1. Generate the output (changes or transformations in the data)

   manually when the following tasks are applied on the input text. Show your output in details.

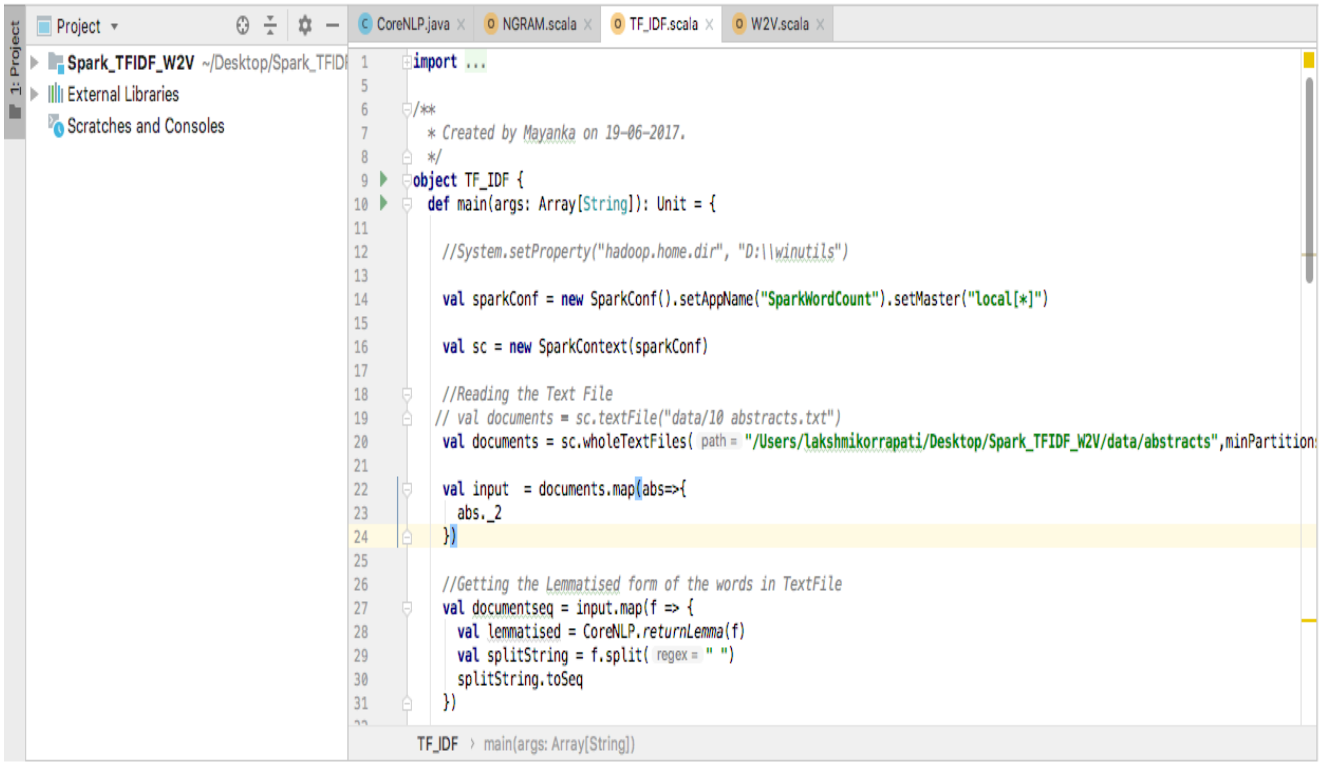
   Input: 5 Abstracts saved in separate files

Tasks:

a. Find out the top TF-IDF words for the above input.
b. Find out the top TF-IDF words for the lemmatized input
c. Find out the top TF-IDF words for the n-gram based input.
2. Write a simple spark program to read a dataset and find the W2V Synonyms for the Top TF-IDF Words
a. Try without NLP
b. Try with Lemmatization
c. Try with NGrams

