. Cambridge Intelligence[EB/OL]. https://cambridge-intelligence.com/, 2021-03-05.
12. AMiner. AMiner – AI赋能科技情报挖掘[DB/OL]. https://www.aminer.cn, 2021-03-05.
13. Jiang Y, Huang X, Cheng H, et al. VizCS: Online Searching and Visualizing Communities in Dynamic Graphs[A]. 2018 IEEE 34th International Conference on Data Engineering (ICDE)[C]. Paris: IEEE, 2018, pp.1585-1588.
14. 阿里巴巴集团, 蚂蚁集团. AntV图可视分析解决方案[DB/OL]. https://gw.alipayobjects.com/os/bmw-prod/c8ddbda8-c742-4c11-9c68-3783dd5954b9.pdf, 2021-03-05.
15. DB-Engines. DB-Engines Ranking of Graph DBMS[DB/OL]. https://db-engines.com/en/ranking/graph+dbms, 2021-03-06.
16. DB-Engines. DB-Engines Ranking - Trend of Graph DBMS Popularity [DB/OL]. https://db-engines.com/en/ranking\_trend/graph+dbms, 2021-03-06.
17. Neo4j图数据库. Neo4j作为图数据库有哪些特色和优势？[EB/OL]. https://zhuanlan.zhihu.com/p/125741905, 2021-03-06.
18. Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, et al. Pregel: a system for large-scale graph processing[A]. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data[C]. New York: ACM, 2010, pp.135-146.
19. Neo4j. Neo4j Graph Data Science Library[DB/OL]. https://neo4j.com/product/graph-data-science-library, 2021-03-06.
20. 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.
21. 国家古籍保护中心办公室. 2019年“中华古籍保护计划”实施情况综述[J].古籍保护研究, 2020(02): 1-6.
22. 毛建军. 古籍数字化的概念与内涵[J]. 图书馆理论与实践, 2007(04): 82-84.
23. 熊晶, 钟珞, 王爱民. 甲骨文知识图谱构建中的实体关系发现研究[J]. 计算机工程与科学, 2015, 37(11): 2188-2194.
24. 于彤, 李敬华, 朱玲, 等. 中医临床知识图谱的构建与应用[J]. 科技新时代, 2017(04): 51-54.
25. 胡腾允. 《里耶秦簡（貳）》所見人名統計表[DB/OL]. http://www.bsm.org.cn/show\_article.php?id=3413, 2021-03-11.
26. 陈曙东, 欧阳小叶. 命名实体识别技术综述[J]. 无线电通信技术, 2020, 46(03): 251-260.
27. Baum L E, Eagon J A. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology[J]. Bulletin of the American Mathematical Society, 1967, 73(3): 360-363.
28. Ratnaparkhi A. A maximum entropy model for part-of-speech tagging[C]//Conference on empirical methods in natural language processing. 1996.
29. McCallum A, Freitag D, Pereira F C N. Maximum entropy Markov models for information extraction and segmentation[C]//Icml. 2000, 17(2000): 591-598.
30. Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297.
31. Lafferty J, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[J]. 2001.
32. Lample G, Ballesteros M, Subramanian S, et al. Neural architectures for named entity recognition[J]. arXiv preprint arXiv:1603.01360, 2016.
33. LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
34. Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]//Eleventh annual conference of the international speech communication association. 2010.
35. Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
36. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.
37. 徐健, 吴振新, 张智雄. 实体关系抽取的技术方法综述[J]. 现代图书情报技术, 2008, 24(8).
38. Kambhatla N. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction[C]//Proceedings of the ACL Interactive Poster and Demonstration Sessions. 2004: 178-181.
39. Zhao S, Grishman R. Extracting relations with integrated information using kernel methods[C]//Proceedings of the 43rd annual meeting of the association for computational linguistics (acl’05). 2005: 419-426.
40. Miao Q, Zhang S, Zhang B, et al. Extracting and visualizing semantic relationships from chinese biomedical text[C]//Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation. 2012: 99-107.
41. Brin S. Extracting patterns and relations from the world wide web[C]//International workshop on the world wide web and databases. Springer, Berlin, Heidelberg, 1998: 172-183.
42. Chen J, Ji D, Tan C L, et al. Relation extraction using label propagation based semi-supervised learning[C]//Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. 2006: 129-136.
43. Hasegawa T, Sekine S, Grishman R. Discovering relations among named entities from large corpora[C]//Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). 2004: 415-422.
44. Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[J]. ACM sigmod record, 2000, 29(2): 1-12.
45. 王伟, 孙玉霞, 齐庆杰,等. 基于BiGRU-Attention神经网络的文本情感分类模型[J]. 计算机应用研究.
46. 李秋键. 用Bi-GRU语义解析, 实现中文人物关系分析[DB/OL]. https://mp.weixin.qq.com/s/UNrf41ypcrzYLP9t4rOk2w, 2020-08-11.
47. 温萍梅, 叶志炜, 丁文健, 刘颖, 徐健. 命名实体消歧研究进展综述[J].数据分析与知识发现, 2020, 4(09): 15-25.
48. Bagga A, Baldwin B. Entity-Based Cross-Document Core f erencing Using the Vector Space Model[C]//36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1. 1998: 79-85.
49. Pedersen T, Purandare A, Kulkarni A. Name discrimination by clustering similar contexts[C]//International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Berlin, Heidelberg, 2005: 226-237.
50. Bekkerman R, McCallum A. Disambiguating web appearances of people in a social network[C]//Proceedings of the 14th international conference on World Wide Web. 2005: 463-470.
51. Han X, Zhao J. Named entity disambiguation by leveraging wikipedia semantic knowledge[C]//Proceedings of the 18th ACM conference on Information and knowledge management. 2009: 215-224.
52. Han X, Zhao J. Structural semantic relatedness: a knowledge-based method to named entity disambiguation[C]//Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. 2010: 50-59.
53. 段宗涛, 李菲, 陈柘. 实体消歧综述[J/OL]. 控制与决策: 1-15[2021-03-16]. https://doi.org/10.13195/j.kzyjc.2020.0388.
54. Jeh G, Widom J. SimRank: a measure of structural-context similarity[C]//Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002: 538-543.
55. Antonellis I, Garcia-Molina H, Chang C C. Simrank++ query rewriting through link analysis of the clickgraph (poster)[C]//Proceedings of the 17th international conference on World Wide Web. 2008: 1177-1178.
56. Neo4j. Graph Visualization Tools[DB/OL]. https://neo4j.com/developer/tools-graph-visualization/#\_graphxr\_resources, 2021-03-16.
57. Vue. What is Vuex? [DB/OL]. https://vuex.vuejs.org/#what-is-a-state-management-pattern, 2021-03-16.
58. Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks[J]. Journal of statistical mechanics: theory and experiment, 2008, 2008(10): P10008.
59. Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.
60. Zhu Y, He J, Ye J, et al. When Structure Meets Keywords: Cohesive Attributed Community Search[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020: 1913-1922.

