CSI4107, Winter 2018

Assignment 2
Due March 31, 22:00

Sentiment Analysis in Twitter Messages [100 points]

Note: **You will work in groups of two students.**

In this assignment, you will classify tweeter messages as expressing a positive opinion, a negative opinion, or no opinion (neutral or objective).

Read more about the task at:
http://www.cs.york.ac.uk/semeval-2013/task2/
We will focus on Task B: Message Polarity Classification: Given a message, classify whether the message is of positive, negative, neutral sentiment, or objective – no opinion (four classes). For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen, or neutral is the two opinions are approximately equal.

The data consists in approximately 8000 tweets and it is available here.

The format of the data is, for each line:

```
<SID><tab><UID><tab><TOPIC><tab><positive|negative|neutral|objective><tab><TWITTER_MESSAGE>

Example of one line:
100032373000896513     15486118       lady gaga      "positive"
        Wow!! Lady Gaga is actually at the Britney Spears Femme Fatale
Concert tonight!!! She still listens to her music!!!! WOW!!!
```

You will use Machine Learning (ML) algorithms from a tool named Weka. First you will need to install Weka. It is written in Java. See more details and documentation about Weka. It can be used through its graphical user interface (or directly from Java programs through its API).

You need to write a program that extracts features from the tweets and save them in an .arff file. After that, you can open the arff file in Weka' GUI and run any machine learning algorithms that are appropriate for your task.

An example of possible format for an .arff file is the following:

```
@RELATION example_rel
@ATTRIBUTE a1  STRING
@ATTRIBUTE a2  {Y,N}
@ATTRIBUTE a3  NUMERIC
@ATTRIBUTE a4  NUMERIC
@ATTRIBUTE class       {C1, C2, C3}
@DATA
```

Str1,Y,1.4,0.2,C1
Str2,N,1.4,0.2,C2
Str3,Y,1.3,0.2,C1
Str1,N,1.5,0.2,C1
Str4,Y,1.4,0.2,C3
….

You will use an evaluation technique called 10-fold cross validation, available in Weka. This means that the data is split into 10 parts (9 for training and one for test). The classifier is trained on 9 training parts, using the provided class labels to learn associations between the data and the classes. Then the classifier is applied on the remaining test part in order to predict new labels. The existing labels in the test data are ignored during the prediction, but they are used at the end in order to compute the accuracy of the classification, by comparing the predicted labels with the expected labels. This procedure is repeated 10 times over all possible splits of the data in training /test parts. Then the reported results are the average over the 10 runs.

Try at least three classifiers from Weka. The main ones to try are SVM (SMO in Weka) because it tends to get the best results, Naive Bayes because it works well with texts, and Decision Trees (J48 in Weka) because you can see the tree that is learnt.

Perform the following experiments:

1. [30 marks] Train a classifier using the bag-of-words (BOW) representation. This means to use words for the messages as features in the arff file. You can eliminate stop words, rare words, punctuation, etc in order to reduce the dimension of the vector space. (Here is the input file already in arff format, for a quick run in Weka, but the words are not extracted. You can use the StringToWord attribute filter in Weka to extract them, but it might not tokenize the way you want. You can change the regular expression in the tokenizer, or better build the arrf file with your own program).

2. [30 marks] Add more features and train more classifiers, in order to try to improve the classification results. For example using the emoticons from the texts as features as should help. Using punctuation marks such as !. !!, !!!, ??, ???, and others elongations could help. Other features can be the number of positive words in the messages, and the number of negative words in the message (you can use lists of positive and negative words in order to count these kinds of words). Try at least two of these resources. (you can use separate features for number of positive/negative words from messages that are found in each resource individually).

[20 marks] Write a report in a file Report (.pdf, .doc, or .txt)
Explain what you did for step 1, and what extra features you computed in step 2. Report the accuracy of the classification on the test set for all the experiments that you ran, for the three classifiers (SVM, NB, DT), the confusion matrices, as well as the Precision, Recall, and F-measure for each of the four classes, as calculated by Weka. Discuss what classifier and what features led to your best results.

[20 marks] Resulst.txt
Submit the predictions of your best classifier in a file named Results.txt, as calculated by Weka (select the option Output predictions in order to get predictions for each Twitter message). The format should be the one produced by Weka. You can copy and paste Weka's results in the Results.txt file (using CTR+C, CTRL+V).

Resources: Any resources you want to use. Include in the Report file explanations on how you used them.
Here are resources that include lists of positive and negative words: General Inquirer, LIWC, List of Adjectives with semantic orientation (from Maite Taboada), Polarity lexicon (from Theresa Wilson), SentiWordNet, list from Kim and Hovy (automatically produced but contains a lot more words than the manually produced lists), etc.

Submission instructions:
- Submit your report and your best results for each message in a file Results.txt:

In the report include:
* the names and student numbers of the students in the group, and specify how the tasks were divided,
* explain what you did for the steps 1 and 2, what ML algorithms you tried and what data representations (features) you used
* discuss what classification method and feature representation led to the best results
* a detailed note about the functionality of your programs that extract features
* complete instructions on how to run them

- Submit your assignment as a zip file, including programs, Report file, and the Result.txt file through the Blackboard Learn. Only one partner in a team needs to submit.

Have fun!!!