Assignment 3 solution

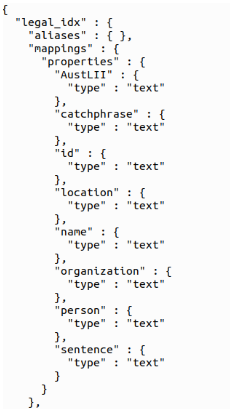
Student id: z5153042

Student name: Hang Zhang

Spark-submit command:

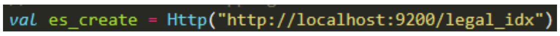
spark-submit --packages org.scalaj:scalaj-http_2.11:2.4.2,org.json:json:20180813 --class "CaseIndex" --master local[2] JAR_FILE FULL_PATH_OF_DIRECTORY_WITH_CASE_FILES

Index and mapping design:



```
{
  "legal_idx" : {
    "aliases" : { },
    "mappings" : {
      "properties" : {
        "AustLII" : {
          "type" : "text"
        },
        "catchphrase" : {
          "type" : "text"
        },
        "id" : {
          "type" : "text"
        },
        "location" : {
          "type" : "text"
        },
        "name" : {
          "type" : "text"
        },
        "organization" : {
          "type" : "text"
        },
        "person" : {
          "type" : "text"
        },
        "sentence" : {
          "type" : "text"
        }
      }
    },
```

Index design:



```
val es_create = Http("http://localhost:9200/legal_idx")
```

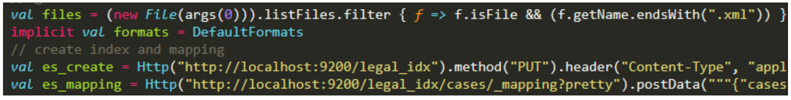
Mapping:

Cases:{ properties:{

1. Id: file name
2. Name: The title of the case
3. AustLII: the url of the case
4. Catchphrase: the summary of the case
5. Sentence: the sentences in the case
6. Person: the list of persons in the case
7. Location: the list of locations in the case
8. Organization: the list of organization in the case }

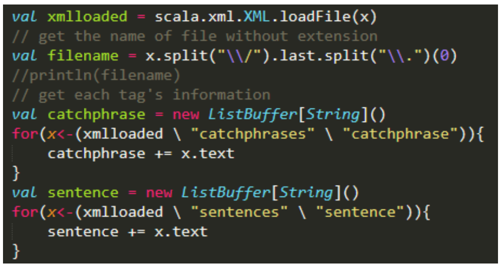
Solution explanation:

1. Get the file names, create the index and mapping in elasticsearch.

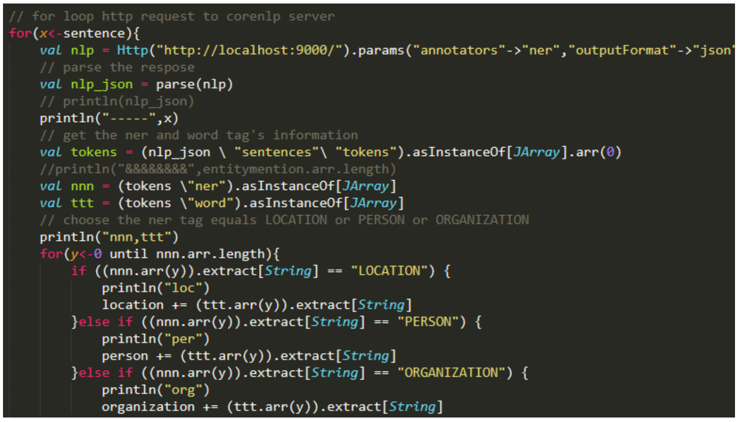


```
val files = (new File(args(0))).listFiles.filter { f => f.isFile && (f.getName.endsWith(".xml")) }
implicit val formats = DefaultFormats
// create index and mapping
val es_create = Http("http://localhost:9200/legal_idx").method("PUT").header("Content-Type", "appl
val es_mapping = Http("http://localhost:9200/legal_idx/cases/_mapping?pretty").postData("""{"cases
```

2. Use the package scala.xml.XML to load xml file and extract information of each tags (sentence and catchphrase). And then send information to corenlp server to get the information from respose of “ner” and “word” tag.

