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.

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 round(5 \* (positive count / (positive count + negative count)). Now the NUMBER\_OF\_REVIEWS\_TO\_ANALYZE is 100000. The results are bellow:

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 on more accurate classification task and compare its performance with the performance of my solution.

TO conclusions:  
Product reviews, etc are relatively easy