NLP Programming Assignment #4 – Sentiment Lexical Induction

Submitted by – Vaibhav Rawat

UIN - 626008171

**Steps to compile and run the code with Python 2.7:**

1. Open the terminal
2. Go to the folder NLPAssign4
3. Enter the following commands:

python SentAnalyzer.py ./processed\_data

# for testing with other test set

python SentAnalyzer.py ./processed\_data ./test\_data

**Implementation Details:**

1. I used following Regular Expressions to extract two-word phrases:

" JJ\d\* NN[S]?\d\* "

"RB[S]?[R]?\d\* JJ\d\* (?![NN][S]?)"

"JJ\d\* JJ\d\* (?![NN][S]?)"

"NN[S]?\d\* JJ\d\* (?![NN][S]?)"

"RB[R]?[S]?\d\* VB[D]?[N]?[G]?\d\* "

Positions for all words are recorded.

Examples of sentiment phrases generated are:

bad films

good intentions

emotional closeness

good chance

slow parts

juvenile comedy

cheesy gore

special effects

awesome spectacle

beautiful cast

fast-paced action

pretty much

well sorry

brutally phony

fully aware

quite astounding

evil dead

automatically start

far surpassing

heavily brought

viciously decapitated

romantic comedy

flawed comedy

pesky ex-boyfriend

humorous cameo

fresh sort

romantic comedy

stubborn father

right time

usual flicks

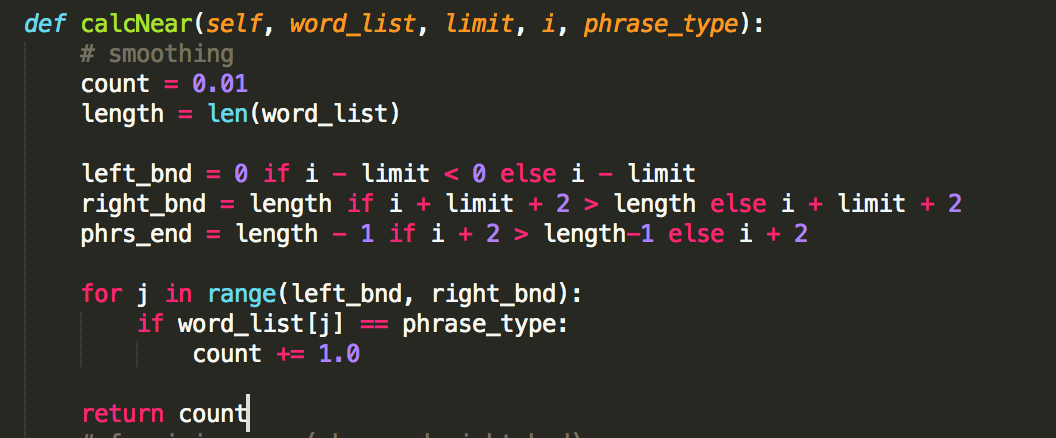
beautiful actress

straightforward guy

romantic comedy

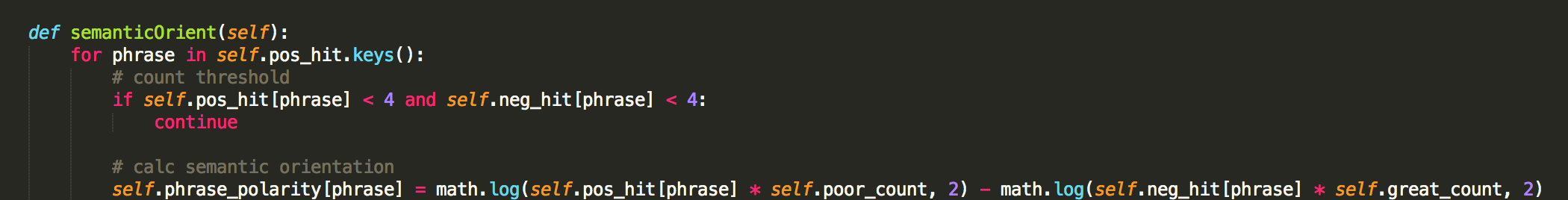
good laughs

2. For each phrase using a window of 10 (or 16) nearby words that come before and after the phrase, I calculated the nearness (hit count) of the phrase to great and poor words. Limit specifies the window length, phrase type is (great/poor) and i the index of phrase. Also to handle zero cases, I initialized count with 0.01.



4. Sematic Orientation for each phrase

Using formula from the paper based on PPI, I calculated semantic Orientation for each phrase which is log of (nearness to great \* poor count) - log of (nearness to poor \* great count). Semantic orientation reflects whether the word is close to “great” keyword or to “poor” keyword.



4. Using semantic orientation of individual phrases, polarity for a review is calculated. If polarity is positive, review is classified as positive else negative.



**Results and Analysis:**

I tried using different lengths for Nearness

Results with Nearness 10

Vaibhavs-MacBook-Pro:pa4-sentimentLexiconInduction vrawat279$ python SentAnalyzer.py processed\_docs/

[INFO] Fold 0 Accuracy: 0.570000

[INFO] Fold 1 Accuracy: 0.555000

[INFO] Fold 2 Accuracy: 0.570000

[INFO] Fold 3 Accuracy: 0.565000

[INFO] Fold 4 Accuracy: 0.510000

[INFO] Fold 5 Accuracy: 0.570000

[INFO] Fold 6 Accuracy: 0.540000

[INFO] Fold 7 Accuracy: 0.500000

[INFO] Fold 8 Accuracy: 0.540000

[INFO] Fold 9 Accuracy: 0.555000

[INFO] Accuracy: 0.547500

Results with Nearness 16 –

Vaibhavs-MacBook-Pro:pa4-sentimentLexiconInduction vrawat279$ python SentAnalyzer.py processed\_docs/

[INFO] Fold 0 Accuracy: 0.595000

[INFO] Fold 1 Accuracy: 0.560000

[INFO] Fold 2 Accuracy: 0.535000

[INFO] Fold 3 Accuracy: 0.560000

[INFO] Fold 4 Accuracy: 0.520000

[INFO] Fold 5 Accuracy: 0.555000

[INFO] Fold 6 Accuracy: 0.545000

[INFO] Fold 7 Accuracy: 0.500000

[INFO] Fold 8 Accuracy: 0.540000

[INFO] Fold 9 Accuracy: 0.595000

[INFO] Accuracy: 0.550500

**Error and Bugs:**

There are no issues.