航行通告NOTAM知识图谱

> 项目开发者：刘俊麟

项目流程

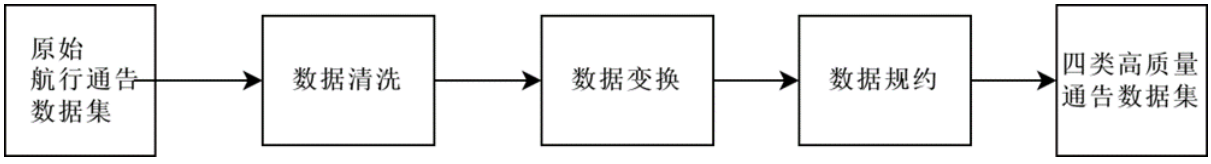
本项目基于现有航行通告数据构建航行通告事件要素实体与对应关系的知识图谱。现有数据为2022年度国内外原始航行通告集。此次航行通告知识图谱构建选择2022年度上海情报区机场类事件数据。

主要工作包括：（1）数据预处理（2）实体抽取与关系构建（3）导入知识图谱与结果展示（4）知识图谱应用

1.数据预处理

针对航行通告结构形式与航行通告格式特点，需要对原电信形式且面向民航人员的航行通告数据集进行数据预处理，转换为格式统一且面向机器的航行通告数据集，以提高数据质量，进而提高挖掘结果的质量。

原始航行通告数据集具有不完整性，含噪声，不一致性。分析航行通告文本语义，其相关数据预处理方法如下：
（1）数据清洗：删除或修正不完整，错误或无关的数据。此步骤清洗掉非C系列的航行通告，非目标情报区的航行通告，雪晴通告，“见下一份”报等。
（2）数据变换：将数据转换为适合分析和建模的格式。在此步骤将E项电报码转换为中文字符，将多个分开电报整合为一份等。
（3）数据规约：选择或提取与分析目标相关的属性。在此步骤根据Q码主题进行内容分类，分为机场类，通信、导航和监视类，空中交通管理类，航空警告类。



具体步骤如下：

从总数据集里提取出上海飞行情报区2022年度航行通告集 → 提取出C系列的非雪情航行通告 → 将航行通告E项中文电报码转译为中文字符 → 进行数据清理，删除带有"
"CNL", "已 编 入 永 久 性 资 料", "SEE NEXT NTM”等的航行通告 → 针对取消，替代报

的时间逻辑删除合并航行通告 → 对非E项进行实体抽取 → 针对Q码对航行通告打上主题，状况标志 → 将总航行通告分类成"航空警告类”，"空中交通管理类”"通信、导航和监视类”"机场类”四个xlsx表格。

注：
以上步骤主要采用正则表达式和字典映射的方法进行数据预处理。参考文件为《民用航空航行通告代码选择规范》与《民用航空航行通告编发规范》。

2.实体抽取与关系构建

（1）结构化数据实体抽取。

航行通告为结构化数据，非E项文本为结构化数据，E项文本为非结构化数据。针对航行通告结构化内容，使用基于正则表达式的实体信息抽取技术。

结构化数据块对应正则表达式如下：

事项块	正则表达式
系列与编号	[C](.{4})/(.{2})
航行通告类型	NOTAM[NCR]?
被取消或被替代编号	[C] (.{4})/(.{2})
飞行情报区/Q代码/飞行种类/飞行目的/影响范围/下限与上限	Q\)\s*([a-zA-Z]{4})/([a-zA-Z]{5})/([a-zA-Z]{0,2})/([a-zA-Z]{0,5})/([a-zA-Z]{0,2})/(.{3})/(.{3})/
标和半径	/(\d{4}[A-Z]\d{5}[A-Z]\d{3})
发生地与生效时间	/(\d{4}[A-Z]\d{5}[A-Z]\d{3})
失效时间	C\)\s*(.*)\n
E项文本	E\)\s*(.+)\)

（2）非结构化数据实体抽取。

由于航行通告E项文本采用明语编写，具有自由性与复杂性，以及其专有名词的特殊性，难以采用统一的格式进行处理。本文选择在模式匹配技术的基础之上，实现信息抽取任务。本文结合Q码主题与状况，采用基于规则的spaCy匹配进行信息提取。

本文构建数据为机场类事件数据，部分示例匹配器设置如下:

示例输入	实体类型	匹配器设置	命名实体识别输出
(1)RWY 18R 跑道状况5/5/5道面湿 观	跑道名称Name	matcher_Name = PhraseMatcher(nlp.vocab,	(1){"Name": [RWY 18R]} (2){"Name":

