سیاوش حسن پور

921042006

Apache Solr & Apache Lucene

:Lucene

lucene چیست :

لوسین یک کتابخانه بازیابی اطلاعات(Information Retrieval) متن باز است که در سال ۱۹۹۹ توسط Doug cutting نوشته شده است. با استفاده از آن می‌توان اسناد را نمایه سازی کرد سپس به روی آنها جستجو کرد. Lucene جزو پروژه‌های Apache و تحت لیسانس ASF است. موتورهای جستجو نیز اطلاعات را نمایه گذاری می‌کنند و جستجو را بر روی نمایه‌ها انجام می‌دهند.

: Index

نمایه وسیله‌ای است برای هدایت منظم به یک متن، محتوا، مجموعه‌ای از مدارک و یا هر گونه اطلاعات ضبط شده‌ای که به شکل معمولاً الفبایی مرتب شده باشد و با استفاده از شیوه خاص و نظام مند

ارجاعی مشخص، موقعیت و محل هر مطلب را در مرحله بازیابی اطلاعات نشان دهد. واحد اصلی نمایه سند است هر سند شامل یک یا چند ستون (Field) است. ساختار نمایه تولید شده توسط لوسین نمایه معکوس است. این ساختمان داده هسته تمام موتورهای جستجو موجود است. ابتدا سند ساخته می‌شود سپس ستون‌ها ساخته و به سند اضافه می‌شوند.

من در زبان python2.7 و در سیستم عامل Ubuntu 14.04 نوشتم .

در ابتدا باید java و ant و Ivy که توضیحات نصب بصورت کامل در این سایت امده است.

سپس برای این که کتابخانه ی lucene را در python بتوانیم استفاده کنیم باید نصب کنیم (sudo apt-get install pylucene) (برای python 2)/

توجه : دانلود lucene و solr از سایت خود apache .

توضیح کد:

Import lucene که کتابخانه lucene را import میکند.

INDEX_DIR آدرس جایی که می خواهیم index ها را ذخیره کنیم.

FILES_TO_INDEX مکان داده ی ما که می خواهیم index کنیم.

.Initialize lucene and JVM : lucene.initVM()

Analyzer نحوه تحلیل ستون‌ها را نشان می‌دهد

analyzer = lucene.StandardAnalyzer(lucene.Version.LUCENE_CURRENT)

مکان ذخیره ی index را باز میکنیم

store = lucene.SimpleFSDirectory(lucene.File(INDEX_DIR))

IndexWriter برای ایجاد نمایه از شیء استفاده می‌شود. شیء IndexWriter در زمان ایجاد دو شیء Analyzer و Directory را دریافت م ی کند. شیء Directory پوشه محل ذخیره‌سازی نمایه را مشخص می‌کند شیء Analyzer .نحوه تحلیل ستون‌ها را نشان می‌دهد متغیر بولین اختیاری بوده و به طور پیش فرض false م ی‌باشد

writer = lucene.IndexWriter(store, analyzer, True, lucene.IndexWriter.MaxFieldLength.LIMITED)

بصورت کلی :

```
INDEX_DIR = "/media/siavash/000AF8440AF837EC/courcess/Information_Retrival/Tweeters_dataset/index"

# FILES_TO_INDEX = "/media/siavash/000AF8440AF837EC/courcess/Information_Retrival/Tweeters_dataset/tweets/*"
FILES_TO_INDEX = "/home/siavash/Desktop/iDesktop/courcess/information retrival/Dataset/Test/*"

# Initialize lucene and JVM
lucene.initVM()

print "lucene version is:", lucene.VERSION

# Get the analyzer
analyzer = lucene.StandardAnalyzer(lucene.Version.LUCENE_CURRENT)

# Get index storage
store = lucene.SimpleFSDirectory(lucene.File(INDEX_DIR))

# Get index writer
writer = lucene.IndexWriter(store, analyzer, True, lucene.IndexWriter.MaxFieldLength.LIMITED)
```

