# Source code in groovy

**package** AI\_lab  
  
*/\*\*  
 \* Created by saroj on 5/31/2016.  
 \*/***class** Probablity\_Calc {  
  
 *//function to count how many times two words have appeared together* **public int** countPairMatch(String first, String second,List wordsList)  
 {  
 **int** count=0;  
 **for**(**int** i=0;i<(wordsList.size()-2);i++)  
 {  
 **if**((wordsList.get(i).toString()==first)&&(wordsList.get(i+1).toString()==second))  
 {  
 count++;  
 }  
 }  
 **return** count;  
 }  
 *//Method to find the conditional probablity of two words* **public float** conditionalProbablity(String first, String second, List wordList, Map wordCount)  
 {  
 **int** pairCount=countPairMatch(second, first, wordList)  
 **int** countFirstLetter=wordCount.get(second)  
 **return** (pairCount/countFirstLetter)  
 }  
  
 **public static void** main(String[] args) {  
 *//Instance of the class to access the methods* Probablity\_Calc probablity\_calc=**new** Probablity\_Calc();  
  
 *//Importer the text form the file to string* String textFromFile=**new** File(**"D:\\shakespeare.txt"**).text  
 *//lowercased every letter in the string that is obtained from the file* textFromFile=textFromFile.toLowerCase()  
 *//parsed the words on the basis of following characters (' .;,:?!) and newline and stored it in list* **def** wordslist=textFromFile.tokenize(**' .;,:?!\n'**)  
 *//new map declared to record the words and their respective appearance in the text* **def** wordsAndRepetitionMap=[:]  
 **def** total\_words=wordslist.size()  
 **def** count;  
 *//iterate through the list that contains every words in the file* wordslist.forEach(){  
 *//initially the count is one* count=1  
 *//If the map already contains the key then it enters the if statement and increases its current count by 1* **if**(wordsAndRepetitionMap.containsKey(it)==**true**)  
 {  
 count=wordsAndRepetitionMap[it]+1  
 }  
 wordsAndRepetitionMap.put(it,count)  
 }  
 */\*1. Calculate the relative frequency (probability estimate) of the words:  
(a) “the" (b) “become" (d) “brave" (e) “treason"\*/*

*//calculate the probablity estimate of word "the"* **def** probablityOfThe=(wordsAndRepetitionMap.get(**'the'**)/wordslist.size())

println(**"the probablity estimate if word 'the' is "**+probablityOfThe)

*//calculate the probablity estimate of word "become"***def** probablityOfBecome=(wordsAndRepetitionMap.get(**'become'**)/wordslist.size())

println(**"the probablity estimate if word 'become' is "**+probablityOfBecome)

*//calculate the probablity estimate of word "brave"*

**def** probablityOfBrave=(wordsAndRepetitionMap.get(**'brave'**)/wordslist.size())

println(**"the probablity estimate if word 'brave' is "**+probablityOfBrave)

*//calculate the probablity estimate of word "treason"***def** probablityOfTreason=(wordsAndRepetitionMap.get(**'treason'**)/wordslist.size())

println(**"the probablity estimate if word 'treason' is "**+probablityOfTreason)  
  
  
 */\* 2. Calculate the following word conditional probabilities:  
 (a) P(court | The) (b) P(word | his) (c) P(qualities | rare) (d) P(men | young)  
 [Read P(B | A) as “the probability with which word B follows word A". Note: P(B | A) = count(A;B)  
 | count(A) ]\*/*

*// (a) P(court | The)*

*//We calculate the conditional probability using conditionalProbablity function* println(**"the probablity of appearing 'court' given 'the' has appeared is "**+probablity\_calc.conditionalProbablity(**'court'**,**'the'**,wordslist,wordsAndRepetitionMap))

*//b) P(word | his)* println(**"the probablity of appearing 'word' given 'his' has appeared is "**+probablity\_calc.conditionalProbablity(**'word'**,**'his'**,wordslist,wordsAndRepetitionMap))