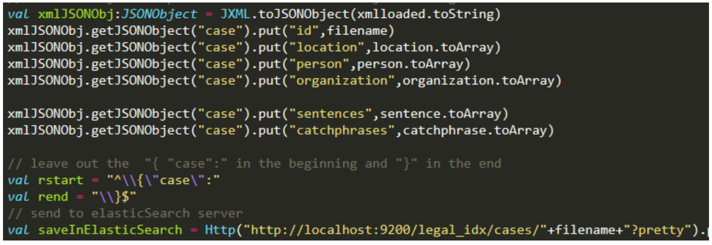


```
val xmlloaded = scala.xml.XML.loadFile(x)
// get the name of file without extension
val filename = x.split("\\/").last.split("\\.")(0)
//println(filename)
// get each tag's information
val catchphrase = new ListBuffer[String]()
for(x<-(xmlloaded \ "catchphrases" \ "catchphrase")){
    catchphrase += x.text
}
val sentence = new ListBuffer[String]()
for(x<-(xmlloaded \ "sentences" \ "sentence")){
    sentence += x.text
}
```



```
// for loop http request to corenlp server
for(x<-sentence){
    val nlp = Http("http://localhost:9000/").params("annotators"->"ner","outputFormat"->"json"
    // parse the respose
    val nlp_json = parse(nlp)
    // println(nlp_json)
    println("-----",x)
    // get the ner and word tag's information
    val tokens = (nlp_json \ "sentences"\ "tokens").asInstanceOf[JArray].arr(0)
    //println("&&&&&&&&",entitymention.arr.length)
    val nnn = (tokens \"ner").asInstanceOf[JArray]
    val ttt = (tokens \"word").asInstanceOf[JArray]
    // choose the ner tag equals LOCATION or PERSON or ORGANIZATION
    println("nnn,ttt")
    for(y<-0 until nnn.arr.length){
        if ((nnn.arr(y)).extract[String] == "LOCATION") {
            println("loc")
            location += (ttt.arr(y)).extract[String]
        }else if ((nnn.arr(y)).extract[String] == "PERSON") {
            println("per")
            person += (ttt.arr(y)).extract[String]
        }else if ((nnn.arr(y)).extract[String] == "ORGANIZATION") {
            println("org")
            organization += (ttt.arr(y)).extract[String]
```

3. Form the string to be passed to elasticsearch by put method of JSONObject and send it.



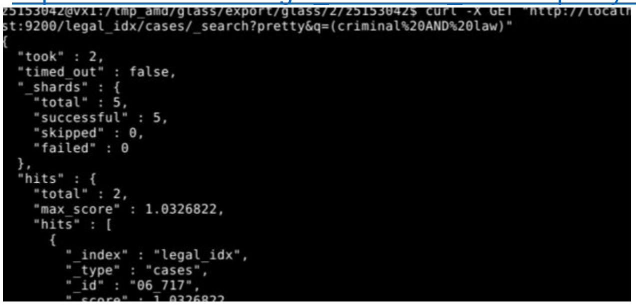
```
val xmlJSONObj:JSONObject = JXML.toJSONObject(xmlloaded.toString)
xmlJSONObj.getJSONObject("case").put("id",filename)
xmlJSONObj.getJSONObject("case").put("location",location.toArray)
xmlJSONObj.getJSONObject("case").put("person",person.toArray)
xmlJSONObj.getJSONObject("case").put("organization",organization.toArray)

xmlJSONObj.getJSONObject("case").put("sentences",sentence.toArray)
xmlJSONObj.getJSONObject("case").put("catchphrases",catchphrase.toArray)

// leave out the  "{ "case":" in the beginning and "}" in the end
val rstart = "^\\{\"case\":"
val rend = "\\}$"
// send to elasticSearch server
val saveInElasticSearch = Http("http://localhost:9200/legal_idx/cases/"+filename+"?pretty").
```

