3.3 Other considerations

In the previous section I described in detail the process of words selection.

However, before beginning classification there is still a work to be done. Consider the following example: "My baby likes this toy", "My baby doesn't like this toy". Using our previous approach there is no difference between those two sentences in a classification matrix. Nevertheless, obviously there is a significant difference – polarity is exactly the opposite. One solution would be to include "not" in the dictionary and hope that classifier will give it significant negative wage. The issue with this approach though is that in more complex reviews, a negation can bear positive sense. Consider an example "I really like this cradle. It looks nice and my baby does not cry as often as before". We can clearly see that there is nothing negative in this review, yet this particular review can be classified as negative. Therefore, another solution should be found.

One way to deal with this issue is to use antonyms. However, a list of antonyms should be defined. Even though it is certainly possible to find antonyms dictionary for lots of languages, this solution lacks generalization. Some domain specific words are probably not defined in such dictionaries. Besides, some words have many antonyms and it is quite ambiguous to decide which selection is the best.

I am going to use the solution proposed in (TODO cite [2] Wake, Lisa.). Every word after negation will be prefixed with “NOT\_” prefix. Because the punctuation is eliminated before text analysis, this transformation will affect only the closest words near a negation. Usually it is exactly what is required. Of course number of words is increased that way. Nevertheless, as described before, because I build vector based on top $k$ used words, in the result there is no difference.

After words are selected there is natural choice between representing text as a list of word counts or as a Boolean vector. According to some studies in Sentiment Analysis the number of occurrences of the word in the text does not make much of a difference. Usually Binarized versions (occurrences clipped to 1) of the algorithms perform better than the ones that use multiple occurrences.

Another reason to tone down words that appear often in a text is that a word that appears regularly is more likely to have a neutral sense. This is particularly true of nouns. In one example from our corpus, the words death, turmoil, and war each appear twice. A single use of any of these words might indicate a comment (e.g., I was bored to death), but repeated use suggests a descriptive narrative.

TODO

Compare my results with:

Prepared lexicons (general lexicon) (Positive negative manual mapping)

List of significant words

First, I am going to compare different configurations of my algorithm on a subset of the data. There are different configurations. In particular:

USE\_STOP\_WORDS = True

USE\_EMOTICONS = False

USE\_NEGATION = True

# If set to false, number of occurances of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

I will fix NUMBER\_OF\_REVIEWS\_TO\_ANALYZE and NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE for training to complete in reasonable time. For every run the data is divided into training and test sets. I will show performance for both 60:40 and 80:20 ratio. I consider confusion matrix and classification report as a performance metric.

Classification report consists of precision, recall, f1-score for each class. Moreover, there are averages of this metrics, which I consider the most crucial part. I think of an algorithm A as a “better” algorithm than an algorithm B if its average f1-score is higher.

COMPARISON 1

NAÏVE BAYES 80:20

USE\_STOP\_WORDS = False

USE\_EMOTICONS = False

USE\_NEGATION = False

# If set to false, number of occurances of words is calculated

USE\_BOOLEAN\_REPRESENTATION = False

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

Confusion matrix:

[[123 23 30 19 47]

[ 43 25 24 14 36]

[ 26 21 34 41 49]

[ 26 27 46 82 164]

[ 52 28 34 100 886]]

Classification report:

precision recall f1-score support

1 0.46 0.51 0.48 242

2 0.20 0.18 0.19 142

3 0.20 0.20 0.20 171

4 0.32 0.24 0.27 345

5 0.75 0.81 0.78 1100

avg / total 0.55 0.57 0.56 2000

USE\_STOP\_WORDS = False

USE\_EMOTICONS = False

USE\_NEGATION = False

# If set to false, number of occurances of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

Confusion matrix:

[[132 19 21 18 52]

[ 44 24 20 17 37]

[ 28 9 38 32 64]

[ 26 20 35 83 181]

[ 43 17 24 82 934]]

Classification report:

precision recall f1-score support

1 0.48 0.55 0.51 242

2 0.27 0.17 0.21 142

3 0.28 0.22 0.25 171

4 0.36 0.24 0.29 345

5 0.74 0.85 0.79 1100

avg / total 0.57 0.61 0.58 2000

USE\_STOP\_WORDS = False

USE\_EMOTICONS = False

USE\_NEGATION = True

# If set to false, number of occurances of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

Confusion matrix:

[[134 23 24 17 44]

