说明文档

|  |  |
| --- | --- |
| 文档编号： | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 版本号 ： | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 项目名称： | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 项目编号： | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 编 制： | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 批 准： | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

版本修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 版本 | 修改描述 | 修改人 | 生效日期 |
| 1 | V1.0 | 文档说明 | 邵妍洁 | 2019.6.4 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[第一章 前言 4](#_Toc10576353)

[1.1说明 4](#_Toc10576354)

[1.2部署说明 4](#_Toc10576355)

[1.3配置说明 4](#_Toc10576356)

[第二章 组件说明 5](#_Toc10576357)

[2.1 通用知识图谱的构建 5](#_Toc10576358)

[2.2 有害知识图谱的构建 6](#_Toc10576359)

[2.3 标注平台数据处理组件 7](#_Toc10576360)

[2.3.1 标注平台数据解析模块 7](#_Toc10576361)

[2.3.2 标注平台数据导入模块 9](#_Toc10576362)

[2.4 有害识别组件 11](#_Toc10576363)

[2.4.1 数据提取模块 11](#_Toc10576364)

[2.4.2 数据导入模块 12](#_Toc10576365)

[2.4.3 有害查询模块 13](#_Toc10576366)

1. 前言

## 1.1说明

本文档为利用知识图谱进行有害信息识别的相关组件说明书，主要包括了对通用知识图谱的构建、有害知识图谱的构建、标注平台数据处理组件以及有害识别组件的说明。

## 1.2部署说明

随文档提交的各部分组件以python代码import的形式调用。部署是需要满足组件基本信息中“部署相关信息”部分的要求，主要包括Neo4j图数据库和常见的python依赖库。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 依赖库 | 通用图谱组件 | 有害图谱组件 | 标注平台组件 | 有害识别组件 |
| xlrd |  |  |  | √ |
| xlwt |  |  |  | √ |
| jieba |  |  |  | √ |
| numpy |  |  |  | √ |
| pandas |  |  |  | √ |
| openpyxl |  | √ |  | √ |
| py2neo |  | √ |  | √ |
| configparser | √ | √ |  | √ |
| json | √ |  | √ |  |
| Neo4j | √ | √ | √ | √ |
| java | √ | √ | √ | √ |

## 1.3配置说明

在具体部署代码时，需要根据Neo4j图数据库的IP和端口号进行连接，可直接在配置文件setup.conf中修改graph\_db相关数据。默认为本机IP和7474端口。连接Neo4j需要使用用户名和密码，默认用户名为“neo4j”，默认密码为“123456”。

1. 组件说明

## 2.1 通用知识图谱的构建

通用知识图谱构建的相关说明如下所示，将wikidata上检索到的网信行业的相关数据（CSV格式存储）转化为了Neo4j的图进行显示。

|  |  |
| --- | --- |
| 基本说明 | 目前主要包含的是网信行业相关的知识，可分为人、地、组织、武器等几个大类。  通过putincsv.py将csv数据导入图数据库。需要提前将所有csv数据放在该图数据库的import路径下，同时需要再配置文件setup.conf中修改importpath。 |
| 功能信息 | 为有害识别的过程提供客观知识支撑。 |
| 接收参数类型 | CSV data |
| 返回类型 | Neo4j graph |
| 接收参数说明： | **实体csv**：带有表头的csv文件，记录节点信息，在没有对应信息的情况下值为None。  表头为：  [wiki\_id, name, descriptions, aliases, name\_zh, descriptions\_zh, aliases\_zh, :LABEL]  **关系csv**：带有表头的csv文件，记录节点间的关系信息。  表头为：  [from\_id, to\_id, relationship] |
| 返回参数说明： | C:\Users\SHAOYA~1\AppData\Local\Temp\1559210133.png  Ulan、Nie Heting两个节点分别包含了相应人物的姓名、别名、描述等属性信息；两者均为politician（政治家）节点（也包含名称、别名等属性）的下级实体。 |
| 部署相关信息： | Java环境  Neo4j |
| **注：** |  |

