COE 312

Data Structures

Fall 2016

*W. FAWAZ*

Project I

# I. Objective

In this project, you are tasked with developing a software tool that serves as a **phishing scanner**. Phishing is a form of identity theft in which, in an email, a sender posing as a trustworthy source attempts to acquire private information, such as your user names, passwords, credit-card numbers, and social security number. Phishing emails claiming to be from popular banks, credit-card companies, auction sites, social networks, and online payment services may look quite legitimate. These fraudulent messages often provide links to fake websites where you are asked to enter sensitive information. Your phishing scanner tool will should be able to scan a file of text to determine **whether or not the file contains a fraudulent message**.

The development of the phishing scanner can be broken down into the following tasks, which can be gradually incorporated into your tool:

1. A list of the most commonly used words in phishing messages must be put together first with a point value (score) being assigned to each word indicating its likeliness to appear in a phishing message. Such a list is already provided for you and can be obtained from: <http://www.wissamfawaz.com/phishing.xml>. As part of this first task, you need to retrieve all of the words contained in this online xml file along with their associated scores and then print to the screen the words having the highest, lowest, and average score values. If you do not find words with a score value matching the average score value, then select the words whose score value matches the integer portion of the average score.
2. The second task consists of processing four input files that are made available to you at:

<http://www.wissamfawaz.com/in1.txt>, <http://www.wissamfawaz.com/in2.txt>, <http://www.wissamfawaz.com/in3.txt>, and <http://www.wissamfawaz.com/in4.txt>, respectively. In particular, your application should scan through each file individually in order to derive an overall score value per file as follows. For each encountered word within the text file, add its score value to the point total of the entire message. If the word is not among the ones you obtained in the first task, then use a score value of 0 for the word. Once a total point value is computed for each file, print the computed values to the screen.

1. Based on the overall score value calculated in the second task, your application should now classify each one of the individual files into one of the four categories found at: <http://www.wissamfawaz.com/categories.json>. This json file provides the classification guidelines to be followed when identifying the class of each of the four text files from the second task.

More specifically, if the score value of the file is found to be less than the threshold value pertaining to the lowest category level (i.e., “not likely” category), then the file is assumed to belong to that category. If, on the other hand, the score value is found to be between the threshold of the lowest category level and the next category level up, then the file is assumed to belong to the next category level, and so on and so forth. Once the categories of the individual files have been correctly identified, make sure to print them out to the screen.

Once you completed the three tasks delineated above, submit your solution through the “online services” website.

**Solution:**

package com.wissamfawaz.phishingscanner;

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.URL;

import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import org.json.JSONArray;

import org.json.JSONObject;

import org.json.JSONTokener;

import org.w3c.dom.Document;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

public class PhishingScanner {

 public static void main(String[] args) throws Exception {

URL phishingURL = new URL("http://www.wissamfawaz.com/phishing.xml");

 // Task 1:

 ArrayList<String> words = new ArrayList<>();

 ArrayList<Integer> scores = new ArrayList<>();

 DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

 DocumentBuilder db = dbf.newDocumentBuilder();

 Document document = db.parse(phishingURL.openStream());

 NodeList valuesList = document.getElementsByTagName("value");

 NodeList scoresList = document.getElementsByTagName("score");

 Node valueNode, scoreNode;

 for(int i=0; i<valuesList.getLength(); i++) {

 valueNode = valuesList.item(i);

 words.add(valueNode.getTextContent());

 }

 int min=Integer.MAX\_VALUE, max = Integer.MIN\_VALUE;

 double average = 0;

 for(int i=0; i<scoresList.getLength(); i++) {

 scoreNode = scoresList.item(i);

 int score = Integer.parseInt(scoreNode.getTextContent());

 scores.add(score);

 if(score < min)

 min = score;

 if(score > max)

 max = score;

 average += score;

 }

 average /= scores.size();

 //System.out.println("Words: " + words);

 //System.out.println("Scores: " + scores);

 if(!scores.contains(average)) {

 average = (int) average;

 }

 //System.out.println("Min: " + min);

 //System.out.println("Max: " + max);

 //System.out.println("Average: " + average);

 ArrayList<String> wordsWithMinScore = new ArrayList<>();

 ArrayList<String> wordsWithMaxScore = new ArrayList<>();

 ArrayList<String> wordsWithAvgScore = new ArrayList<>();

 for(int i=0; i<scores.size(); i++) {

 int score = scores.get(i);

 if(score == min)

 wordsWithMinScore.add(words.get(i));

 if(score == max)

 wordsWithMaxScore.add(words.get(i));

 if(score == average)

 wordsWithAvgScore.add(words.get(i));

 }

 System.out.println("Words with min score: " + wordsWithMinScore);

 System.out.println("Words with max score: " + wordsWithMaxScore);

 System.out.println("Words with avg score: " + wordsWithAvgScore);

 // Task 2:

 URL in1URL = new URL("http://www.wissamfawaz.com/in1.txt");

 URL in2URL = new URL("http://www.wissamfawaz.com/in2.txt");

 URL in3URL = new URL("http://www.wissamfawaz.com/in3.txt");

 URL in4URL = new URL("http://www.wissamfawaz.com/in4.txt");

 InputStream in1AsIS = in1URL.openStream();

 InputStream in2AsIS = in2URL.openStream();

 InputStream in3AsIS = in3URL.openStream();

 InputStream in4AsIS = in4URL.openStream();