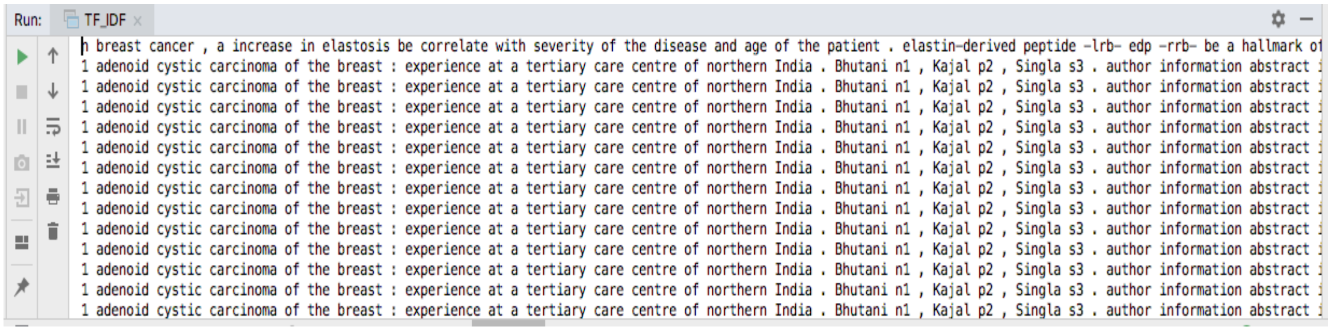
   Compare the results from (a) , (b) and (c)

   TF-IDF:



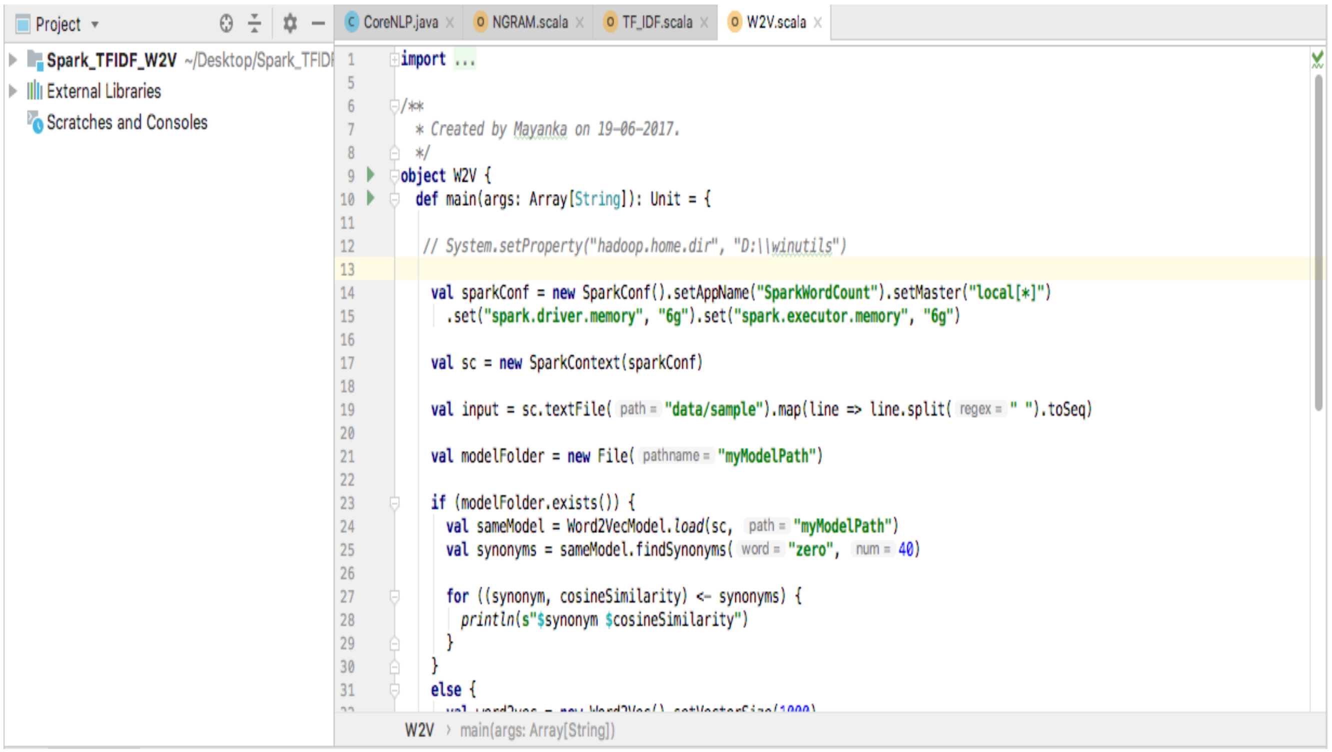
```
import ...

/**
  * Created by Mayanka on 19-06-2017.
  */
object TF_IDF {
  def main(args: Array[String]): Unit = {

    //System.setProperty("hadoop.home.dir", "D:\\winutils")

    val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")

    val sc = new SparkContext(sparkConf)

    //Reading the Text File
   // val documents = sc.textFile("data/10 abstracts.txt")
    val documents = sc.wholeTextFiles( path = "/Users/lakshmikorrapati/Desktop/Spark_TFIDF_W2V/data/abstracts",minPartition

    val input  = documents.map(abs=>{
      abs._2
    })

    //Getting the Lemmatised form of the words in TextFile
    val documentseq = input.map(f => {
      val lemmatised = CoreNLP.returnLemma(f)
      val splitString = f.split( regex = " ")
      splitString.toSeq
    })
```