## 2.2 有害知识图谱的构建

在判定文本信息是否有害的过程中，需要将其中的实体、关系抽取出来并生成相应的知识图谱，即有害知识图谱。有害知识图谱构建的相关说明如下所示，将中科软科技股份有限公司所提供的标注好了实体、关系的有害数据（Excel格式存储）转化为了Neo4j的图进行显示。

|  |  |
| --- | --- |
| 基本说明 | 目前主要包含了7111条被标注为了不同有害类型的文本信息。 |
| 功能信息 | 和通用知识图谱一起，为有害文本的规则的提取提供基础，即因为存在何种实体、关系而造成文本信息有害，从而扩展到对更多文本进行有害判定。 |
| 接收参数类型 | Excel文件 |
| 返回类型 | Neo4j graph |
| 接收参数说明： | Excel：带有表头的Excel文件，记录数据文本，有害类型，标注的文本三元组，在没有对应信息的情况下值为None。  表头为：  [text, label, relat,] |
| 返回参数说明： | DiggBt和负面信息分别是三元组的两个实体，关系是发布，两个实体与原文以come from的关系连接，图中显示的是原文的有害类型。 |
| 部署相关信息： | Java环境  Neo4j  openpyxl  代码模块为triple\_to\_neo4j.py。 |
| **注：** | \* 标注的文本信息中的实体、关系待调整。  \* excel文件的后缀必须是.xlsx。 |

## 2.3 标注平台数据处理组件

#### 2.3.1 标注平台数据解析模块

标注平台数据解析模块对标注平台输出的数据进行数据清洗与数据分类，输出符合Neo4j图数据库导入接口的结构化数据。

|  |  |
| --- | --- |
| 接口方法原型 | read\_annotation |
| 接口方法接收参数类型 | json |
| 接口方法返回类型 | list |
| 抛出异常 | 无 |
| 接收参数说明： | {  "text": "巴格达市中心于23日遭火箭弹袭击",  "harm\_type": "3\_2\_2\_3",  "triple": [  {  "subject": {  "word": "火箭弹",  "label": "ITEM"  },  "predicate": {  "word": "袭击",  "label": "ITEM-LOC",  "is\_mutual": false,  "sentiment": "0",  "tendency": "0",  "attribute": [  {  "time": "2019-09-23"  }  ]  },  "object": {  "word": "巴格达市中心",  "label": "LOC"  }  }  ],  "entity": [  {  "word": "巴格达",  "label": "LOC",  "attribute": []  }  ]  }  即标注平台数据的接口结果，具体见中科软科技有限公司的三元组标注规范说明书。 |
| 返回参数说明： | --- main\_entity ---  {'label': 'text', 'harm\_type': '3\_2\_2\_3', 'text': '巴格达市中心于23日遭火箭弹袭击'}  --- triples ---  [{'subject': {'word': '火箭弹', 'label': 'ITEM'}, 'predicate': {'word': '袭击', 'label': 'ITEM-LOC', 'is\_mutual': False, 'sentiment': '0', 'tendency': '0', 'attribute': [{'time': '2019-09-23'}]}, 'object': {'word': '巴格达市中心', 'label': 'LOC'}}]  --- entities ---  [{'word': '巴格达', 'label': 'LOC', 'attribute': [{'time': '2019-09-23'}]}]  结果列表的第一项为原文本信息，本身依旧以json格式存储；第二项为三元组列表，第三项为实体列表。 |
| 部署相关信息： | 代码模块为annotation\_data.py，可直接从该模块import接口方法原型。 |
| **注：** | \* 解析模块和导入模块共同构成标注平台数据处理组件。 |

