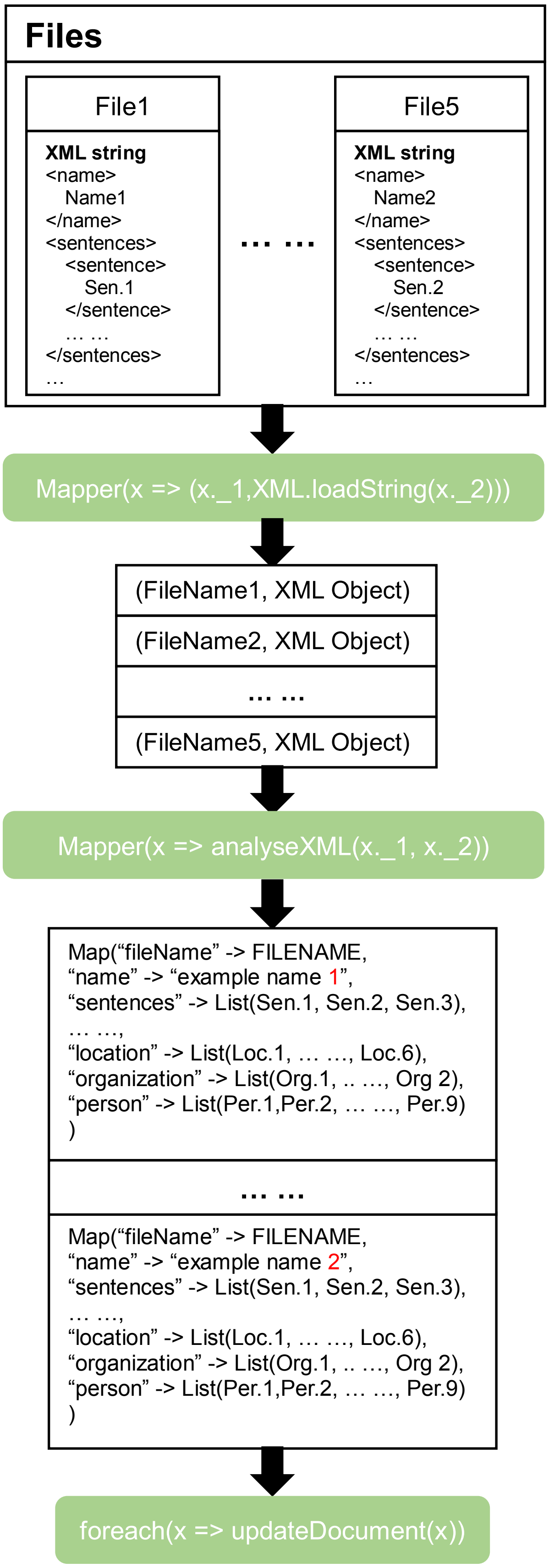
Assignment 3 Report

Implement



Files
File1
XML string
<name>
Name1
</name>
<sentences>
<sentence>
Sen.1
</sentence>
… …
</sentences>
…
… …
File5
XML string
<name>
Name2
</name>
<sentences>
<sentence>
Sen.2
</sentence>
… …
</sentences>
…
Mapper(x => (x._1,XML.loadString(x._2)))
(FileName1, XML Object)
(FileName2, XML Object)
… …
(FileName5, XML Object)
Mapper(x => analyseXML(x._1, x._2))
Map("fileName" -> FILENAME,
"name" -> "example name 1",
"sentences" -> List(Sen.1, Sen.2, Sen.3),
… …,
"location" -> List(Loc.1, … …, Loc.6),
"organization" -> List(Org.1, .. …, Org 2),
"person" -> List(Per.1,Per.2, … …, Per.9)
)
… …
Map("fileName" -> FILENAME,
"name" -> "example name 2",
"sentences" -> List(Sen.1, Sen.2, Sen.3),
… …,
"location" -> List(Loc.1, … …, Loc.6),
"organization" -> List(Org.1, .. …, Org 2),
"person" -> List(Per.1,Per.2, … …, Per.9)
)
foreach(x => updateDocument(x))

In this assignment, I use scalaj-http for handling HTTP, scala-xml for handling XML, and JSON4S for handling JSON.

Run 'spark-submit' with '--packages --packages "org.scalaj:scalaj-http_2.11:2.4.2","org.json4s:json4s-native_2.11:3.5.3"'

Firstly, get the RDD of all files via method named 'wholeTextFiles'

Secondly, convert every file from string to XML object.

Then, we get an array contains key-value pairs whose value is the XML object.

