

知识图谱定义及可视化系统 COIN

(A system for COnstructing and vlsualizing kNowledge graph)

需求规格说明文档

团队名称: nekoCoin

变更记录

修改人员	日期	原因	版本
冯波云	2021/4/1	初稿: 迭代二	ver1.0
冯波云	2021/4/18	更新: 迭代二	ver2.0
冯波云	2021/5/19	更新: 迭代三	ver3.0

1. 引言

1.1 目的

- 本文档描述了知识图谱COIN的功能需求和非功能需求。开发小组的软件系统实现和验证工作都以此文档为依据。
- 除特殊说明外, 本文档所包含的需求都是高优先级需求。

1.2 范围

- "知识图谱定义及可视化系统.pdf"文档中对迭代二功能的规定。
- "CSEIII 03-COIN.pdf"文档中对迭代三功能的规定。

1.3 参考文献

- 《COIN知识图谱定义及可视化系统》
- 《软件工程与计算(二) 软件开发的技术基础》
- 《软件工程与计算(三) 软件开发的团队基础》
- <https://zhuanlan.zhihu.com/p/48612853>
- <https://zhuanlan.zhihu.com/p/26262151>

2. 总体描述

2.1 背景与机遇

随着智能信息服务应用的不断发展, 知识图谱已被广泛应用于智能搜索、智能问答、个性化推荐、情报分析、反欺诈等领域。知识图谱以其强大的语义处理能力与开放互联能力, 可为万维网上的知识互联奠定扎实的基础, 使 Web 3.0 提出的“知识之网”愿景成为了可能。知识图谱是将复杂的信息通过计算处理成能够结构化表示的知识, 所表示的知识可以通过图形绘制而展现出来, 为人们的学习提供有价值的参考, 为信息的检索提供便利。通常知识图谱通过对错综复杂的文档的数据进行有效的加工、处理、整

合，转化为简单、清晰的“实体,关系,实体”的三元组，最后聚合大量知识，从而实现知识的快速响应和推理。然而，市面上大部分的知识图谱平台普遍存在手工操作复杂、可视化过于繁乱等问题。

人们想要获取信息，一般是通过传统的信息检索系统，比如互联网搜索引擎，这种全网式基于关键词搜索，网页集合需要用户浏览过滤筛选，来判断是否是自己需要的信息，经常出现搜索不准确，信息模糊，耽误时间。而智能问答系统的出现，通过简单的交互方式，为用户快速搜索识别准确的答案。无论是信息检索，还是简单的业务办理，智能问答系统都是统一入口，快速便捷完成用户诉求，用户只需提出问题，智能系统会自动识别自然语言，了解用户的真实意图，并及时回复准确答案，缩短了用户获取价值信息的时间，改善了用户体验。

2.2 用户特征

用户：

- 用户使用本项目能够获得知识图谱，并对其进行增加、修改和删除等操作，以此获得结构化的信息，能更好地分析问题。
- 用户可以通过智能问答提出与知识图谱相关的问题并得到答案。

管理员：主要维护知识图谱数据库信息与用户信息

2.3 项目功能

SF1: 允许用户修改图元形状

SF2: 实现以较优的方式自适应展示图谱(减少图元的交叉和重叠)

SF3: 能够在线改变图谱布局，并持久化。(拖动任意节点?)