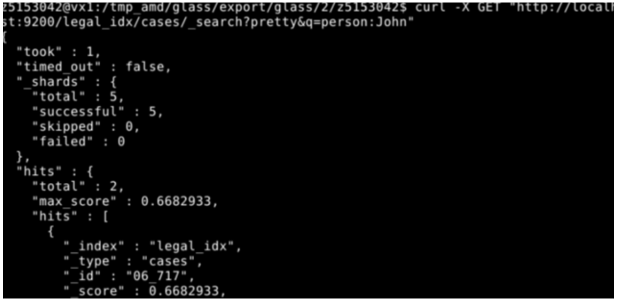
Queries example:

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



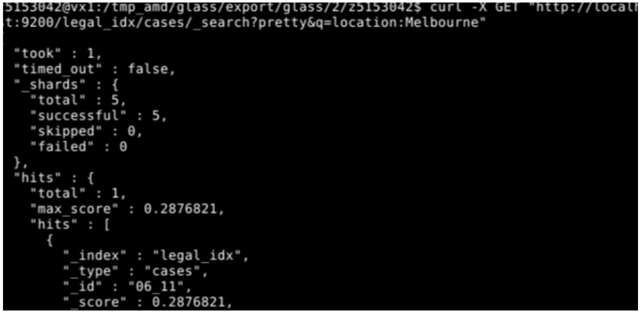
```
z5153042@vx1:/tmp_amd/glass/export/glass/2/z5153042$ curl -X GET "http://localh
st:9200/legal_idx/cases/_search?pretty&q=(criminal%20AND%20law)"
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 1.0326822,
    "hits" : [
      {
        "_index" : "legal_idx",
        "_type" : "cases",
        "_id" : "06_717",
        "_score" : 1.0326822
```

2. Entity search:
   Command: curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=person:John"



```
z5153042@vx1:/tmp_amd/glass/export/glass/2/z5153042$ curl -X GET "http://localh
st:9200/legal_idx/cases/_search?pretty&q=person:John"
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 0.6682933,
    "hits" : [
      {
        "_index" : "legal_idx",
        "_type" : "cases",
        "_id" : "06_717",
        "_score" : 0.6682933,
```

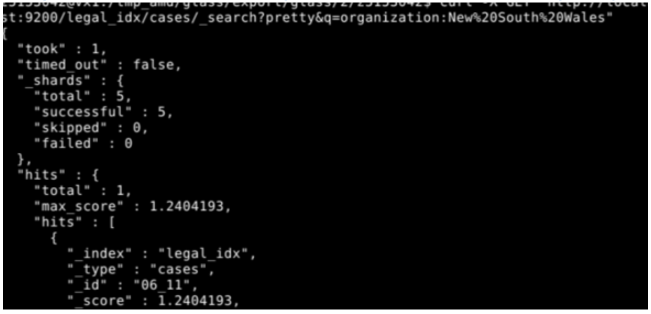
   Command: curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=location:Melbourne"



```
5153042@vx1:/tmp_amd/glass/export/glass/2/z5153042$ curl -X GET "http://localh
t:9200/legal_idx/cases/_search?pretty&q=location:Melbourne"

 "took" : 1,
 "timed_out" : false,
 "_shards" : {
   "total" : 5,
   "successful" : 5,
   "skipped" : 0,
   "failed" : 0
 },
 "hits" : {
   "total" : 1,
   "max_score" : 0.2876821,
   "hits" : [
     {
       "_index" : "legal_idx",
       "_type" : "cases",
       "_id" : "06_11",
       "_score" : 0.2876821,
```

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



```
z5153042@vx1:/tmp_amd/glass/export/glass/2/z5153042$ curl -X GET "http://local
st:9200/legal_idx/cases/_search?pretty&q=organization:New%20South%20Wales"
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.2404193,
    "hits" : [
      {
        "_index" : "legal_idx",
        "_type" : "cases",
        "_id" : "06_11",
        "_score" : 1.2404193,
```