Sentiment Analysis for Movie Reviews

Introduction:

Providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer.

In this project we aim to use Sentiment Analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie was, i.e. if they liked the movie or they hated it. We aim to utilize the relationships of the words in the review to predict the overall polarity of the review.

Dataset:

It contains 50,000 training examples collected from IMDb [1] where each review is labelled with the rating of the movie on scale of 1-10. As sentiments are usually bipolar like good/bad or happy/sad or like/dislike, we categorized these ratings as either 1 (like) or 0 (dislike) based on the ratings. If the rating was above 5, we deduced that the person liked the movie otherwise he did not.

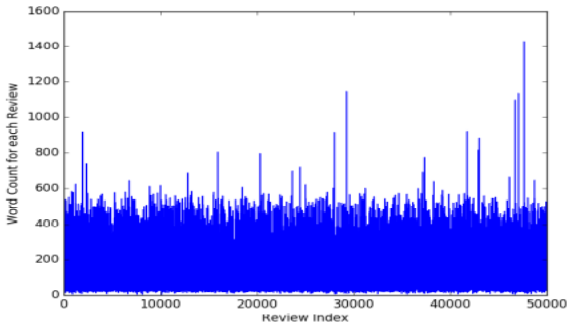
the dataset was divided into two subsets containing 25,000 examples each for training and testing, division to be sub-optimal as the number of training examples was very small and leading to under-fitting, then 40,000 for training and 10,000 for testing. Finally, we decided to use Cross Validation [2] in which the complete dataset is divided into multiple folds with different samples for training and validation each time and the final performance statistic of the classifier is averaged over all results.

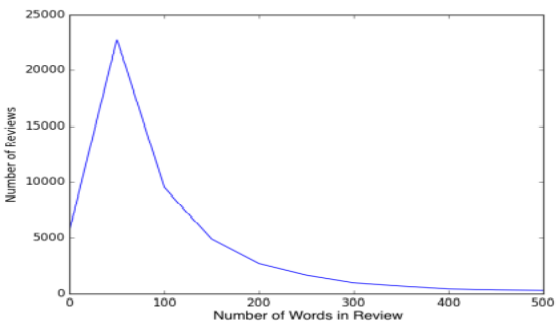
Predictive Task:

The main aim of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. In this project, we try to classify whether a person liked the movie or not based on the review they give for the movie. As a part of this project, we aim to study several feature extraction techniques used in text mining e.g. keyword spotting, lexical affinity and statistical methods, and understand their relevance to our problem, we also look into different classification techniques and explore how well they perform for different kinds of feature representations. We finally draw a conclusion regarding which combination of feature representations and classification techniques are most accurate for the current predictive task.

Exploratory Analysis:

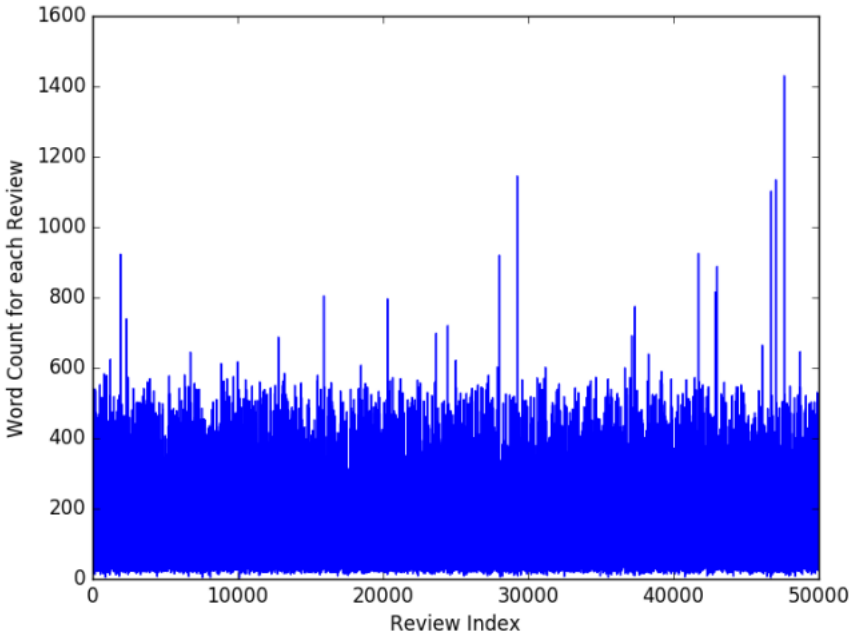
One of the starting points while working with review text is to calculate the average size of reviews to get some insight on quality of reviews. The average number of words per review is around 120. The graphs below clearly indicate the variation of the word count for each review. From this information we deduced that in general people tend to write pretty descriptive reviews for movies and as such this is a good topic for sentiment analysis. Also, people generally write reviews when they have strong opinions about a movie; they either loved it or hated it.





