**Aishat Mohammed**

**100134862**

**Project report**

**Sentiment classification using Naïve Bayes Multinomial**

**Introduction**

Sentiment classification done on a Mixed Pros and con file with the intention of labelling each line under Pros and Cons. Sentiment classification classifies text as positive and negative. Our training data consists of 2000 samples labelled Pros and Cons to denote positive and negative attributes accordingly. There are a set of products which may be from the same brand or different brands that a client is interested in. One query Product (Px) has a set of reviews and each review is a sequence of sentences. The data was acquired from reviews of epinions.com concerning things such as digital cameras, printers and Strollers.

In order to compare consumer opinions on a set of products, we need to analyze the reviews of each product Px to find all the explicit and implicit product features on which reviewers have expressed their opinions, and to produce the positive opinion set and the negative opinion set for each feature. Success will be determined by correctly classified instances in the training set. Aim to achieve at least 90% successful classification.

The file XmlToArrff is the program written to convert the data to arff format to be used in weka.

**Background**

The following research information used towards building my model was obtained from:

Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web"Proceedings of the 14th international World Wide Web conference (WWW-2005), May 10-14, 2005, in Chiba, Japan

**Product feature**: A product feature for Px is an attribute of the product that has been commented on in each sentence. Assume that sentence is labelled. If a feature appears in the sentence, it is considered explicit while if it doesn’t, but it is implied, it is considered explicit. Consider the example below:

"photo size restrictions"

The example above contains explicit features pertaining to size as it is stated in the sentence. Th sentence below however uses implicit features pertaining to size as they are implied but never stated.

“can fit in pocket”

**Opinion segment of a feature**: The opinion segment of a feature in a review is a set of consecutive sentences that expresses a positive or negative opinion on a feature of product Px. One sentence can be used to express opinions of more than one feature of product Px. See example below:

“The picture quality is good, but the battery life is short”

**Positive opinion set of a feature**: The positive opinion set of features of product Px is the set of opinion segments of that feature that expresses positive opinions about Product Px from all the reviews. The negative opinion set can be defined in the same way.

Reviews can be analyzed and visualised at different levels of detail. At the highest level (level 1) we can aggregate positive opinion sets and negative opinion sets of all features of the product Px to show an overall customer opinion on the product. At level 2, we can focus on each main component of the product and generate its positive opinion and negative opinion set. In visualization, we simply use the size of positive opinion and negative opinion set of each feature to show the number of positive or negative opinions on the feature. At level 3, we can study specific problems of each feature, e.g., “the picture is blurry” and “the picture is dark”.

**Method and Approach**

**Data preparation**: C++ code to convert given txt file to weka compatible arff file. This removes any quotation and replaces them with an apostrophe. Noisy data will be removed manually using a text editor and replaced with appropriate replacement. i.e. “&amp;amp;” gets replaces with an ‘&’ sign. I will also be removing any digits that work with overly specifying data like measurements e.g. (11 by 12 inch) as it hinders generalisation. Using StringtoWordVector filter in weka to produce attributes with an Iterated Lovin stemmer to convert all words to their base words. Also, Using N-gram to group certain words to develop stronger attributes.

Naïve Bayes is considered supervised learning but because of the assumption of conditional independence, it works well in text classification as it can use probabilities to determine class. Using Naïve Bayes multinomial classifier in a 10-fold cross validation in building model and predicting class. Another possible Solution would be to use the EM clustering to determine class. Compare models using correctly classified instances. Aim for above 90% but below 99% to avoid overfitting.

I use a method like the one used by “Opinion Observer: Analyzing and Comparing Opinions on the Web” but another solution could be the used as in *Murthy Ganapathibhotla and Bing Liu. "Mining Opinions in Comparative Sentences." Proceedings of the 22nd International Conference on Computational Linguistics (Coling-2008), Manchester, 18-22 August, 2008.*

* + **SEE META DATA REPORT IN FOLDER**

**Results**

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 1811 90.55 %

Incorrectly Classified Instances 189 9.45 %

Kappa statistic 0.8108

Mean absolute error 0.1247

Root mean squared error 0.2789

Relative absolute error 24.9561 %

Root relative squared error 55.7826 %

Total Number of Instances 2000

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.887 0.077 0.918 0.887 0.902 0.811 0.956 0.958 Pros

0.923 0.113 0.894 0.923 0.909 0.811 0.956 0.950 Cons

Weighted Avg. 0.906 0.095 0.906 0.906 0.905 0.811 0.956 0.954

=== Confusion Matrix ===

a b <-- classified as

870 111 | a = Pros

78 941 | b = Cons

These are the results generated from the training set used in weka. I was able to get a classification accuracy of 90.55%. See MDR for variable names and test set predictions. Average prediction error of 93%. Each classification compares the number of occurrences of a word from either the positive or negative opinion set. It calculates the probability of each classification based on the occurrences of either positive of negative words.

**Conclusion**

My model did well because of the reliance on probabilities in determining the classes. The preparation of the data also increased the accuracy of the model as it reduced the amount of garbage data present in the final creation of the model. Grouping and tokenizing the words in each sentence using N-Gram ensured that stronger attributes would be created to be used in analysing the classes of each review. The initial data is included in the zip folder in the original Xml format as well as the converted version in the arff format. The prep training data is the final training data used for building the model. It excludes parts of each review I have considered unnecessary in determining the classes. M success criteria successfully met as both prediction error and classification accuracy is above 90%. The classification is done using the number of occurrences of positive and negative opinion set of a feature available in each review.