测时间 2210300031 (2)RWY15/33 关闭，因跑道除胶维护		attr="SHAPE") matcher_Name.add("Name", [nlp("RWY 01L/19R"), nlp("RWY 01"), nlp("RWY 01L"), nlp("RWY01"), nlp("RWY 08/26"), nlp("RWY01L/19R")])	[RWY15/33]}
	状态Act	matcher_act = PhraseMatcher(nlp.vocab, attr="TEXT") matcher_act.add("Act", [nlp("关闭"),nlp("取消"), nlp("开放")])	(1){"Act":[""]} (2) {"Act":["关闭"]}
	原因Due	pattern = [{"POS": "ADP"}, {"POS": "NOUN", "OP": "*"}, {"POS": "VERB"}]	(1){""} (2){"因 跑道 除 胶 维 护"}

（3）固定关系构建

对航行通告拍发格式与内容的分析，可以从不同事项中提取不同的实体，并结合航行通告语义为实体与实体间添加适合的属性，即关系。航行通告体现了特定民航事件的时间性与空间性，由此可以从中提取相关事件实体，时间实体，空间实体。其中E项没有规范格式，存在非结构化文本，可以结合Q码的主题与状况高度概括与提取航行通告事件实体，可提取出具体名称Name，活动状态Act，限制时间Time，事件原因Due。

主要固定关系如下：

特性	属性/关系
事件性	Q-type-飞行类型， Q-purpose-飞行目的， 具体名称-Act-活动状态 Name-Due-事件原因
时间性	Q-StartOn-生效时间 Q-EndOn-失效时间 具体名称-Time-限制时间
空间性	Q-Geo-坐标 Q-Upper-上限 Q-Lower-下限 Q-Scope-影响范围 Q-hashappendOn-机场

3.导入知识图谱与结果展示

我们将处理好的数据文件`airport_Element.csv`与`airport_Relation.csv`存入neo4j图数据库下`import`文件夹，并在neo4j中创建相应节点标签和关系类型，代码如下：

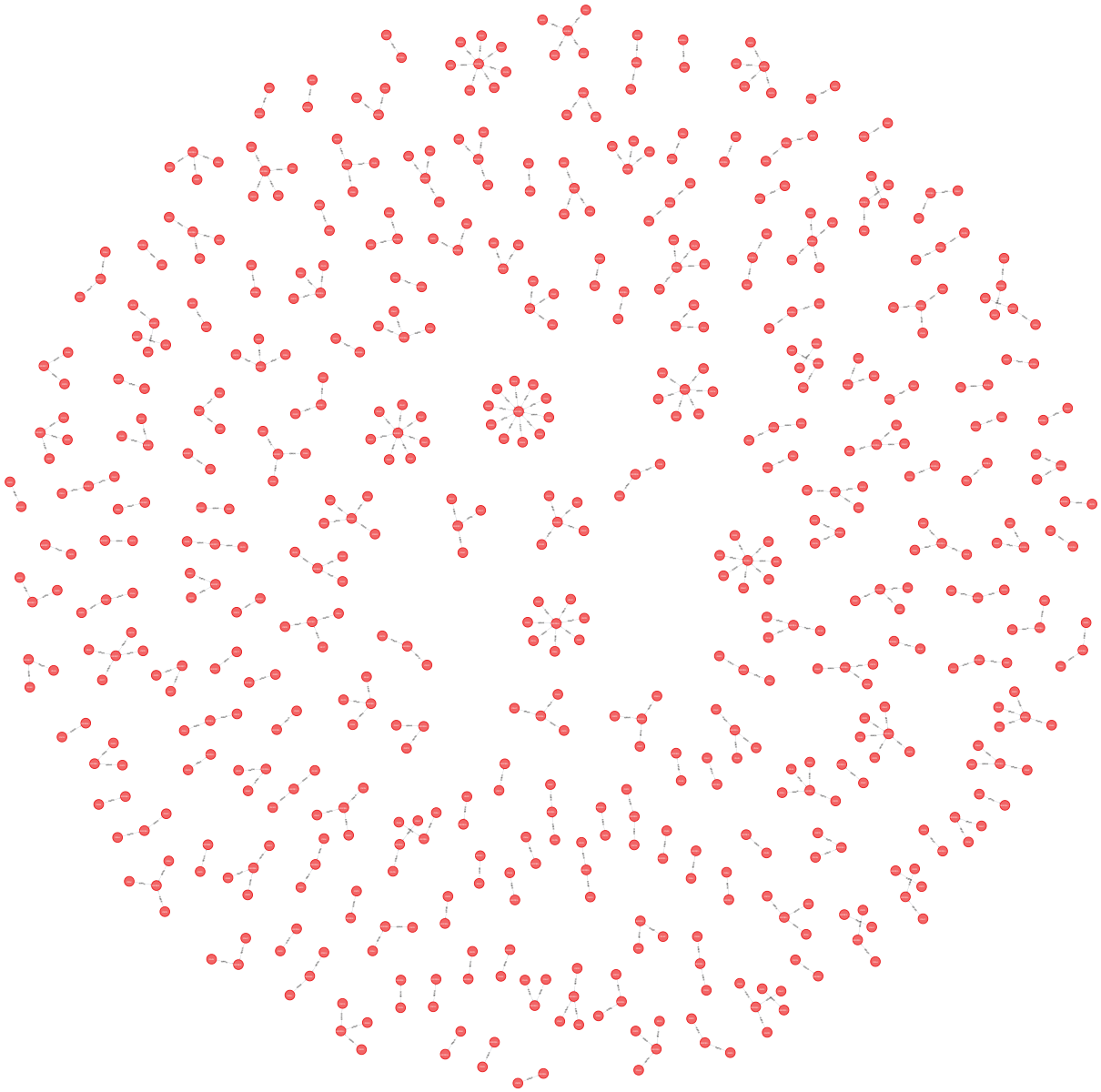
```
#导入机场事件数据
load csv from 'file:///airport_relation.csv' as line
```

```
create(:airtportRelation{from:line[0],subject:line[1],to:line[2],object:line[3],relation:line
[4]})
load csv from 'file:///airportTEST.csv' as line
create(:airtportElement{num:line[0],name:line[1]})

#创建航行通告机场事件关系图谱
match (n:airportElement),(m:airportRelation),(s:airportElement) where n.num=m.subject and s.num
=m.object
create (n)-[r:airport2{relation:m.relation}]->(s)
```