Apart from the word count per review another interesting metric was occurrence count of words across reviews. Some words have higher occurrence counts as compared to others depending on their relative importance. Below is the list of 20 most occurring words in negative and positive reviews along with a graph showing variability of word occurrences across all reviews

Negative Reviews		**Positive Reviews**	
Movie	Film	Film	Movie
Like	Even	Like	Good
Good	Bad	Great	Story
Would	Really	See	Time
Time	See	Well	Also
Don't	Get	Really	Would
Much	Story	Even	Much
People	Could	First	Films
Make	Made	Love	People
Movies	First	Best	Get



Feature Extraction:

We used 3 methods for extraction of meaningful features from the review text which could be used for training purposes. These features were then used for training several classifiers.

- Bag of Words.
- N-Gram Modelling.
- TF-IDF Modelling.

Models:

The overall task in this project is for classification of reviews as favorable or unfavorable. Therefore, for this classification task we explored multiple classification models on above feature representations. We used the models ranging from the simple Logistic Regression to the stateof-art SVM Classifier. We also used other classification models like SGD Classifier and Random Forest Classifier. Apart from these, we also trained the above feature representations on Naïve Bayes' Classifier as this is primarily used in case of text mining in combination with Bag of Words and N-Gram Modelling. We also trained a model based on k-Nearest Neighbors to match the similarity between the reviews and classify them accordingly.

For all of the above models, we used sklearn[3] modules by tuning their parameters and not changing their implementations and so we will not go into their theory in this report.

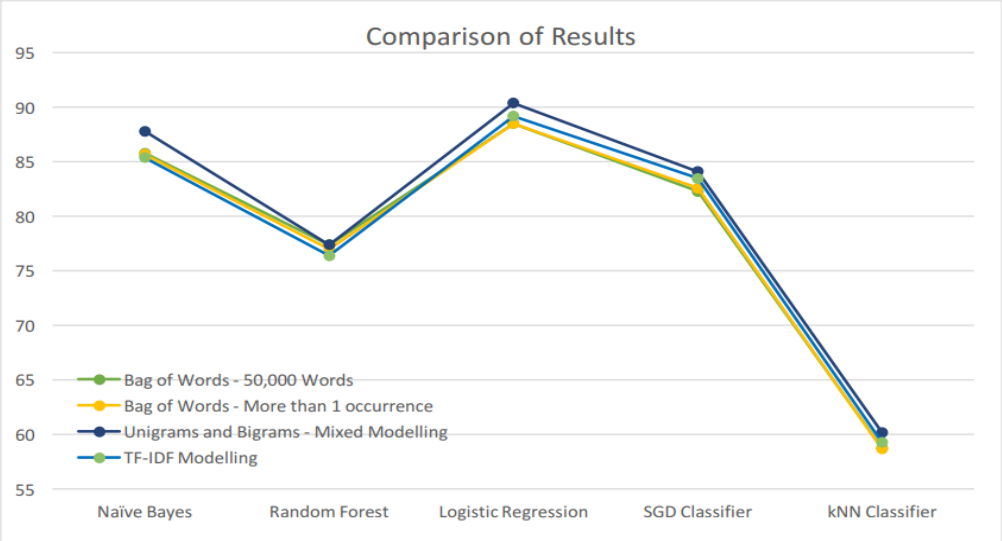
Before using the above feature representations for training classifiers, we tried reducing the size of representation set by using PCA on it. But it did not give us much improvement as the feature vector was reduced only by 15% and hence we did not incorporate those reductions. One important point to note is that for performance measure we are using Mean Absolute Error and not Mean Squared Error (MSE). This is because MAE will directly tell us the amount of misclassification we are doing for each model.

