CSCE-689, Programming Assignment #4 Sentiment Lexicon Induction
Due: Sunday, November 6, 2016 by 11:00pm

Your goal for this homework is to induce a sentiment phrasal lexicon and use it to perform Sentiment Analysis, following the approach as described in the paper:

Turney, Peter D. 2002. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews." Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).

with the change as specified below:

IR search: instead of using an external search engine to collect hit counts, you should write your own search code to search in the training set, including implementing the 'NEAR' operator.

In addition, POS tags are needed to generate sentiment prhases. to generate POS tags, please use the following online downloadable software:

`http://www.umiacs.umd.edu/~hal/TagChunk/`

This project is a closely related to PA #2, we will use the same data set and the same 10-fold cross-validation setting. But instead of developing statistical classifiers, we will develop a close-to-deterministic approach but exploit the idea of building a fancy phrasal sentiment lexicon and using statistical measures.

10-fold cross-validation: For each fold, you generate sentiment phrases from the test set and derive their semantic orientation values using the training set. Your final sentiment analysis accuracy is the average of the 10 runs.

It's your choice to reuse part of the python starter code as provided for PA #2.

This is a mini real NLP project involving a few basic processing steps:

Key Steps:

(1 Points!) run the specified POS tagger through the dataset. Please show the commands you used to run the tagger in your report.

(2 Point!) design regular expressions to generate sentiment phrases as specified in the paper. Please show all your regular expressions and examples of sentiment phrases in your report.

(3 Points!) write code to conduct search including implementing the "NEAR" operator. Please paste the relevant part of code in your report.

(3 Points!) calcualte the semantic orientation for each sentiment phrase. Please paste the relevant part of code in your report.

(1 Points!) calculate the polarity score for each test review. Please paste the relevant part of code in your report.

OUTPUT FORMATTING

The results will look something like the figure in the next page. Even if you program using a language different from Python and do not use the starter code, your program output should follow the same structure.

GRADING CRITERIA

Your program will be graded all based on cross validation results on the provided training set, and mainly based on the correctness of your code (please refer back to the first section for how we will weight each task). We understand that you may produce slightly different results if you have implemented some details in a different way. Therefore, please describe enough details in your written report. In addition, we will test your code on a separate test set to make sure it runs properly. For instance, if we found any experimental result has been hard coded, you won't get any credit for that part.

ELECTRONIC SUBMISSION INSTRUCTIONS (a.k.a. "What to turn in and how to tun in")

You need to submit 2 things:

1. The source code files for your program. Be sure to include all files that we will need to compile and run your program!
2. A report file that includes the following information:
   - how to compile and run your code
   - results and analysis
   - any known bugs, problems, or limitations of your program

   REMINDER: your program **must** compile and run. We will not grade programs that cannot be run.

How to turn in:

1. please include everything in a single package, have it compressed and emailed to our grader Guanlun at "guanlun@tamu.edu".
2. please wrtie the subject line as: "PA #4 Sentiment Lexicon Induction for CSCE689 - UIN - LastName", please replace the placeholders "UIN" and "LastName" with your own UIN and LastName.

```
[INFO] Fold 0 Accuracy: 0.500000

[INFO] Fold 1 Accuracy: 0.500000

[INFO] Fold 2 Accuracy: 0.500000

[INFO] Fold 3 Accuracy: 0.500000

[INFO] Fold 4 Accuracy: 0.500000

[INFO] Fold 5 Accuracy: 0.500000

[INFO] Fold 6 Accuracy: 0.500000

[INFO] Fold 7 Accuracy: 0.500000

[INFO] Fold 8 Accuracy: 0.500000

[INFO] Fold 9 Accuracy: 0.500000

[INFO] Accuracy: 0.500000
```

Figure 1: Sample Output for Sentiment Analyzer