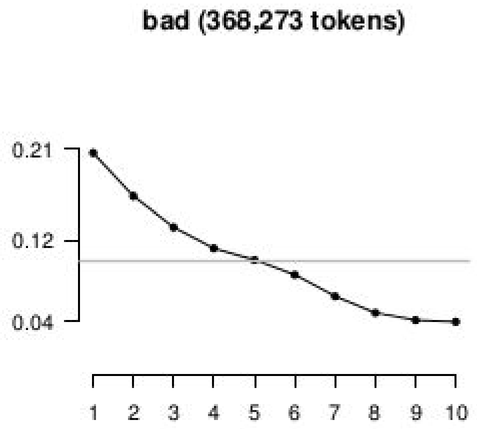
IMDB Movie Reviews

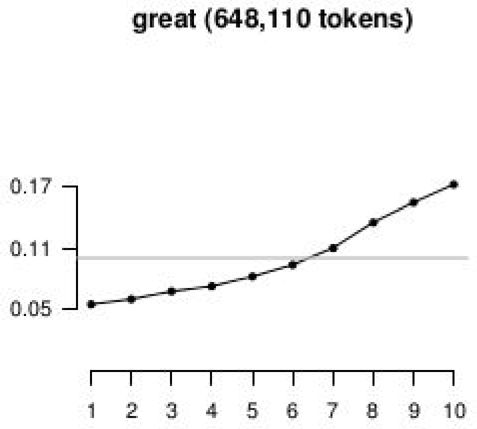
Jeremy Benson, Priyaranjan JC,
Jessica Jones, Xinyu Chen

Background

Score reviews based on words’ polarity as calculated by Christopher Potts (2011); a naive Bayes approach.

Probability review is in category given word:





Data Representation and Preprocessing

Data in vectorized format, where we consider only the numeric values; i.e., we don’t actually read the text.
For example: 7 0:10 1:6 2:4 3:4 4:4 5:1 6:1 8:3

Remove stop words based on open-source lists:

- We first used stopwords from nltk, which contains 127 stop words in English.
- We then tried a more complete list from http://www.ranks.nl/stopwords to remove 665 stop words. This list turns out to increase our accuracy.

MapReduce in Hadoop

Input to mappers includes the Bag of Words file (labeledBow.feat) and the dictionary file (imdbErNew.txt). Mapper calculates a score for each review, then outputs <docId, realScore, calculatedScore> as the <key, value> pair.

The reducers aggregate the results and update the confusion matrix.

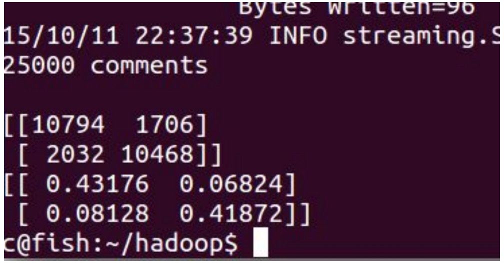
To specify the dictionary file, we use -cacheFile of the hadoop.streaming API. The Bag of Words and dictionary files are put into the dfs storage.

Pseudocode

```
MAPPER

for each review:

    for each (word:count) pair: multiply word count * word rating, add to review score

REDUCER

    for each review: compare review’s computed score to ground truth, update confusion matrix
```

Results

With stop words: 83.68% accuracy



```
                Bytes Written=96
15/10/11 22:37:39 INFO streaming.S
25000 comments

[[10794  1706]
 [ 2032 10468]]
[[ 0.43176  0.06824]
 [ 0.08128  0.41872]]
c@fish:~/hadoop$ █
```

Remove 665 stop words: 85.05%

By neutralizing unwanted weight for certain words: 86.72%

Additional Tests and Analyses

We collected most frequent words having higher weight from both True predictions and False predictions and neutralized the words which accounted for most of the false predictions.

By this we increased the accuracy by 1.5%.

Other Methods:

- Gave random weights for important words
- Gave weightage according to their false predictions

Thank you!!!