Lab.2 – Solr

TA. Wang Haiwen，Yuting Jia

Dept. Computer Science & Engineering,

Shanghai Jiaotong University

wanghaiwencn@foxmail.com

DDL: April. 21

Introduction

- *Previous (Lab.1)：*
  - 从http://acemap.sjtu.edu.cn/app/EE101/labs/lab1.zip下载数据，根据数据建立相关表，通过python链接数据库并将文本数据插入到数据库的表中。
- *Lab.2 ：*
  - 搭建关于学术论文搜索的solr core，设计schema，导入数据并进行查询测试。
  - 参考课程ppt。

Lab.2 - Solr

- **练习一：**
  - 创建针对Paper的solr core。
  - 配置schema（需要将每篇论文视作一个document，至少要包含PaperID, Title, Authors' ID, Authors' Name, ConferenceID, ConferenceName, Year字段，其中Title、Authors' Name, ConferenceName需要能够检索， Authors' ID, Authors' Name 两个字段需要支持多值）。
  - 注：配置完成的schema需要在report中进行详细介绍。

Lab.2 - Solr

- **练习二：**
  - 利用pysolr向该core中添加论文信息
  - 利用pysolr进行数据检索，需要分别包含对Title、Authors' Name, JournalName, ConferenceName字段的搜索测试。
  - **注：**
    - *每个字段的测试与返回结果需要在report中进行介绍。*

Notice

- 报告要求：latex编写，英文完成
- 提交方式：上传ftp
- 地址：ftp://public.sjtu.edu.cn
- 用户名：hnxxjyt
- 密码：public
- 打开方式：文件资源管理器/filezilla等软件
- 报告上传：/upload/lab2
- 报告命名格式：StudentID_Name_Lab2_Version
  - *e.g. 518030910000_Xiaoming_* Lab2_ *1*

常见问题&解答

- 1. 本次作业整体思路
- 本次作业主要分为“创建solr core并设计schema”，“使用pysolr库，从lab1搭建的mysql数据库中读取相关数据并插入solr”，“使用pysolr对插入的数据进行检索测试”三个部分。
- 其中第1和第3中的一些常见问题会在后续问题中进行解答
- 第二部分主要需要由python脚本完成，推荐整体思路如下：
  - 从Papers表中读取论文信息(包括ID、标题、年份等)，存储在python程序的内存中
  - 针对每篇论文，根据PaperID获取相关AuthorName (通过SQL中join语句即可完成)
  - 针对每篇论文，根据ConferenceID获取会议名称
  - 将每篇论文所有的信息组成一个dict，之后将所有的dict根据pysolr库提供的add方法添加到solr中
  - **因为可能反复需要执行代码，最好在代码最前面添加清空Solr的操作：*

    solr.delete(':*')*

常见问题&解答

- 1. 本次作业整体思路
- 第二部分伪代码如下：
  - solr.delete(‘*:*’)
  - cursor.execute(‘select xxx from Papers’)
  - papers = [{}, {}, {},…]
  - for paper in papers:
    - cursor.execute(‘select xxx from Authors join …’)
    - paper[‘AuthorNames’] = […]
    - cursor.execute(‘select xxx from Conferences …’)
    - paper[‘ConferenceName’] = ‘…’
  - solr.add(papers)
  - solr.commit()

常见问题&解答

- 2. schema类型如何设置
- type: 常用的有两种：string与text_en。string会按照字符串解析，solr不会做任何处理，搜索时也必须完全匹配才能搜索到。text_en则被认为是英文文本，solr会进行去除词根等操作，使得同一个词的不同时态或者单复数都能够相互匹配到。一般来说需要精确匹配的用string，需要模糊匹配的用text_en。
- stored: 如为true,则该字段会在查询结果内，false则说明该字段不会出现在查询结果中
- indexed: 如果想要对该字段进行检索，则必须设置为true
- multiValued: 是否会包含多个值（比如作者相关字段就是会有多个值）
- required: 是否要求必须有该字段，类似mysql的not null，如果设置为true，而在add时候没有该字段，则会有报错。
- 另外两个（uninvertible和docValues）一般情况下保持默认即可。更多信息可以参考https://www.jianshu.com/p/4d7c1f87c68e

常见问题&解答

- 3. solr add函数执行后找不到查询结果
- （1）一种情况时未成功commit，此时在solr网页中搜索*:*也找不到任何内容，pysolr库在近期更新中改成了默认不进行commit，需要在跑完solr.add()函数后再执行solr.commit()才行。
- （2）另一种情况是插入的doc内容不对，比如针对我们给的例子，有些同学在自己抄写时将第二篇文档名打成了“The Banana:Tasty or Dangerous?”，即冒号后无空格，此时Banana:Tasty会被认为是一个完整的词语，只搜索Banana部分是搜索不到的，必须搜索完整的Banana:Tasty才行。因此大家插入数据时候也需要注意空格不要漏掉。

常见问题&解答

- 4. 不设置schema，pysolr依然可以插入数据
- Solr的内核是lucene，而lucene实际上是‘schemeless’的，而我们设置的schema是让solr针对每个字段进行相应的处理，之后再插入到lucene中去的，因此使用pysolr插入不在schema中的程序时，目前我也不确定插入之后的数据到底是按照什么规则进行处理。因此在本次作业中要求每个字段都必须先设定schema之后再进行插入。
- 更多说明可以参考https://stackoverflow.com/a/7337249

常见问题&解答

- 5. 查询多个词时无法找到想要的结果
- 如查询Title: word1 word2 word3，此时返回值会与Title: word1的返回值完全一样，这是因为solr识别成了“Title: word1”+ word2+word3这三段。因此应当将搜索改为Title: word1+word2+word3，即多个词之间用+连接，而不是空格。