SF4: 支持搜索图谱内容，包括节点、关系，节点属性的取值。支持模糊匹配。支持搜索历史操作的提示，支持对命中的节点进行标记或者高亮。

SF5: 显示效果调节: 包括调整节点之间的距离、节点 图标的直径大小、调整节点文字大小、是否显示关系的标签。

SF6: 支持图谱统计

SF7: 支持图谱缩放

SF8: 支持切换展示模式

SF9: 支持用户针对知识库提出问题并回答

2.4 约束

CON1: 系统将运行在网页浏览器上。

CON2: 用户、网站管理人员都通过web使用本系统。

2.5 假设与依赖

i. 网络畅通，无故障

ii. 所有操作任务的完成时间不会超过 1 个小时，即使用暂停与中断不会超过 1 个小时

3.1 对外接口需求

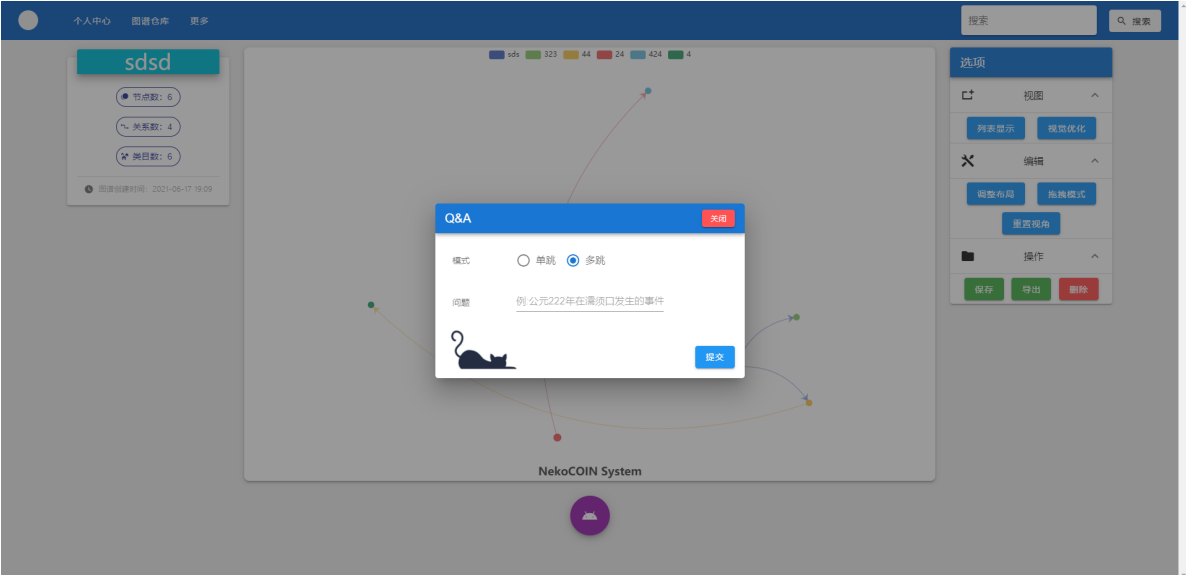
3.1.1 用户界面

整体界面采取扁平化设计，界面最上方为导航栏；右侧有为用户提供的相关操作按钮：列表显示知识图谱，修改知识图谱(文本形式)，导出知识图谱和删除知识图谱。界面中间为知识图片的可视化区域。

点击知识图谱中的节点和关系时，浏览器右侧会出现矩形信息栏。

点击安卓图片，会出点QA窗口。

支持布局调整、视觉优化和节点搜索。



3.1.2 通信接口

本系统用户端与服务器采用http方式进行通信

3.2 功能需求

3.2.1 图元形状可选择

3.2.1.1特征描述

- 优先级：高
- 用户可以根据自身喜好选择图元形状

3.2.1.2 刺激/响应序列

- 刺激：用户点击某图元或新建图元。
- 响应：系统提示可选的图元形状。
- 刺激：用户选择形状后并确认。
- 响应：系统响应并按照形状显示图元。

3.2.1.3 相关功能需求

FrontEnd.graph.addNode	增加节点形状属性传给后端
BackEnd.NodeService.Insert	新增带有形状属性的节点
FrontEnd.graph.editNodes	修改节点属性
BackEnd.NodeService.updateNode	修改节点属性

3.2.2 支持多图谱

3.2.2.1 特征描述

- 优先级：高
- 用户可以上传新图谱或新建图谱和删除图谱

3.2.2.2 刺激/响应序列

- 刺激：用户选择新建图谱
- 响应：系统提示用户新建空图谱或上传已有图谱
- 刺激：用户选择
- 响应：图谱创建成功
- 刺激：用户选择删除某图谱
- 响应：图谱删除成功