BufferedReader br1 = new BufferedReader(new InputStreamReader(in1AsIS));

BufferedReader br2 = new BufferedReader(new InputStreamReader(in2AsIS));

BufferedReader br3 = new BufferedReader(new InputStreamReader(in3AsIS));

BufferedReader br4 = new BufferedReader(new InputStreamReader(in4AsIS));

 String line;

 String[] wordsInLine;

 int scoreIn1 = 0, scoreIn2 = 0, scoreIn3 = 0, scoreIn4 = 0;

 while((line = br1.readLine()) != null) {

 wordsInLine = line.split(" ");

 for(String wordInLine : wordsInLine) {

 if(words.contains(wordInLine))

 scoreIn1 += scores.get(words.indexOf(wordInLine));

 }

 }

 System.out.println("Score for in1.txt: " + scoreIn1);

 while((line = br2.readLine()) != null) {

 wordsInLine = line.split(" ");

 for(String wordInLine : wordsInLine) {

 if(words.contains(wordInLine))

 scoreIn2 += scores.get(words.indexOf(wordInLine));

 }

 }

 System.out.println("Score for in2.txt: " + scoreIn2);

 while((line = br3.readLine()) != null) {

 wordsInLine = line.split(" ");

 for(String wordInLine : wordsInLine) {

 if(words.contains(wordInLine))

 scoreIn3 += scores.get(words.indexOf(wordInLine));

 }

 }

 System.out.println("Score for in3.txt: " + scoreIn3);

 while((line = br4.readLine()) != null) {

 wordsInLine = line.split(" ");

 for(String wordInLine : wordsInLine) {

 if(words.contains(wordInLine))

 scoreIn4 += scores.get(words.indexOf(wordInLine));

 }

 }

 System.out.println("Score for in4.txt: " + scoreIn4);

 // Task3:

URL categoriesURL = new URL("http://www.wissamfawaz.com/categories.json");

int notlikely=0, somewhatlikely=0, moderatelylikely=0, highlylikely=0;

JSONTokener tokener = new JSONTokener(categoriesURL.openStream());

 JSONArray categoriesJSONArray = new JSONArray(tokener);

 JSONObject categoryJSONObject;

 String category;

 int threshold;

 for(int i=0; i<categoriesJSONArray.length(); i++) {

 categoryJSONObject = categoriesJSONArray.getJSONObject(i);

 category = categoryJSONObject.getString("category");

 threshold = categoryJSONObject.getInt("threshold");

 if(category.equals("not likely"))

 notlikely = threshold;

 else if(category.equals("somewhat likely"))

 somewhatlikely = threshold;

 else if(category.equals("moderately likely"))

 moderatelylikely = threshold;

 else if(category.equals("highly likely"))

 highlylikely = threshold;

 }

 /\*System.out.println("Not likely threshold: " + notlikely);

 System.out.println("Somewhat likely threshold: " + somewhatlikely);

 System.out.println("Moderately likely threshold: " + moderatelylikely);

 System.out.println("Highly likely threshold: " + highlylikely);\*/

 if(scoreIn1 <= notlikely) {

System.out.println("in1.txt is not likely to be a fraudulent message");

 } else if(scoreIn1>notlikely && scoreIn1 <=somewhatlikely) {

System.out.println("in1.txt is somewhat likely to be a fraudulent message");

 } else if(scoreIn1>somewhatlikely && scoreIn1 <=moderatelylikely) {

System.out.println("in1.txt is moderately likely to be a fraudulent message");

 } else if(scoreIn1>moderatelylikely && scoreIn1 <=highlylikely)

System.out.println("in1.txt is highly likely to be a fraudulent message");

 if(scoreIn2 <= notlikely) {

 System.out.println("in2.txt is not likely to be a fraudulent message");

 } else if(scoreIn2>notlikely && scoreIn2 <=somewhatlikely) {

 System.out.println("in2.txt is somewhat likely to be a fraudulent message");

 } else if(scoreIn2>somewhatlikely && scoreIn2 <=moderatelylikely) {

 System.out.println("in2.txt is moderately likely to be a fraudulent message");

 } else if(scoreIn2>moderatelylikely && scoreIn2 <=highlylikely)

 System.out.println("in2.txt is highly likely to be a fraudulent message");

 if(scoreIn3 <= notlikely) {

 System.out.println("in3.txt is not likely to be a fraudulent message");

 } else if(scoreIn3>notlikely && scoreIn3 <=somewhatlikely) {

 System.out.println("in3.txt is somewhat likely to be a fraudulent message");

 } else if(scoreIn3>somewhatlikely && scoreIn3 <=moderatelylikely) {

 System.out.println("in3.txt is moderately likely to be a fraudulent message");

 } else if(scoreIn3>moderatelylikely && scoreIn3 <=highlylikely)

 System.out.println("in3.txt is highly likely to be a fraudulent message");

 if(scoreIn4 <= notlikely) {

 System.out.println("in4.txt is not likely to be a fraudulent message");

 } else if(scoreIn4>notlikely && scoreIn4 <=somewhatlikely) {

 System.out.println("in4.txt is somewhat likely to be a fraudulent message");

 } else if(scoreIn4>somewhatlikely && scoreIn4 <=moderatelylikely) {

 System.out.println("in4.txt is moderately likely to be a fraudulent message");

 } else if(scoreIn4>moderatelylikely && scoreIn4 <=highlylikely)

 System.out.println("in1.txt is highly likely to be a fraudulent message");

 }

}