

政靠“谱”:基于图嵌入的政策智能知识图谱推荐系统后端文档

目录

- 政靠“谱”:基于图嵌入的政策智能知识图谱推荐系统后端文档 1
 - 介绍 1
 - 运行 1
 - 模块 2
 - 路由函数 2
 - index() 2
 - search() 2
 - qa_system() 2
 - all_relation() 3
 - edit_relation() 3
 - get_profile() 3
 - qa_system_answer() 3
 - search_name() 3
 - search_all_objects() 4
 - add_relation() 4
 - del_relation() 4

介绍

这是一个使用 Flask 框架开发的后端应用，主要功能是提供一个问答系统和一个图数据库编辑器。问答系统可以根据用户输入的问题，从图数据库中查询相关的实体和关系，并返回答案。图数据库编辑器可以让用户查看、添加、删除图数据库中的实体和关系。

运行

要运行这个应用，需要先安装好依赖的库，然后在命令行中执行以下命令：

```
python app.py
```

复制

这将启动一个本地服务器，监听 8080 端口。然后可以在浏览器中访问以下地址：

- <http://localhost:8080/>：首页
- <http://localhost:8080/search>：搜索页面
- http://localhost:8080/qa_system：问答系统页面
- http://localhost:8080/all_relation：所有关系页面
- http://localhost:8080/edit_relation：编辑关系页面

模块

这个应用主要由以下几个模块组成：

- `app.py`: 主模块，定义了 Flask 应用对象和路由函数
- `neo4j_db`: 图数据库模块，封装了与图数据库的交互方法
- `qa_system`: 问答系统模块，封装了问答系统的逻辑和自然语言处理方法
- `templates`: 模板文件夹，存放了 HTML 文件

路由函数

路由函数是定义在 `app.py` 中的函数，它们用装饰器 `@app.route` 来指定对应的 URL 路径和请求方法。路由函数负责接收请求参数，调用相应的模块方法，渲染模板文件，返回响应数据。以下是各个路由函数的功能和参数说明：

`index()`

该函数对应于 URL 路径 `/'和'/index'`，支持 GET 和 POST 方法。它渲染并返回首页的 HTML 文件。

参数：

- `name`: 可选参数，表示用户的姓名

`search()`

该函数对应于 URL 路径 `/'search'`，支持 GET 和 POST 方法。它渲染并返回搜索页面的 HTML 文件。

`qa_system()`

该函数对应于 URL 路径'/qa_system'，支持 GET 和 POST 方法。它渲染并返回问答系统页面的 HTML 文件。

all_relation()

该函数对应于 URL 路径'/all_relation'，支持 GET 和 POST 方法。它渲染并返回所有关系页面的 HTML 文件。

edit_relation()

该函数对应于 URL 路径'/edit_relation'，支持 GET 和 POST 方法。它渲染并返回编辑关系页面的 HTML 文件。

get_profile()

该函数对应于 URL 路径'/get_profile'，支持 GET 和 POST 方法。它接收一个参数 `character_name`，调用 `neo4j_db` 模块的 `get_answer_profile` 方法，查询图数据库中该实体的属性，并返回一个 JSON 格式的数据。

参数：

- `character_name`: 必选参数，表示要查询的实体名称

qa_system_answer()

该函数对应于 URL 路径'/qa_system_answer'，支持 GET 和 POST 方法。它接收一个参数 `name`，表示用户输入的问题，调用 `qa_system` 模块的 `get_target_array` 和 `get_qa_system_answer` 方法，对问题进行自然语言处理和图数据库查询，并返回一个 JSON 格式的数据，包含问题的答案和相关的实体和关系。

参数：

- `name`: 必选参数，表示用户输入的问题

search_name()

该函数对应于 URL 路径'/search_name'，支持 GET 和 POST 方法。它接收一个参数 `name`，表示要搜索的实体名称，调用 `neo4j_db` 模块的 `query` 方法，查询图数据库中该实体的所有关系，并返回一个 JSON 格式的数据，包含实体的属性和关系。

参数：

- `name`: 必选参数，表示要搜索的实体名称

search_all_objects()

该函数对应于 URL 路径'/search_all_objects'，支持 GET 和 POST 方法。它调用 neo4j_db 模块的 query_all_objects 方法，查询图数据库中所有的实体，并返回一个 JSON 格式的数据，包含所有实体的名称和类别。

add_relation()

该函数对应于 URL 路径'/add_relation'，支持 GET 和 POST 方法。它接收五个参数 obj1, obj2, relation, cate1, cate2，表示要添加的关系的两个实体、关系名称、两个实体的类别。它调用 neo4j_db 模块的 add_relation_graph 方法，向图数据库中添加一条新的关系，并返回一个 JSON 格式的数据，表示添加结果。

参数：

- obj1: 必选参数，表示关系的起始实体
- obj2: 必选参数，表示关系的终止实体
- relation: 必选参数，表示关系的名称
- cate1: 必选参数，表示起始实体的类别
- cate2: 必选参数，表示终止实体的类别

del_relation()

该函数对应于 URL 路径'/del_relation'，支持 GET 和 POST 方法。它暂未实现。