3.2.2.3 相关功能需求

FrontEnd.addGraphModal.setMode	记录用户选择：新建空图谱或上传已有图谱
FrontEnd.addGraphModal.createGraph	新建空图谱
FrontEnd.addGraphModal.uploadGraph	导入已有图谱
BackEnd.GraphService.uploadGraphFile	持久化用户图谱

3.2.3 能够在线改变图谱布局，并持久化

3.2.3.1 特征描述

- 优先级：高
- 用户对知识图谱进行在线操作，可以拖动节点，布局会被保存。

3.2.3.2 刺激/响应序列

- 刺激：用户点击拖拽模式
- 响应：系统进入拖拽模式，节点可拖拽
- 刺激：用户左键某一节点图元并拖动
- 响应：图元跟随光标移动到光标停止位置。
- 刺激：点击保存
- 响应：拖拽后的节点位置将持久化
- 刺激：用户再次点击拖拽模式
- 响应：系统退出拖拽模式。

3.2.3.3 相关功能需求

BackEnd.NodeService.updateNode	持久化实体位置信息到数据库
FrontEnd.optionModal.enableDraggable	使节点可以拖拽
FrontEnd.optionModal.saveLayout	前端保存新的布局方法

3.2.4 节点搜索

3.2.4.1 特征描述

- 优先级：高
- 用户可以根据一些图元属性搜索节点，支持模糊匹配，保存搜索记录。命中节点高亮。支持历史记录功能

3.2.4.2 刺激/响应序列

- 刺激：在右上角输入框键入关键字并点击搜索按钮
- 响应：命中的节点和相关关系将被高亮
- 刺激：用户点击取消搜索
- 响应：节点恢复正常显示

3.2.4.3 相关功能需求

FrontEnd.search.getEntity	从localStorage中获取搜索记录
FrontEnd.search.renewHistoryAndSearch	用户点击搜索按钮后执行，更新历史记录并高亮目标节点

3.2.5 显示效果调节

3.2.5.1 特征描述

- 优先级：高
- 用户可以调整节点之间的距离、节点图标直径大小、调整节点文字大小、是否显示关系的标签。支持保存。

3.2.5.2 刺激/响应序列

- 刺激：用户点击显示效果调节
- 响应：系统弹出选择表格
- 刺激：用户拖动滑轮
- 响应：系统根据滑轮的改变更新图谱显示

3.2.5.3 相关功能需求

FrontEnd.modifyViewModal.symbolSizeChange	改变节点大小
FrontEnd.modifyViewModal.showLabelChange	修改节点文字和关系文字显示范围
FrontEnd.modifyViewModal.commonChange	修改力导图模式下节点间斥力和节点文字大小
FrontEnd.modifyViewModal.submit	保存用户修改的视觉效果

3.2.6 支持图谱统计

3.2.6.1 特征描述

- 优先级：高
- 用户可以获取图谱的相关统计数据(节点、关系数量，节点类目数量)。

3.2.6.2 刺激/响应序列

无，系统自动显示

3.2.6.3 相关功能需求

FrontEnd.graphInfoModal	显示统计卡片
BackEnd.GraphService.getAllNodesAndRelations	获取当前图谱所有关系和节点，通过数组长度得到节点、关系、类目数量

3.2.7 支持图谱缩放和节点过滤

3.2.7.1 特征描述

- 优先级：高
- 知识图谱可以缩放，可与进行节点过滤

3.2.7.2 刺激/响应序列

- 刺激：用户通过鼠标滚轮缩放指数图谱
- 响应：系统调整图谱大小。
- 刺激：用户点击某类目的标签
- 响应：该类目的节点和相关关系将不显示
- 刺激：用户再次点击某类目的标签
- 响应：该类目的节点和相关关系将重新显示

3.2.7.3 相关功能需求

无：echarts实现

3.2.8 显示模式切换和重置视角

3.2.8.1 特征描述

- 优先级：高
- 知识图谱支持多种显示模式

3.2.8.2 刺激/响应序列

- 刺激：用户点击显示模式切换
- 响应：系统给出可选模式
- 刺激：用户选择模式并确认
- 响应：图谱显示模式改变
- 刺激：用户点击重置视角按钮
- 响应：图谱移动到初始状态

