Assignment 3

Spark-submit command

```
spark-submit --class "CaseIndex" --packages org.scalaj:scalaj-
http_2.11:2.4.2,org.scalatestplus.play:scalatestplus-play_2.11:4.0.3 --master local[2] JAR_FILE
FULL_PATH_OF_DIRECTORY_WITH_CASE_FILES
```

Index Design

My mapping has 8 fields:

1. Id: filename of case report
2. Name: The title of the case
3. URL: the original source of the legal report
4. Catchphrase: The summary of the case
5. Sentence: The list of sentences contained in the legal case report
6. Person: The list of people found in the legal case report
7. Location: The list of locations found in the legal case report
8. Organization: The list of organizations found in the legal case report

Solution Implementation

1. First, create an index and mapping as mentioned previously by HTTP request.
2. Then, get directory path from argument. To process each file, starting with getting information from each file.

```
val name = (xml \ "name").text.filter(_ >= ' ')
val url =  (xml \ "AustLII").text.filter(_ >= ' ')
val catchphrases = (xml \ "catchphrases" \ "catchphrase")
val catchphrase = new StringBuilder("");
catchphrases.foreach(_catchphrase=>{
        catchphrase ++=  _catchphrase.text + " "
})
val sentences = (xml \ "sentences" \ "sentence")
val sentence_list = new ListBuffer[String]()
sentences.foreach(sentence => {
        sentence_list += sentence.text.replace("\"","\\\"")
})
```

3. Next, sending each sentence to CoreNLP server to find name entities. Parsing the response to JSON object and extracting “person”, “location” and “organization” entities from response.

```
sentence_list.foreach(e_sentence=>{
NLP_result =
Http("""http://localhost:9000/?properties=%7B'annotators':'ner','ner.applyFineGrained':'false','outputFormat':'json
'%7D""").postData(e_sentence).method("POST").header("Content-Type",
"application/json").option(HttpOptions.connTimeout(10000)).option(HttpOptions.readTimeout(10000)).asString.body
val NLP_json: JsValue = Json.parse(NLP_result)
val tokens = NLP_json \\ " tokens "
        tokens.foreach(token=>{
                val text = token \\ "word"
                val ner = token \\ "ner"
                var idx = 0
                for(idx <- 0 until text.length){
                        if(ner(idx).toString == "\"PERSON\""){
                        people += text(idx).toString
                } else if(ner(idx).toString == "\"LOCATION\""){
                        locations += text(idx).toString
                } else if(ner(idx).toString == "\"ORGANIZATION\""){
                        organizations += text(idx).toString
                }}})})
```

4. Finally, create a new document and sending HTTP request to Elasticsearch server.

```
val post_Data =
s"""{"id":"${filename}","name":"${name}","url":"${url}","catchphrase":"${catchphrase.toString.filter(_ >= '
')}","sentence":${new_sentence_list},"person":${people_list},"location":${locations_list},"organization":${organiza
tions_list}}"""

val new_document_result =
Http("http://localhost:9200/legal_idx/cases/"+filename+"?pretty").postData(post_Data).method("PUT").header("Content
-Type",
"application/json")").option(HttpOptions.connTimeout(60000)).option(HttpOptions.readTimeout(60000)).asString
```

Queries example

1. General term search

Command: `curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=(criminal%20AND%20law)"`
Result:


```
{
  "took":52,
  "timed_out":false,
  "_shards":{
    "total":5,
    "successful":5,
    "skipped":0,
    "failed":0
  },
  "hits":{
    "total":2,
    "max_score":1.0326371,
    "hits":[ ]
  }
}
```


2. Entity search

Command: `curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=person:John"`
Result:


```
{
  "took":3,
  "timed_out":false,
  "_shards":{
    "total":5,
    "successful":5,
    "skipped":0,
    "failed":0
  },
  "hits":{
    "total":2,
    "max_score":0.6548752,
    "hits":[ ]
  }
}
```


Command: `curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=location:Melbourne"`
Result:


```
{
  "took":3,
  "timed_out":false,
  "_shards":{
    "total":5,
    "successful":5,
    "skipped":0,
    "failed":0
  },
  "hits":{
    "total":1,
    "max_score":0.2876821,
    "hits":[ ]
  }
}
```


Command: `curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=organization:State%20Bank%20of%20New%20South%20Wales"`
Result:


```
{
  "took":20,
  "timed_out":false,
  "_shards":{
    "total":5,
    "successful":5,
    "skipped":0,
    "failed":0
  },
  "hits":{
    "total":5,
    "max_score":3.1138277,
    "hits":[ ]
  }
}
```