*//(c) P(qualities | rare)* println(**"the probablity of appearing 'qualities' given 'rare' has appeared is "**+probablity\_calc.conditionalProbablity(**'qualities'**,**'rare'**,wordslist,wordsAndRepetitionMap))

*//(d) P(men | young)* println(**"the probablity of appearing 'men' given 'young' has appeared is "**+probablity\_calc.conditionalProbablity(**'men'**,**'young'**,wordslist,wordsAndRepetitionMap))  
  
  
*/\* 3. Calculate the probability:  
 (a) P(have, sent) (b) P(will, look, upon) (c) P(I, am, no, baby) (d) P(wherefore, art, thou, Romeo)  
 Hint à use the chain rule (multiplication rule):\*/*

*//We calculate the conditional probability of two words using conditionalProbablity function*

*//(a) P(have, sent)* println(**"the probablity of appearing 'have' given 'sent' has appeared is "**+probablity\_calc.conditionalProbablity(**'have'**,**'sent'**,wordslist,wordsAndRepetitionMap))

*//P(will, look, upon)* println(**"the probablity of appearing 'will' , 'look' and 'upon' together is "**+((wordsAndRepetitionMap.get(**'will'**)/total\_words)\*(probablity\_calc.conditionalProbablity(**'look'**,**'will'**,wordslist,wordsAndRepetitionMap)/total\_words)\*(probablity\_calc.conditionalProbablity(**'upon'**,**'look'**,wordslist,wordsAndRepetitionMap)/total\_words)))

*//(c) P(I, am, no, baby)* println(**"the probablity of appearing 'i' , 'am', 'no' and 'baby' together is "**+((wordsAndRepetitionMap.get(**'i'**)/total\_words)\*(probablity\_calc.conditionalProbablity(**'am'**,**'i'**,wordslist,wordsAndRepetitionMap)/total\_words)\*(probablity\_calc.conditionalProbablity(**'no'**,**'am'**,wordslist,wordsAndRepetitionMap)/total\_words)\*(probablity\_calc.conditionalProbablity(**'baby'**,**'no'**,wordslist,wordsAndRepetitionMap)/total\_words)))

*//(d) P(wherefore, art, thou, Romeo)* println(**"the probablity of appearing 'wherefore' , 'art', 'thou' and 'Romeo together is "**+((wordsAndRepetitionMap.get(**'wherefore'**)/total\_words)\*(probablity\_calc.conditionalProbablity(**'art'**,**'wherefore'**,wordslist,wordsAndRepetitionMap)/total\_words)\*(probablity\_calc.conditionalProbablity(**'thou'**,**'art'**,wordslist,wordsAndRepetitionMap)/total\_words)\*(probablity\_calc.conditionalProbablity(**'romeo'**,**'thou'**,wordslist,wordsAndRepetitionMap)/total\_words)))  
  
  
  
  
 }  
}

# Output

the probablity estimate if word 'the' is 0.0329619284

the probablity estimate if word 'become' is 0.0001769570

the probablity estimate if word 'brave' is 0.0001929323

the probablity estimate if word 'treason' is 0.0001130559

the probablity of appearing 'court' given 'the' has appeared is 0.003989114

the probablity of appearing 'word' given 'his' has appeared is 0.0027565083

the probablity of appearing 'qualities' given 'rare' has appeared is 0.017857144

the probablity of appearing 'men' given 'young' has appeared is 0.021531101

the probablity of appearing 'have' given 'sent' has appeared is 0.0

the probablity of appearing 'will' , 'look' and 'upon' together is 1.5848241089349301E-18

the probablity of appearing 'i' , 'am', 'no' and 'baby' together is 2.3958831432254025E-26

the probablity of appearing 'wherefore' , 'art', 'thou' and 'Romeo together is 4.329073442154424E-28