#### 2.3.2 标注平台数据导入模块

标注平台数据导入模块自动化地将解析后的结构化数据转化为Neo4j的图进行显示。

|  |  |
| --- | --- |
| 接口方法原型 | store\_annotation |
| 接口方法接收参数类型 | json |
| 接口方法返回类型 | Neo4j graph |
| 抛出异常 | 无 |
| 接收参数说明： | --- main\_entity ---  {'label': 'text', 'harm\_type': '3\_2\_2\_3', 'text': '巴格达市中心于23日遭火箭弹袭击'}  --- triples ---  [{'subject': {'word': '火箭弹', 'label': 'ITEM'}, 'predicate': {'word': '袭击', 'label': 'ITEM-LOC', 'is\_mutual': False, 'sentiment': '0', 'tendency': '0', 'attribute': [{'time': '2019-09-23'}]}, 'object': {'word': '巴格达市中心', 'label': 'LOC'}}]  --- entities ---  [{'word': '巴格达', 'label': 'LOC', 'attribute': [{'time': '2019-09-23'}]}]  即上述数据导入模块的返回参数。说明同上。 |
| 返回参数说明： | **C:\Users\SHAOYA~1\AppData\Local\Temp\WeChat Files\6edec9d5ef609765028746cf7ce4ac7.jpg**  **C:\Users\SHAOYA~1\AppData\Local\Temp\WeChat Files\3b28562dbc107895e3c8c82222d3994.jpg**  comeFrom都指向的为原文本信息实体节点，有害类别等均以其属性形式进行了存储；“火箭弹”、“巴格达市中心”分别为三元组的主语实体和宾语实体；“袭击”为三元组的谓语，在图中以关系形式呈现，同时'is\_mutual'、'sentiment'、'tendency'等均以其属性形式进行了存储；“巴格达”为提取出的地点实体。 |
| 部署相关信息： | 代码模块为annotation\_data.py，可直接从该模块import接口方法原型。 |
| **注：** | \* 解析模块和导入模块共同构成标注平台数据处理组件。 |

## 2.4 有害识别组件

#### 2.4.1 数据提取模块

数据提取模块对中科软科技有限公司提供的线下标注好了实体、关系的Excel文件中的数据进行清洗，然后以定义的结构化形式输出。

|  |  |
| --- | --- |
| 接口方法原型 | load\_data |
| 接口方法接收参数类型 | Excel文件 |
| 接口方法返回类型 | list, list, list |
| 抛出异常 | 无 |
| 接收参数说明： | Excel：带有表头的Excel文件，记录数据文本，有害类型，标注的文本三元组，在没有对应信息的情况下值为None。  表头为：  [text, label, relat,] |
| 返回参数说明： | 返回参数为text、label、triple，分别表示原文本、有害类型和标注的文本三元组。 |
| 部署相关信息： | openpyxl |
| **注：** | \* Excel文件的后缀必须是.xlsx。 |

#### 2.4.2 数据导入模块

数据导入模块将数据提取模块的输出导入Neo4j图数据库中进行存储。

|  |  |
| --- | --- |
| 接口方法原型 | import\_data |
| 接口方法接收参数类型 | list, list, list |
| 接口方法返回类型 | Neo4j graph |
| 抛出异常 | IndexError, ValueError |
| 接收参数说明： | 返回参数为text、label、triple，分别表示原文本、有害类型和标注的文本三元组。 |
| 返回参数说明： | DiggBt和负面信息分别是三元组的两个实体，关系是发布，两个实体与原文以come from的关系连接，图中显示的是原文的有害类型。 |
| 部署相关信息： | py2neo  Neo4j  configparser |
| **注：** | \* 当抛出异常时，表明标注的三元组有误。 |

#### 2.4.3 有害查询模块

有害查询模块在通用知识图谱里检索文本信息中出现的实体及本体概念，并将结果与提取出的有害文本规则进行比对。若一致，则认为有害，并标记上相应规则所属的有害类型。最终返回的三个参数分别是原文本信息、有害类型以及文本信息中与有害规则相关的关键词列表。

|  |  |
| --- | --- |
| 接口方法原型 | identify\_harmful |
| 接口方法接收参数类型 | string（原文本信息） |
| 接口方法返回类型 | string, string, list（分别对应原文本信息、有害类型以及文本信息中与有害规则相关的关键词列表） |
| 抛出异常 | 无 |
| 接收参数说明： | text = ‘xijinping谈七五事件’ |
| 返回参数说明： | text = ‘xijinping谈七五事件’  harmful\_type = ‘1\_1\_1’  keywords = ['xijinping', '七五事件']  （text：原文本信息，harmful\_type：有害类型，keywords：文本信息中与有害规则相关的关键词列表） |
| 部署相关信息： | 代码模块为judge\_harmfulstring.py，可直接从该模块import接口方法原型。 |
| **注：** |  |