“Strong Words” for Sentiment Analysis on Yelp Reviews

Nils Jungenfelt

22 november 2015

Introduction

In this report I propose a new methodolgy to select “strong words” - words that have strong impact on sentiment in reviews. The results presented here are of interest to any business owner who wants to understand their customers, but also to a data scientist who’d like to improve their sentiment analysis methodology.

To verify the impact of those strong words, I have built a prediction model to predict the number of stars given by a reviewer based on the text in their review.

Methods and Data

This analysis is based on the review dataset from The Yelp Dataset Challenge, and is limited to the text field and the score (number of stars) given.

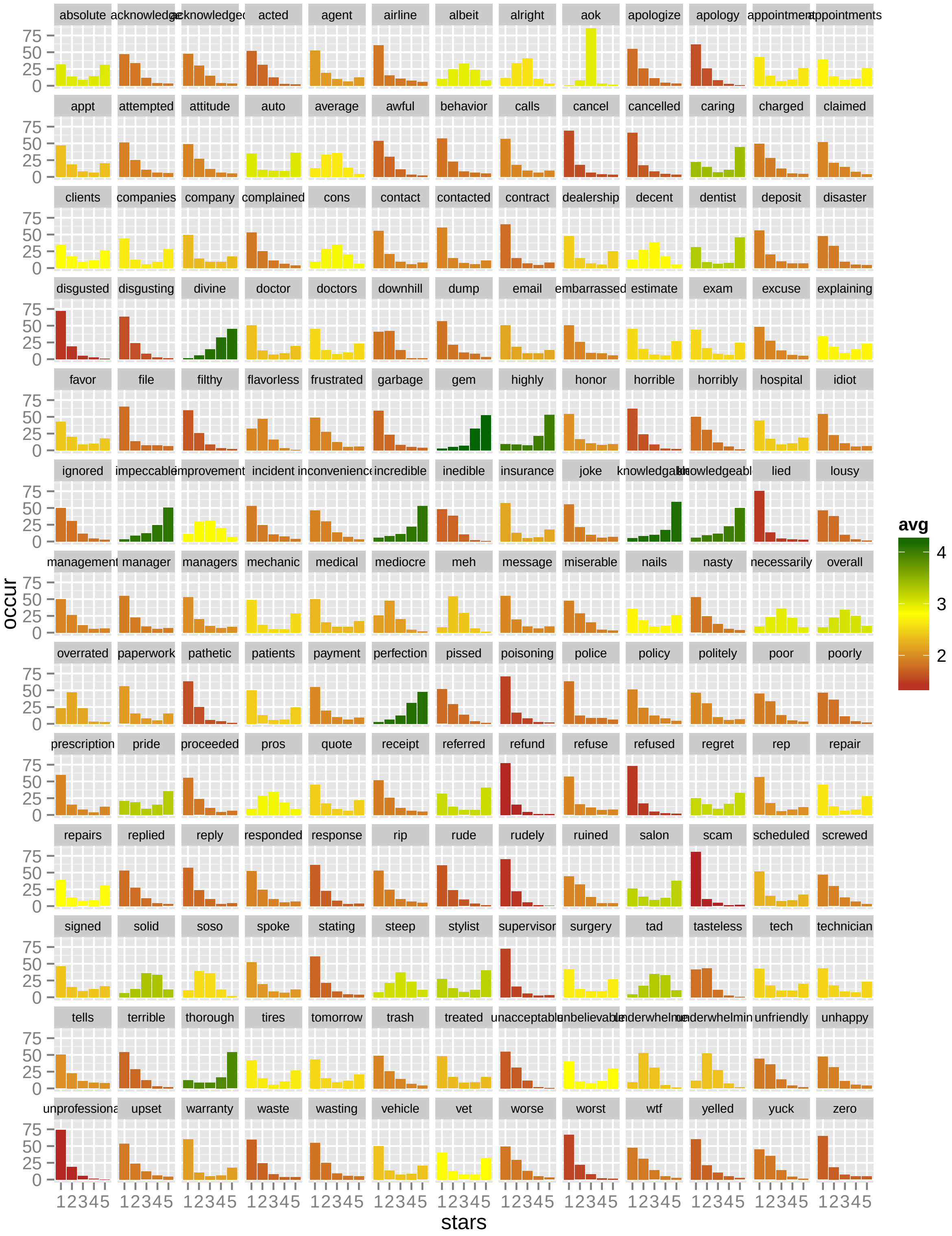
To begin with, 250,000 out of 1,569,264 reviews are randomly sampled, but with equally many (50,000) from every score. The reason for this semi-random sampling is to avoid over-sampling of postive words as mosts scores are fives. The review set is then structured into a matix using the ‘tm’ package in R. In addition to stopwords, words that appear less than $\sqrt{250,000} = 500$ times are removed.

For every word the distibution (as a percentage for each score) is calculated. For a “weak word”, this distribution would uniform, 20% in each score. In this report, a strong word is considered to have high standard deviation among the distribution bins.

In the particular example presented here, the 117 words with the highest variation among the distribution bins are selected. I also added 39 words with a ∪-shaped distribution (few 3-stars but an average around 3) and 13 words with a ∩-shaped distribution (few 1- or 5-stars and an average around 3). The reason for that is that such words may have high impact in the context of other (strong) words, allthough they are not considered in traditional sentiment analysis. In addition, predicting 3-stars is also important!

To be able to benchmark the methodolgy proposed, a prediction model using a neural network with one hidden layer is used. I used 20% of the reviews (that hade at least one stong word in them!) to train the model and 80% for verification.

Results



The words in the plot above are the ones I found using this method. They appear to exist in about 50% of all reviews.

The neural network prediction model perform well on reviews that have any of the stong words, the out of sample accuracy is almost 50% with a p-value below $2.2 \cdot 10^{-16}$. The full details on the statistics are shown below.

```
## Loading required package: lattice
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2     3     4     5
##          1 23916 10707  4235  2273  2249
##          2  4151  6881  3564  1273   748
##          3  1393  5814  9694  5409  1820
##          4    84   283   816  1099   637
##          5  1595  1387  1575  3723  8649
##
## Overall Statistics
##
##                Accuracy : 0.4832
##                  95% CI : (0.4801, 0.4862)
##     No Information Rate : 0.2995
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3256
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.7680  0.27445  0.48753  0.07977  0.61327
## Specificity            0.7328  0.87661  0.82833  0.97982  0.90787
## Pos Pred Value         0.5513  0.41409  0.40174  0.37650  0.51090
## Neg Pred Value         0.8808  0.79176  0.87238  0.87454  0.93734
## Prevalence             0.2995  0.24113  0.19124  0.13250  0.13564
## Detection Rate         0.2300  0.06618  0.09323  0.01057  0.08318
## Detection Prevalence   0.4172  0.15982  0.23208  0.02807  0.16282
## Balanced Accuracy      0.7504  0.57553  0.65793  0.52980  0.76057
```

Discussion

Some interesting stong words were found, and they seem to explain a lot of the sentiment in a review. However, the list of words should be made longer to cover a bigger percentage of reviews.