COM6115 Text Processing

Assessment: Sentiment Analysis

Quick Summary

To better understand the strengths and limitations of Bayesian text classification, in this assignment, you are going to investigate Sentiment Analysis using two sentiment datasets provided. You will also be provided a python script that implements Naive Bayes. You will need to write a report (**no more than 800 words**) to describe your result and findings.

Note: This assessment accounts for 30% of your total mark of the course. Your report may be submitted for plagiarism check (e.g., Turnitin). For any clarifications on this assessment, please contact **Dr Chenghua Lin** (c.lin@sheffield.ac.uk).

Assessment Tasks

STEP 1: Download the data from Blackboard. This contains the following:

1. A dataset with snippets of movie review from the Rotten Tomatoes website (in a format where one text file each containing positive and negative reviews).
    1. rt-polarity.pos
    2. rt-polarity.neg
2. A smaller dataset with snippets of reviews of Nokia phones (again 2 files)
    1. nokia-pos.txt
    2. nokia-neg.txt
3. A sentiment dictionary of positive and negative sentiment words:
    1. negative-words.txt contains 4783 negative-sentiment words
    2. positive-words.txt contains 2006 positive-sentiment words
4. The python script with my implementation of Naive Bayes, a knowledge-based classifier using the sentiment dictionary, as well as some helper functions:
    1. Sentiment.py (you will need to run Python3)

STEP 2: Familiarise:

1. The code splits the Rotten Tomatoes Data into a training and test set in readFiles(), then builds the p(word|sentiment) model on the training data in trainBayes(), and finally applies Naive Bayes to the test data in testBayes().
2. Write a function which will print out Accuracy, Precision, Recall and F-measure for the test data. **[10 pt]**
3. Run the code and report the classification results. **[5 pt]**

STEP 3: Run Naive Bayes on other data:

1. In the python script, towards the end of the file (lines 272 and 274), uncomment out the other two calls to testBayes(). These run NaiveBayes on the training data and on Nokia product reviews
2. What do you observe? Why are the results so different? **[10 pt]**

STEP 4: What is the model being learnt?

1. Which are the most useful words for predicting sentiment? The code you have downloaded contains another function mostUseful() that prints the most useful words for deciding sentiment. **[5 pt]**
2. Uncomment the call to mostUseful(pWordPos, pWordNeg, pWord, 50) at the bottom of the program, and run the code again. This prints the words with the highest predictive value. Are the words selected by the model good sentiment terms? How many are in the sentiment dictionary? **[5 pt]**

STEP 5: How does a rule-based system compare?

1. Add some code for the function testDictionary() which will print out Accuracy, Precision, Recall and F-measure for the test data. **[5 pt]**
   Uncomment out the three lines towards the end of the program that call the function testDictionary() and run the program again. All this code does is add up the number of negative and positive words present in a review and predict the larger class.
2. How does the dictionary-based approach compare to Naive Bayes on the two domains? What conclusions do you draw about statistical and rule-based approaches from your observations? **[5 pt]**
3. Write a new function to improve the rule-based system, e.g., to take into account negation. Run the program again and analyse the results. [**20 pt**]

STEP 6: Error Analysis.

1. Comment out all but one of the testBayes/testDictionary calls.
2. At the top of the program, set PRINT_ERRORS=1
3. Run the program again, and it will print out the mistakes made. List the mistakes in the report **[5 pt]**
4. Please explain why the model is making mistakes (e.g., analyse the errors and report any patterns or generalisations). **[15 pt]**

Marking Criteria

Submit a report to describe your result and findings by following the tasks detailed in the in the 6 steps above.

1. Quality of the report, including structure and clarity. No more than 800 words **[15 pt]**
2. Step 2 **[15 pt]**
3. Step 3 **[10 pt]**
4. Step 4 **[10 pt]**
5. Step 5 **[30 pt]**
6. Step 6 **[20 pt]**

Submission Guideline

You should submit **a PDF version of your report along with your code** via Blackboard **by Morning 23:59am on Friday 18th December 2020**. The name of the PDF file should have the form “COM6115_Assessment-SA_< your Surname>_<your first name>_<Your Student ID>”. For instance, “COM6115_Assessment-SA_Smith_John_XXXXX.pdf”, where

XXXXX is your student ID.