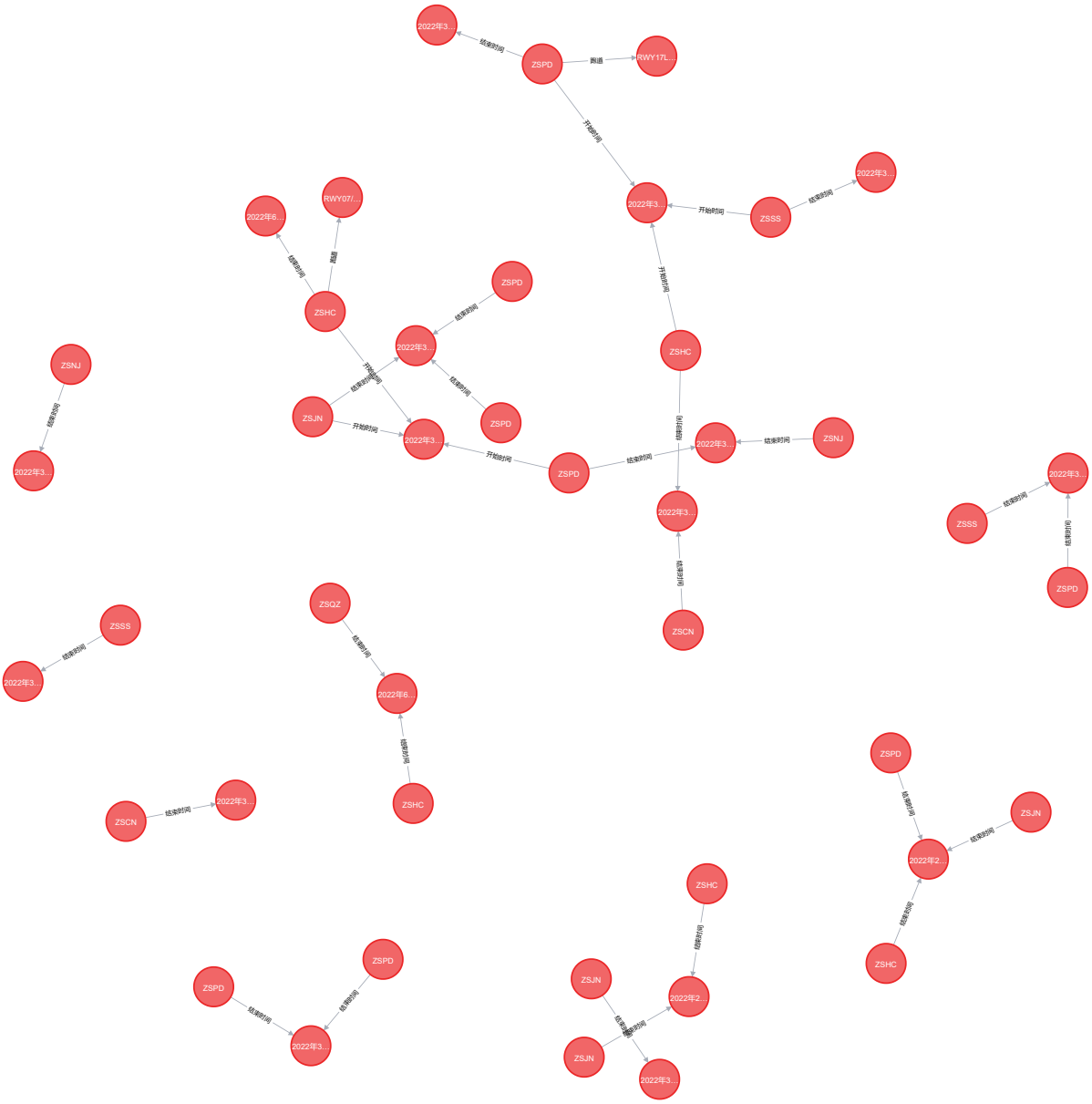
知识图谱整体可视化结果展示如下（仅显示前600个节点）：

```
MATCH p=()-[r:airport2]->() RETURN p
```



