实践3　　基于Elasticsearch的KBQA系统实现

课程名称：　知识工程　　实验日期：　2023/6/22

班级：　人工智能 x 班　　姓名：　　学号：

一、实践要求

1、基于 Elasticsearch 搭建 KBQA 系统，完成一个问答 demo。

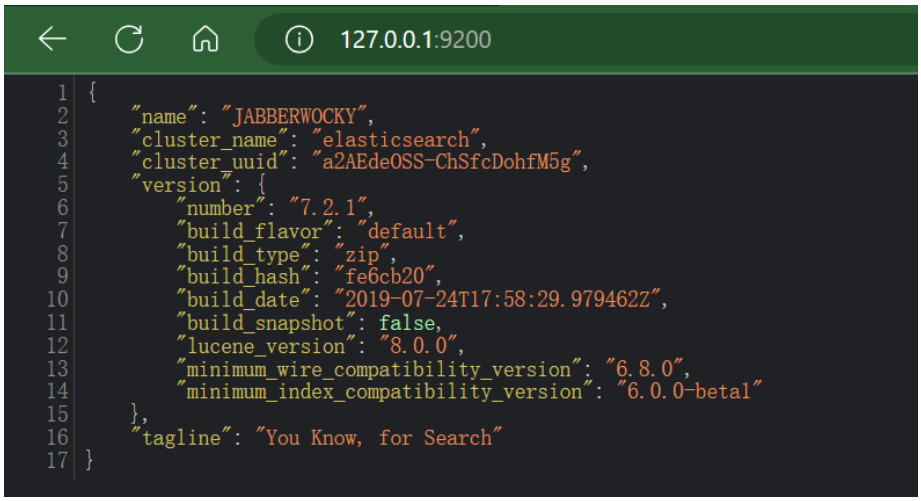
二、实践内容

1、搭建 KBQA 系统

2、解析输入的自然语言查询语句生成 Elasticsearch 查询

三、实践步骤（包括源码与过程截图）

2. 环境准备

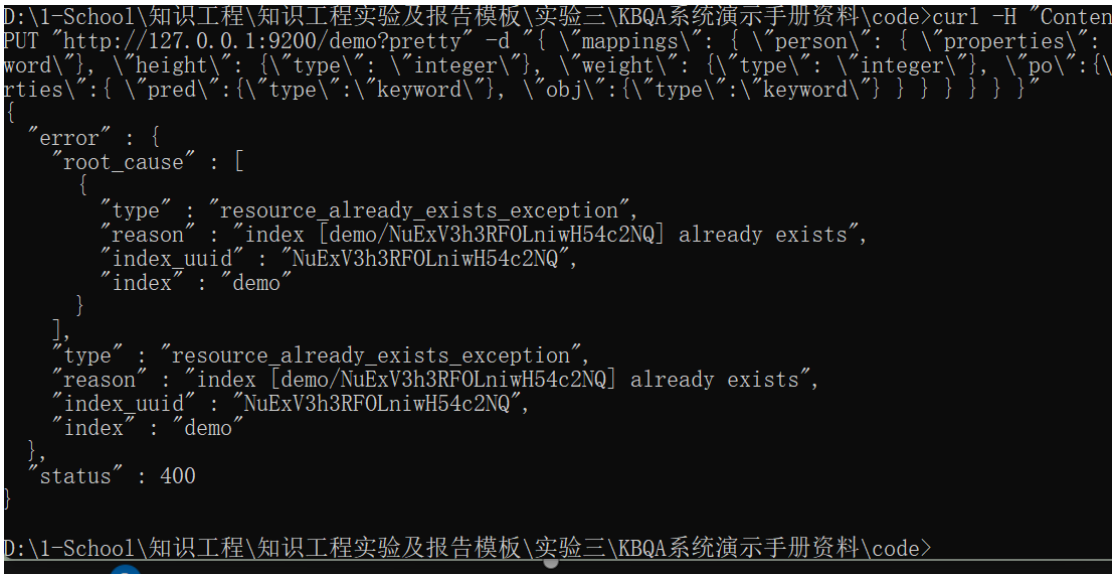


3. 数据准备

4. 导入 elasticsearch

4.2 在 elasticsearch 中新建 index 和 type

curl -H "Content-Type: application/json" -XPUT "http://127.0.0.1:9200/demo?pretty" -d "{ \"mappings\": { \"person\": { \"properties\": { \"subj\": {\"type\": \"keyword\"}, \"height\": {\"type\": \"integer\"}, \"weight\": {\"type\": \"integer\"}, \"po\":{\"type\": \"nested\", \"properties\":{ \"pred\":{\"type\":\"keyword\"}, \"obj\":{\"type\":\"keyword\"} } } } } } }"



显示已经存在

4.3 导入数据 (insert.py)