[ 50 18 30 15 29]

[ 24 15 42 33 57]

[ 20 20 35 92 178]

[ 36 15 32 90 927]]

Classification report:

precision recall f1-score support

1 0.51 0.55 0.53 242

2 0.20 0.13 0.15 142

3 0.26 0.25 0.25 171

4 0.37 0.27 0.31 345

5 0.75 0.84 0.79 1100

avg / total 0.57 0.61 0.59 2000

USE\_MY\_METHOD = True

USE\_STOP\_WORDS = False

USE\_EMOTICONS = True

USE\_NEGATION = True

# If set to false, number of occurances of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

Confusion matrix:

[[136 24 20 18 44]

[ 49 19 30 14 30]

[ 25 15 43 32 56]

[ 21 20 32 92 180]

[ 35 15 29 97 924]]

Classification report:

precision recall f1-score support

1 0.51 0.56 0.54 242

2 0.20 0.13 0.16 142

3 0.28 0.25 0.26 171

4 0.36 0.27 0.31 345

5 0.75 0.84 0.79 1100

avg / total 0.57 0.61 0.59 2000

USE\_STOP\_WORDS = True

USE\_EMOTICONS = True

USE\_NEGATION = True

# If set to false, number of occurances of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

Confusion matrix:

[[135 25 20 16 46]

[ 52 19 17 20 34]

[ 32 9 38 33 59]

[ 26 18 33 77 191]

[ 39 11 21 80 949]]

Classification report:

precision recall f1-score support

1 0.48 0.56 0.51 242

2 0.23 0.13 0.17 142

3 0.29 0.22 0.25 171

4 0.34 0.22 0.27 345

5 0.74 0.86 0.80 1100

avg / total 0.57 0.61 0.58 2000

As we can see using stop words in sentiment analysis not only does not improve, but even harm the performance. Use of emoticons does not decrease the performance but the gain, if it is present, is too insignificant.

This result is applicable only for the subset of the date with NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000. For bigger number of words results might be different. Yet using boolean representation and negation clearly improve performance.

Using this results I have decided to use the following configuration in the next comparison:

USE\_STOP\_WORDS = False

USE\_EMOTICONS = False

USE\_NEGATION = True

# If set to false, number of occurences of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 10000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

COMPARISON 2

Let’s compare performance of Naive Bayes, Logistic Regression and KNN on 10k dataset.

80:20

Naive Bayes

Confusion matrix:

[[134 23 24 17 44]

[ 50 18 30 15 29]

[ 24 15 42 33 57]

[ 20 20 35 92 178]

[ 36 15 32 90 927]]

Classification report:

precision recall f1-score support

1 0.51 0.55 0.53 242

2 0.20 0.13 0.15 142

3 0.26 0.25 0.25 171

4 0.37 0.27 0.31 345

5 0.75 0.84 0.79 1100

avg / total 0.57 0.61 0.59 2000

Logistic regression

Confusion matrix:

[[124 28 20 19 51]

[ 43 32 26 11 30]

[ 24 26 37 22 62]

[ 18 28 32 88 179]

[ 51 24 30 111 884]]

Classification report:

precision recall f1-score support

1 0.48 0.51 0.49 242

2 0.23 0.23 0.23 142

3 0.26 0.22 0.23 171

4 0.35 0.26 0.30 345

5 0.73 0.80 0.77 1100

avg / total 0.56 0.58 0.57 2000

KNN

Confusion matrix:

[[ 47 14 12 11 158]

[ 21 10 10 11 90]

[ 15 2 13 20 121]

[ 15 3 14 28 285]

[ 52 21 31 49 947]]

Classification report:

precision recall f1-score support

1 0.31 0.19 0.24 242

2 0.20 0.07 0.10 142

3 0.16 0.08 0.10 171

4 0.24 0.08 0.12 345

5 0.59 0.86 0.70 1100

avg / total 0.43 0.52 0.45 2000

As we can see KNN performs poorly and there is no reason to use it in the future.