展示部分知识图谱可视化结果如下：

```
MATCH p=()-[r:airport2]->() RETURN p LIMIT 25
```



4.知识图谱应用

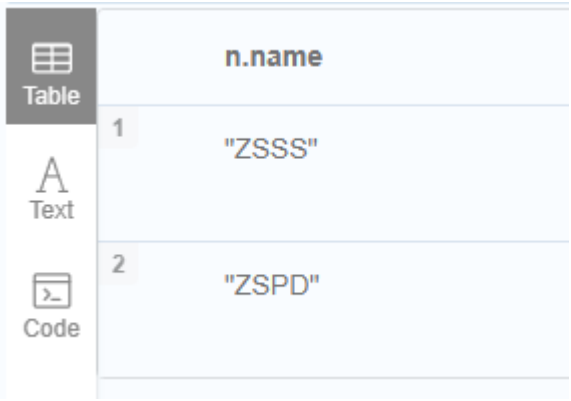
（1）CQL语句基本查询

构建完成知识图谱后，基于知识图谱现有的关系，我们可以对其进行查询问答。

可以使用CQL语句基本查询，例如查询“2022年1月3日开始发生机场相关事件的机场有哪些？”。

```
MATCH p=(n)-[r:airport2{relation:'开始时间'}]->(:airportElement{name:'2022年3月4日'}) RETURN n.name
```

输出结果如下：



（2）航行通告智能问答系统

更进一步，可以构建基于知识图谱的航行通告智能问答系统。基于已构建好的数据库，算法具体流程如下：

1：将neo4j与python连接，主要借助`py2neo`库。

2：问题意图匹配。将用户输入语句与预设问题模板语句进行匹配计算，获取匹配程度最高的模板语句类型即为问题意图。主要借助`fuzzywuzzy`模糊匹配库。
在航行通告示例问答系统中，仅设置两类问题，即：

```
start_time =['某机场开始发生机场跑道关闭事件的开始时间是什么？']
runway = ['某机场发生机场跑道关闭事件的跑道是什么？']
```

3：主要实体提取。提取问题中的机场名称，这是后续查询的基础。使用`jieba`分词的自定义词典提取，提前将机场名称导入并标注好词性`jc`，方便单独提取出机场名称。

4：cyphere语句查询。输入上续步骤获取的意图进行判断问题种类，并结合机场名称进行查询，返回目标结果。例如针对`start_time`类问题：

```
#MATCH p=(a:airportElement)-[r:airport2{relation:'开始时间'}]->(b) WHERE a.name="ZSPD" RETURN
b.name LIMIT 25
    if classification == 'start_time':
        cyphere = 'MATCH p=(a:airportElement)-[r:airport2{relation:"开始时间"}]->(b) WHERE a.na
me = "' + str(subject) + '"RETURN b.name LIMIT 25'
        object = graph.run(cyphere)
        result = []
        for item in set(object):
            result += item
```

5：回答语句匹配。将目标结果等带入回答模板中。回答模板与问题模板相对应，即：

```
start_timeRes = '{}机场发生跑道关闭的开始时间是：{}'
runwayRes = '{}机场发生跑道关闭的跑道为：{}'
```

问答与输出结果如下：

```
问题： ZSPD机场发生机场跑道关闭事件的跑道是什么？
回答： ZSPD机场发生跑道关闭的跑道为：['RWY16L/34R', 'RWY17R/35L', 'RWY17L/35R', 'RWY16R/34L']
```

```
问题： ZSOF机场开始发生机场跑道关闭事件的开始时间是什么？
回答： ZSOF机场发生跑道关闭的开始时间是：['2022年12月16日', '2022年12月23日', '2022年12月21日']
```