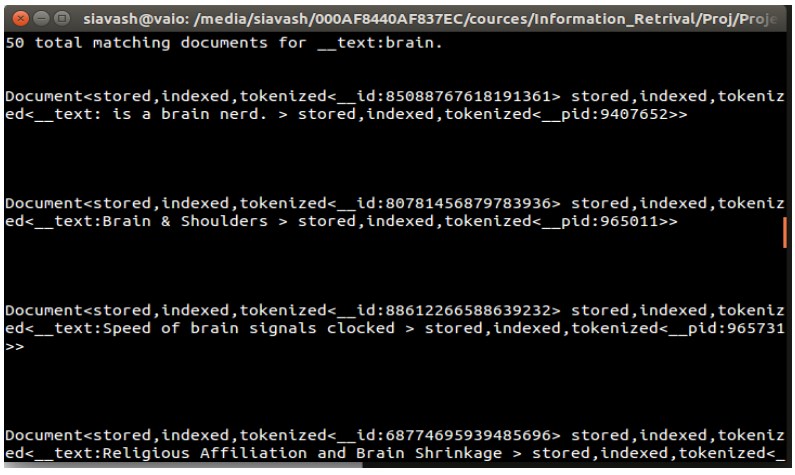
حال با توجه به گفته های بالا و با توجه به این که هر file ما شامل چندین tweet هست و در هر tweet فقط فیلد های text و id مورد نیاز هست پس باید هر فایل را split کرده و فقط text , id, pid را به عنوان فیلد به doc داده و در آخر این doc را به writer میدهیم تا index کند.

```
for files in glob.glob(FILES_TO_INDEX):
    try:
        with open(files) as f:
            t = f.read()

            # print t

            s = list(t.split("\n"))
            # print s

            b = 0

            d = []
            for j in s:
                if "Text:" in j:
                    i = j.replace('Text: ', '')
                    d.append(i)
                    b += 1
                if "ID:" in j:
                    i = j.replace('ID: ', '')
                    d.append(i)
                    b += 1


                if b == 2:
                    b = 0
                    # create a document that would we added to the index
                    doc = lucene.Document()

                    # Add a field to this document
                    # field = lucene.Field("title", "India", lucene.Field.Store.YES, lucene.Field.Index.ANALYZED)
                    name = os.path.basename(f.name)
                    field = lucene.Field("__id", d[1], lucene.Field.Store.YES, lucene.Field.Index.ANALYZED)
                    doc.add(field)
                    field = lucene.Field("__text", d[0], lucene.Field.Store.YES, lucene.Field.Index.ANALYZED)
                    doc.add(field)
                    field = lucene.Field("__pid", name, lucene.Field.Store.YES, lucene.Field.Index.ANALYZED)
                    doc.add(field)

                    # Add the document to the index
                    writer.addDocument(doc)
                    d = []
                    # print name
    except():
        # print "Failed in indexDocs:", e
        print "Oops!!!..."
        continue
```

حال برای تست کردن این که درست index میکند یا نه در همان lucene یک search نوشته ام:

```
from apt import cache
from ibus import exception

FIELD_CONTENTS = "__text"
FIELD_PATH = "/media/siavash/000AF8440AF837EC/cources/Information_Retrival/Proj/Datasets/Tweeters_dataset/Test(100)/*"

QUERY_STRING = "brain"

STORE_DIR = "/media/siavash/000AF8440AF837EC/cources/Information_Retrival/Proj/Datasets/Tweeters_dataset/index/"

if __name__ == '__main__':
    try:
        lucene.initVM()
        print 'lucene', lucene.VERSION

        # Get handle to index directory
        directory = lucene.SimpleFSDirectory(lucene.File(STORE_DIR))

        # Creates a searcher searching the provided index.
        ireader  = lucene.IndexReader.open(directory, True)

        # Implements search over a single IndexReader.
        # Use a single instance and use it across queries
        # to improve performance.
        searcher = lucene.IndexSearcher(ireader)

        # Get the analyzer
        analyzer = lucene.StandardAnalyzer(lucene.Version.LUCENE_CURRENT)

        # Constructs a query parser. We specify what field to search into.
        queryParser = lucene.QueryParser(lucene.Version.LUCENE_CURRENT, FIELD_CONTENTS, analyzer)

        # Create the query
        query = queryParser.parse(QUERY_STRING)

        # Run the query and get top 50 results
        topDocs = searcher.search(query, 50)

        # Get top hits
        scoreDocs = topDocs.scoreDocs
        # print scoreDocs
        print "%s total matching documents for %s." % (len(scoreDocs), query)

        for scoreDoc in scoreDocs:
            doc = searcher.doc(scoreDoc.doc)
            print "\n"
            print doc
            print "\n"
            # print doc.get(FIELD_PATH)
    except exception:
        print exception
```

که خروجی برای q = brain بصورت زیر است :



Solr :

Apache solr یک موتور جستوجو open source است. که ما داده ها را یا توسط خود solr یا با lucene , index کرده و میتوانیم با انواع query سرچ کنیم.

