CSI5386, Winter 2015

Assignment 2

Due Friday March 27, 22:00

**Emotion Analysis in Blogs [100 points]**

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

In this assignment, you will classify sentences by the emotional state expressed by the writer of blogs. The classes will be: happiness (hp), sadness (sd), fear (fr), surprise (sp), disgust (dg), anger (ag), and neutral (ne). Given a sentence, you need to classify it into one of these 7 classes.

For sentences conveying more than one emotion, whichever is the stronger emotion should be chosen.

The data consists of over 2000 annotated sentences and it is available [here](http://www.site.uottawa.ca/~diana/csi5386/Saimadata.arff). For more details about the dataset see this [paper](http://www-scf.usc.edu/~saman/pubs/2008-ijcnlp-final.pdf).

Each line contains one sentence (surrounded by apostrophes) and the class assigned by the two annotators, after the comma. For example:

' This was the best summer I have ever experienced ',hp

You will use Machine Learning (ML) algorithms from a tool named [Weka](http://www.cs.waikato.ac.nz/ml/weka/downloading.html). First you will need to install Weka. It is written in Java. See more details and [documentation](http://www.cs.waikato.ac.nz/ml/weka/documentation.html) about Weka. It can be used through its graphical user interface (or directly from Java programs through its API). Feel free to use other M: tool instead of Weka.

You can write a program that extracts features from the sentences and save them in the .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 or another tool. 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 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. In fact the file is already in .arff format, with a simple header that you should modify as needed. You can use the Weka filter StringToVector for a first quick experiment.

**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 emotion words in the sentences (you can use lists of emotion words from different resources). Try at least one such resource, for example [WordNet Affect](http://wndomains.fbk.eu/wnaffect.html) (here are the word lists for [download](http://www.site.uottawa.ca/~diana/csi5386/WordNetAffectEmotionLists.zip)). If you use more resources, you can use separate features for number of emotion words for each class that are found in each resource individually. Lists of positive and negative words might also help.

**[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 class, 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 blog sentence. The format should be the one produced byWeka. You can copy and paste Weka’s results in the Results.txt file (using CTR+C, CTRL+V).

**Submission instructions**:

   - Submit your report and your best results for each sentence 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 or by email. Only one partner in a team needs to submit.

**Have fun!!!**