3.2.8.3 相关功能需求

FrontEnd.optionModal.alterMode	切换显示模式
FrontEnd.optionModal.resetMonitor	重置视角
FrontEnd.optionModal.saveLayout	支持排版模式下实体和关系的位置的持久化

3.2.9 支持用户针对知识库提出问题并回答

3.2.9.1 特征描述

- 优先级：高
- 支持单跳和多跳两种问题。

3.2.9.2 刺激/响应序列

- 刺激：用户点击问答按钮
- 响应：进入问答界面
- 刺激：用户键入问题并点击发送
- 响应：从后端返回答案并显示

3.2.9.3 相关功能需求

Bayes.Code.py	接收前端请求并映射到相关服务
Bayes.preprocess_data.py	接收原始问题,对原始问题进行分词、词性标注等处理 对问题进行抽象
Bayes.question_classification.py	对问题进行分类
Bayes.question_template.py	针对不同类型问题实现不同查询语句, 返回答案
Bert.answer()	接收前端请求并映射到相关服务
Bert.text_match()	文字直接匹配, 查看属性的词语是否在句子之中
Bert.select_neo4j()	连接数据库并查询
Bert.semantic_matching()	语义匹配
Bert.get_entity()	获取问题实体
Bert.get_sim_model()	获取sim模型
Bert.get_ner_model()	获取ner模型

3.3 非功能需求

3.3.1 安全性

- Safety1：用户只能查看和维护自己的个人信息、知识图谱。
- Safety2：网站管理人员不能随意违规查看、修改用户的信息。
- Safety3：为防信息泄露，用户的账号、密码、姓名（名称）、联系方式等敏感数据必须密文存储。

3.3.2 可维护性

- Modifiability1: 关系和实体, 将来可能会出现新的类型:
 - Modifiability1.1: 当关系和实体出现新类型时, 系统应该能够在4个人一天内更新完毕

3.3.3 易用性

- Usability1: 客户初次使用时, 能在十分钟内完成从知识图谱可视化到增加或删除节点的操作。

3.3.4 可靠性

- Reliability1:在客户端与服务器通信时, 如果网络故障, 系统不能出现故障。
 - Reliability1.1:客户端应该检测到故障, 并尝试重新连接网络3次, 每次15秒。
 - Reliability1.1.1:重新连接后, 客户端应该继续之前的工作。
 - Reliability1.1.2:如果重新连接不成功, 客户端应该等待5分钟后再次尝试重新连接。
 - Reliability1.1.2.1:重新连接后, 客户端应该继续之前的工作。
 - Reliability1.1.2.2:如果重新连接仍然不成功, 客户端报警。

3.3.5 约束

- IC1:系统要在网络上分布为一个服务器和多个客户端。

3.4 数据需求

3.4.1 数据定义

- DR1:系统需要存储单个用户的完成的增删改查记录。
- DR3:数据实体及其关系详情见下表:

对象类型	持有的数据
实体	所属图谱ID String graphId 实体名称 String name 自身属性Map<String, Object> entityProperties
关系	关系出发节点名称String source 关系目标节点String target 关系值String value 所属图谱ID String graphId 自身属性Map<String, Object> styles
图谱	组成图谱的节点List<Entity> entities 组成图谱的关系List<Relation> links 图谱包含的类目List<Category> categories 图谱属性Map<String, Object>graphProperties
类目	类目属性Map<String, Object>categoryProperties

3.4.2 默认数据

- Default1:实体类目默认为0。
- Default2:类目名称默认为"类目0"。
- Default3:系统时间默认为当天。
- Default4:关系的连线类型默认为实线。

- Default5:节点图元形状默认为圆形。

3.4.3 数据格式要求

- Format1:节点的数量必须为正整数。
- Format2: 坐标为双精度浮点数，可以是负值。
- Format3: 密码格式 长度 ≥ 6 的字符串。
- Format4: 日期格式 yyyy-MM-dd。
- Format5: 时间格式 HH:mm:ss。
- Format6: 日期时间，都采用北京时间，即“+8:00”。
- Format7: 数量，都是非负整数。