بعد از دانلود solr و extract کردن ان مراحل زیر را برای راه اندازی solr به ترتیب انجام می دهیم:

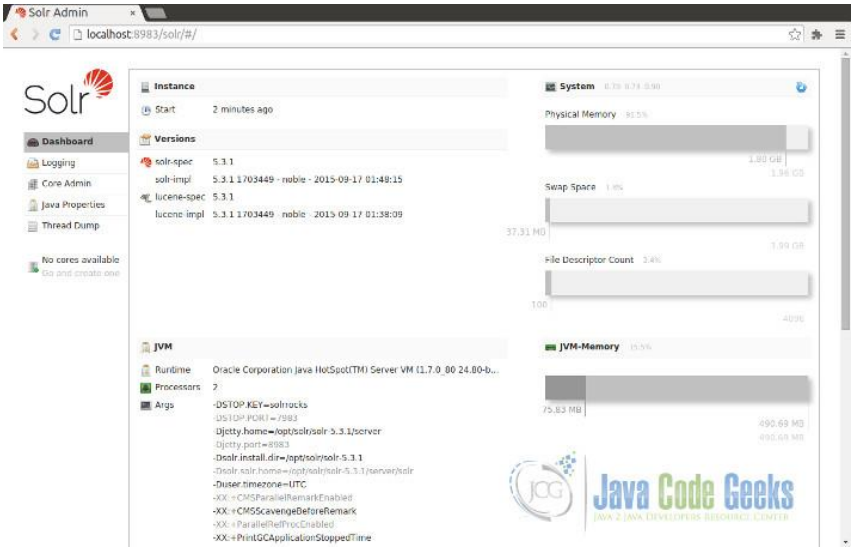
1. ابتدا جاوا 8 نصب را نصب میکنیم
   1. sudo apt-get install python-software-properties
   2. sudo add-apt-repository ppa:webupd8team/java
   3. sudo apt-get update
   4. sudo apt-get install oracle-java8-installer

2. نصب solr :
   1. cd ~
   2. wget http://apache.mirror1.spango.com/lucene/solr/5.2.1/solr-5.2.1.tgz
   3. tar xzf solr-5.2.1.tgz solr-5.2.1/bin/install_solr_service.sh --strip-components=2
   4. sudo bash ./install_solr_service.sh solr-5.2.1.tgz
   5. حال به پوشه ی bin رفته و دستور sudo service solr status را اجرا می کنیم

3. ایجاد core :
   - sudo su - solr -c "/opt/solr/bin/solr create -c core_name -n data_driven_schema_configs"

4. تغییر managed_schema :
   - ابتدا به فایل coreایجاد شده میرویم (/var/solr/core_name) سپس فایل در managed_schema تغییرات زیر را در قسمت <uniqueKey>id</uniqueKey> اعمال میکنیم:
     - <field name="__id" type="text_general" indexed="true" stored="true"/>
     - <field name="__text" type="text_general" indexed="true" stored="true"/>
     - <field name="__pid" type="text_general" indexed="true" stored="true"/>
   - توجه نام ها بر اساس نام فیلد ها در python تعیین میشود.
   - حال باید solr , restart شود : sudo service solr restart

5. حال داده های index شده در lucene را در /var/solr/core_name/data/index/ کپی کرده و دوباره restart می کنیم.



مراحل config و راه اندازی solr به پایان رسید و حالا میتوانیم سلر را در مرورگر با آدرسه localhost:8983 مشاهده کرد
که core ساخته شده در سمت چپ دیده میشود که وقتی بر روی آن کلیک کرده تعداد index ها دیده میشود .
Query را نیز میتوان در قسمت url بجای (q=hello) داد.
http://localhost:8983/solr/query?debug=query&q=hello

با تشکر

حسن پور