Thirdly, analyse each XML object via analyseXML method. This method will split XML object to XML elements and send each of them to NLP server for getting named entity recognition.

After that, we get several map objects, containing filename, sentences and NERs, which will be sent to Elasticsearch server.

Finally, the Map object will be converted to JSON string in updateDocument and be sent to Elasticsearch server.

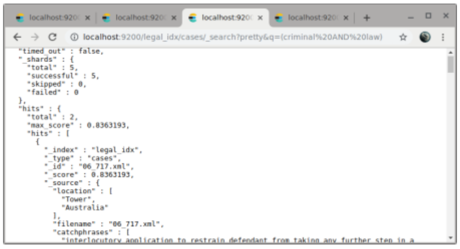
Index design

filename	Text	Filename, this document from
name	Text	Name of case
AustLII	Text	URL of this case
catchphrases	Text(List)	Summarize of case stored in text list
sentences	Text(List)	Sentences contained in the legal case report and stored in text list
person	Text(List)	Store person NER analysed from XML
location	Text(List)	Store location NER analysed from XML
organization	Text(List)	Store organization NER analysed from XML

Example queries:

Query based on general terms:

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

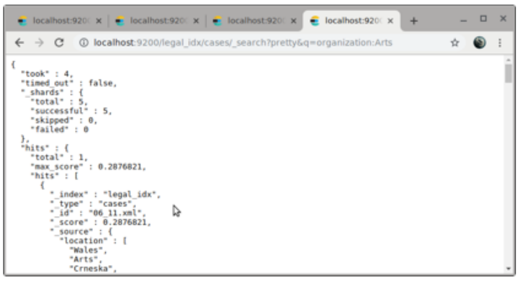


localhost:9200/legal_idx/cases/_search?pretty&q=(criminal%20AND%20law)

```
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 0.8363193,
    "hits" : [
      {
        "_index" : "legal_idx",
        "_type" : "cases",
        "_id" : "06_717.xml",
        "_score" : 0.8363193,
        "_source" : {
          "location" : [
            "Tower",
            "Australia"
          ],
          "filename" : "06_717.xml",
          "catchphrases" : [
            "interlocutory application to restrain defendant from taking any further step in a
```

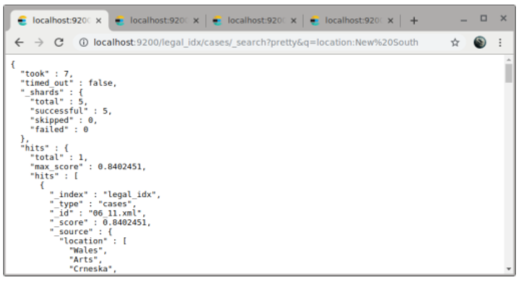
Queries based on entity type:

```
curl -X GET \
"http://localhost:9200/legal_idx/cases/_search?pretty&q=location:New%20South%20Walse"
curl -X GET \
"http://localhost:9200/legal_idx/cases/_search?pretty&q=person:John"
curl -X GET \
"http://localhost:9200/legal_idx/cases/_search?pretty&q=organization:Arts"
```



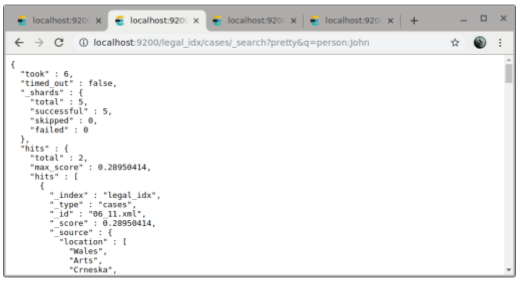
localhost:9200/legal_idx/cases/_search?pretty&q=organization:Arts

```
{
  "took" : 4,
  "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.xml",
        "_score" : 0.2876821,
        "_source" : {
          "location" : [
            "Wales",
            "Arts",
            "Crneska",
```



localhost:9200/legal_idx/cases/_search?pretty&q=location:New%20South

```
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 0.8402451,
    "hits" : [
      {
        "_index" : "legal_idx",
        "_type" : "cases",
        "_id" : "06_11.xml",
        "_score" : 0.8402451,
        "_source" : {
          "location" : [
            "Wales",
            "Arts",
            "Crneska",
```



localhost:9200/legal_idx/cases/_search?pretty&q=person:John

```
{
  "took" : 6,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 0.28950414,
    "hits" : [
      {
        "_index" : "legal_idx",
        "_type" : "cases",
        "_id" : "06_11.xml",
        "_score" : 0.28950414,
        "_source" : {
          "location" : [
            "Wales",
            "Arts",
            "Crneska",
```