# 致谢

两年的学习时光转瞬即逝，硕士毕业论文的完成也意味着我的学生生涯基本结束了，不禁感慨良多。

首先，衷心感谢我的导师祝园园副教授，本课题的完成离不开她的指导和监督。祝老师学术功底深厚、教学严谨负责、待人随和、关爱学生，这两年来，在学业上和生活上都给予了我很大帮助。

感谢实验室的小伙伴们，与你们一起学习、一起交流令我收获颇多，种草了很多好用的工具，也学会了很多技术和技巧。

感谢我的三位室友，我们一起购物拼单、一起点奶茶、一起分享校内的各种信息、一起参加秋招，令我的校园生活丰富多彩。

感谢中学和大学认识的朋友们，我们是“最熟悉的陌生人”，曾经一起度过了几年的时光，现在又散落在天南海北，时不时与你们聊天，却格外能给我力量，不管选择怎样的人生道路，大家都在努力。

感谢傻乎乎的网友们，我们年龄不同、城市不同，在机缘巧合下结识，逐渐成为了能够分享日常、分享好消息、排解负能量的朋友，和你们聊多了，我好像又回到了只需要烦恼下一顿吃什么和期末考试的本科时光。

最后，感谢我的父母，一直以来都为我的成长殷勤付出，尽可能地满足我的愿望，给予了我很多支持，特别是经济支持。