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL

61000
62000
63000
64000
65000
66000
67000
68000
69000
70000
71000
72000
73000
74000
75000
76000
77000
78000
79000
80000
81000
82000
PS D:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料>
```

建立字典：

```
76
77  if __name__ == '__main__':
78      dump_ac_attr_dict(attr_list_file, attr_out_path)
79      dump_ac_entity_dict(entity_list_file,entity_out_path)
80      print(load_ac_dict(attr_out_path))
81      print(load_attr_map(attr_list_file))
82      print(load_entity_dict(entity_list_file))
83      print(load_val_dict(val_list_file))
84

PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL

  File "d:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料\code\demo\search\build_dict.py", line 80, in <module>
    print(load_ac_dict(attr_out_path))
  File "d:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料\code\demo\search\build_dict.py", line 41, in load_ac_dict
    A = cPickle.load(open(out_path,"rb",encoding='utf-8'))
ValueError: binary mode doesn't take an encoding argument
PS D:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料>  d:; cd 'd:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料'; & 'D:\anaconda\python.exe' 'c:\Users\zq\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '55725' '--' 'd:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料\code\demo\search\build_dict.py'
<ahocorasick.Automaton object at 0x00000248FB38E030>
defaultdict(<class 'list'>, {'weight': ['weight'], '重量': ['weight'], '多重': ['weight'], '体重': ['weight'], 'WEIGHT': ['weight'], 'relatedTo': ['relatedTo'], '相关': ['relatedTo'], '有关': ['relatedTo'], 'RELATEDTO': ['relatedTo'], 'telephone': ['telephone'], '电话': ['telephone'], '号码': ['telephone'], '电话号': ['telephone'], '电话号码': ['telephone'], '手机': ['telephone'], '手机号': ['telephone'], '手机号码': ['telephone'], 'TELEPHONE': ['telephone'], 'birthDate': ['birthDate'], '出生日期': ['birthDate'], '出生时间': ['birthDate'], '生日': ['birthDate'], '时候出生': ['birthDate'], '年出生': ['birthDate'], 'BIRTHDATE': ['birthDate'], 'height': ['height'], '高度': ['height', '海拔'], '海拔': ['height', '海拔'], '多高': ['height', '海拔'], '身高': ['height'], 'HEIGHT': ['height'], 'sibling': ['sibling'], '兄弟': ['sibling'], '哥哥': ['sibling'], '姐姐': ['sibling'], '弟弟': ['sibling'], '妹妹': ['sibling'], '姐妹': ['sibling'], 'SIBLING': ['sibling'], 'workLocation': ['workLocation'], '工作地点': ['workLocation'], '在哪工作': ['workLocation'], '在哪上班': ['workLocation'], '上班地点': ['workLocation'], 'WORKLOCATION': ['workLocation'], 'children': ['children'], '子女': ['children'], '孩子': ['children'], '女儿': ['children'], '儿子': ['children'], 'CHILDREN': ['children'], '年龄': ['年龄'], '几岁': ['年龄'], '多大': ['年龄', '面积'], '代表作品': ['代表作品'], '代表作': ['
```

此时已经可以检索该知识库了，例如，按照实体名称检索：

curl -H "Content-Type: application/json" -XGET "http://127.0.0.1:9200/demo/person/_search?&pretty" -d "{ \"query\": { \"bool\": { \"must\": { \"term\": {\"subj\": \"姚明\"} } } } }"

```
C:\Users\zq>curl -H "Content-Type: application/json" -XGET "http://127.0.0.1:9200/demo/person/_search?&pretty" -d "{ \"query\": { \"bool\": { \"must\": { \"term\": {\"subj\": \"姚明\"} } } } }"
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 0,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  }
}
```

四、实践结果

6.4 根据多对(属性名，属性值)检索实体

```
print(_map_predicate('姚明的身高和体重'))
print(_entity_linking('姚明的身高和体重'))
print(_val_linking("姚明的身高和体重"))
# print(_search_single_subj('姚明'))
```

```
PS D:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料>  d:; cd 'd:\
系统演示手册资料'; & 'D:\anaconda\python.exe' 'c:\Users\zq\.vscode\extensions\ms-python.
ter/../..\debugpy\launcher' '56712' '--' 'd:\1-School\知识工程\知识工程实验及报告模板\实

['height', 'weight']
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\zq\AppData\Local\Temp\jieba.cache
Loading model cost 0.367 seconds.
Prefix dict has been built successfully.
['姚明']
{'姚明': 'spouse'}
PS D:\1-School\知识工程\知识工程实验及报告模板\实验三\KBQA系统演示手册资料>
```

四、实践心得

源代码一大堆屁用没有的冗余代码，删了一大堆终于成功了。这款分词系统对于不同的系统来说有不同的处理方式，对新手而言极不友好，还没有现成的教材，能转起来就已经是筋疲力尽了。