Output:



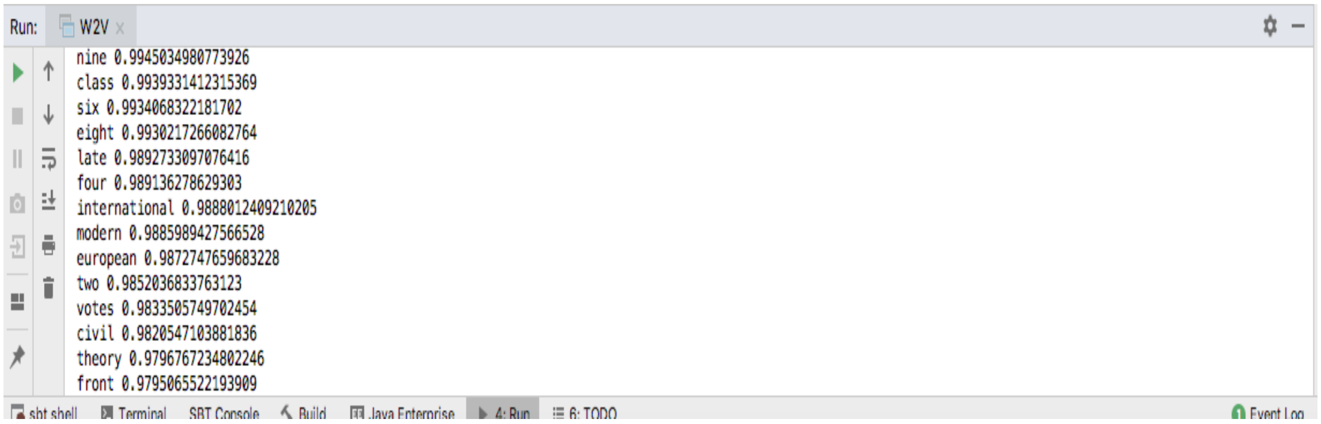
```
h breast cancer , a increase in elastosis be correlate with severity of the disease and age of the patient . elastin-derived peptide -lrb- edp -rrb- be a hallmark of
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
1 adenoid cystic carcinoma of the breast : experience at a tertiary care centre of northern India . Bhutani n1 , Kajal p2 , Singla s3 . author information abstract i
```

W2V for TF-IDF words:



```scala
import ...

/**
 * Created by Mayanka on 19-06-2017.
 */
object W2V {
  def main(args: Array[String]): Unit = {

   // System.setProperty("hadoop.home.dir", "D:\\winutils")

    val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
      .set("spark.driver.memory", "6g").set("spark.executor.memory", "6g")

    val sc = new SparkContext(sparkConf)

    val input = sc.textFile("data/sample").map(line => line.split(" ").toSeq)

    val modelFolder = new File("myModelPath")

    if (modelFolder.exists()) {
      val sameModel = Word2VecModel.load(sc, "myModelPath")
      val synonyms = sameModel.findSynonyms("zero", 40)

      for ((synonym, cosineSimilarity) <- synonyms) {
        println(s"$synonym $cosineSimilarity")
      }
    }
    else {
```

Output:



```
nine 0.9945034980773926
class 0.9939331412315369
six 0.9934068322181702
eight 0.9930217266082764
late 0.9892733097076416
four 0.989136278629303
international 0.9888012409210205
modern 0.9885989427566528
european 0.9872747659683228
two 0.9852036833763123
votes 0.9833505749702454
civil 0.9820547103881836
theory 0.9796767234802246
front 0.9795065522193909
```