Also, as mentioned previously, we ran these training exercises to fit parameters on set selection using cross-validation techniques.

Results:

As discussed above, we tried multiple classification models on various feature representations of the textual information in the reviews. Out of these SVM Classifier failed to even converge for all of our feature sets and hence we could not get a satisfactory answer for it. Among the remaining models, Logistic Regression model seemed to have best performance across all feature representations with classification accuracy around 89%. Also, k-Nearest Neighbors classifier had the worst accuracy of around 60% across all feature representations. The general order of performance for the model was LogisticRegression > NaïveBayes > SGDClassifier > RandomForestClassifier > kNNClassifier. For a given classifier, the model that performed best used a feature set of a mixture of unigrams and bigram.

	Naïve Bayes	Random Forest	Logistic Regression	SGD Classifier	kNN Classifier
Bag of Words – 50,000 Words	85.8	77.4	88.5	82.3	58.8
Bag of Words – 1,00,000 Words	85.9	76.8	88.6	83.4	58.7
Bag of Words – More than 1 occurrence	85.7	77.0	88.5	82.6	58.7
Bag of Words – More than 5 occurrence	85.6	77.5	88.4	82.3	58.6
BiGram Modelling	86.5	77.1	88.7	83.2	58.6
Unigram and Bigram Mixed Modelling	87.8	77.4	90.4	84.1	60.2
Mixed Modelling – N = 5	86.8	77.2	89.1	83.6	59.2
TF-IDF Modelling	85.4	76.4	89.2	83.5	59.3



Conclusions:

From the results above, we can infer that for our problem statement, Logistic Regression Model with feature set using mixture of Unigrams and Bigrams is best. Apart from this, one can also use a Naïve Bayes' Classifier or a SGD classifier as they also provide good accuracy percentage. One peculiar thing to note is low accuracy with Random Forest classifier. This might be because of over-fitting of decision trees to the training data. Also, low accuracy of kNN Classifiers shows us that people have varied writing styles and kNN Models are not suited to data with high variance. One of the major improvements that can be incorporated as we move ahead in this project is to merge words with similar meanings before training the classifiers[3]. Another point of improvement can be to model this problem as a multi-class classification problem where we classify the sentiments of reviewer in more than binary fashion like "Happy", "Bored", "Afraid", etc[4]. This problem can be further remodeled as a regression problem where we can predict the degree of affinity for the movie instead of complete like/dislike.

Results:

As discussed above, we tried multiple classification models on various feature representations of the textual information in the reviews. Out of these SVM Classifier failed to even converge for all of our feature sets and hence we could not get a satisfactory answer for it. Among the remaining models, Logistic Regression model seemed to have best performance across all feature representations with classification accuracy around 89%. Also, k-Nearest Neighbors classifier had the worst accuracy of around 60% across all feature representations. The general order of performance for the model was LogisticRegression > NaïveBayes > SGDClassifier > RandomForestClassifier > kNNClassifier. For a given classifier, the model that performed best used a feature set of a mixture of unigrams and bigram.

References:

[1] Internet Movie Database – http://www.imdb.com/
[2] Cross Validation – Wikipedia - https://en.wikipedia.org/wiki/Cross_validation_%28statistics%29
[3] Scikit-learn API Reference: http://scikit-learn.org/stable/modules/classes.html
[4] Ortony, Andrew; Clore, G; Collins, A (1988). The Cognitive Structure of Emotions