60:40

Naive Bayes

Confusion matrix:

[[ 271 35 42 41 82]

[ 104 34 43 46 65]

[ 50 30 68 82 111]

[ 41 36 62 191 366]

[ 84 37 48 196 1835]]

Classification report:

precision recall f1-score support

1 0.49 0.58 0.53 471

2 0.20 0.12 0.15 292

3 0.26 0.20 0.23 341

4 0.34 0.27 0.31 696

5 0.75 0.83 0.79 2200

avg / total 0.56 0.60 0.58 4000

Logistic regression

Confusion matrix:

[[ 230 66 56 39 80]

[ 91 53 44 40 64]

[ 41 56 75 69 100]

[ 39 49 81 205 322]

[ 102 62 85 309 1642]]

Classification report:

precision recall f1-score support

1 0.46 0.49 0.47 471

2 0.19 0.18 0.18 292

3 0.22 0.22 0.22 341

4 0.31 0.29 0.30 696

5 0.74 0.75 0.75 2200

avg / total 0.55 0.55 0.55 4000

COMPARISON 3  
  
For NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 100000, ther was no difference whether use stop words and emoticons or not. I used:

USE\_STOP\_WORDS = True

USE\_EMOTICONS = False

USE\_NEGATION = True

# If set to false, number of occurrences of words is calculated

USE\_BOOLEAN\_REPRESENTATION = True

NUMBER\_OF\_REVIEWS\_TO\_ANALYZE = 100000

NUMBER\_OF\_POPULAR\_WORDS\_TO\_USE = 1000

Naive Bayes 80:20

Confusion matrix:

[[1085 177 155 67 331]

[ 382 165 245 142 335]

[ 243 137 449 401 596]

[ 176 64 318 1166 1888]

[ 325 87 168 927 9971]]

Classification report:

precision recall

1 0.49 0.60

2 0.26 0.13

3 0.34 0.25

4 0.43 0.32

5 0.76 0.87

avg / total 0.61 0.64

f1-score support

0.54 1815

0.17 1269

0.28 1826

0.37 3612

0.81 11478

0.62 20000

Logistic regression 80:20

Confusion matrix:

[[ 1067 157 117 42 432]

[ 394 169 200 96 410]

[ 214 166 405 310 731]

[ 88 79 274 922 2249]

[ 133 66 123 501 10655]]

Classification report:

precision recall f1-score support

1 0.56 0.59 0.58 1815

2 0.27 0.13 0.18 1269

3 0.36 0.22 0.28 1826

4 0.49 0.26 0.34 3612

5 0.74 0.93 0.82 11478

avg / total 0.61 0.66 0.62 20000

Now let us compare my solution with predictor based on prepared lexicon. In particular the lexicon of positive and negative words (TODO cite) is used. The formula of prediction is 1 + round(4 \* (positive count / (positive count + negative count)). Now the NUMBER\_OF\_REVIEWS\_TO\_ANALYZE is 100000. The results are bellow:

Confusion matrix:

[[ 1169 1818 3353 1683 897]

[ 413 807 2345 1762 997]

[ 381 707 2850 3015 2204]

[ 296 736 4224 7058 5977]

[ 707 1296 9185 18505 27615]]

Classification report:

precision recall f1-score support

1 0.39 0.13 0.20 8920

2 0.15 0.13 0.14 6324

3 0.13 0.31 0.18 9157

4 0.22 0.39 0.28 18291

5 0.73 0.48 0.58 57308

avg / total 0.52 0.39 0.43 100000

For this dataset there is a list of words suggested for binary sentiment analysis. I have tried using it for this purpose and it produced fairly good results. Now let us compare how it performs at more accurate classification task and compare its performance with the performance of my solution.

Naïve Bayes 80:20  
Confusion matrix:

[[ 511 9 10 0 1285]

[ 214 13 20 1 1021]

[ 148 8 20 4 1646]

[ 76 4 21 3 3508]

[ 153 7 25 7 11286]]

Classification report:

precision recall f1-score support

1 0.46 0.28 0.35 1815

2 0.32 0.01 0.02 1269

3 0.21 0.01 0.02 1826

4 0.20 0.00 0.00 3612

5 0.60 0.98 0.75 11478

avg / total 0.46 0.59 0.46 20000

Logistic regression:

Confusion matrix:

[[ 510 9 4 4 1288]

[ 219 11 9 13 1017]

[ 144 7 14 10 1651]

[ 63 8 10 22 3509]

[ 141 2 5 18 11312]]

Classification report:

precision recall f1-score support

1 0.47 0.28 0.35 1815

2 0.30 0.01 0.02 1269

3 0.33 0.01 0.01 1826

4 0.33 0.01 0.01 3612

5 0.60 0.99 0.75 11478

avg / total 0.50 0.59 0.47 20000

TO conclusions:  
Product reviews, etc are relatively easy