Sentiment analysis of movie reviews using a naive bayes classifier with various features

Nikhil Raj Sriram (Roll no. 2022114003)

December 2023

1 Introduction

Sentiment analysis refers to the task of analyzing the sentiment expressed by a particular text. It finds great use in various mainstream statistical analyses. For example, companies often depend on sentiment analysis to gauge whether their products are doing well in the market or not. In this project, we will be conducting sentiment analysis on movie reviews.
A popular approach used for sentiment analysis is the "naive Bayes" approach. Given a set of documents to be classified, this involves representing each document as a bag of words, calculating the probabilities of occurrence of the words (for each class) independently and "naively" multiplying them to get the final likelihood for each class, after which the class with the higher likelihood is considered. Formally,

$$P(C_k|x) = P(C_k) \prod_{i=1}^{n} P(w_i|C_k)$$

or, equivalently, taking log for faster computation and using add-1 smoothing,

$$\log P(C_k|x) = \log P(C_k) + \sum_{i=1}^{n} \log \left(\frac{count_{w_i|C_k} + 1}{count_{C_k} + |V|} \right)$$

The feature used to calculate the probability of each word is typically the frequency of that word;however, in this project, we will be exploring other features to see how they affect the decision-making of the naive bayes classifier. Specifically, we will be looking at the following 2 types of features, and analyzing the results obtained:
1)Linguistically motivated feature: POS(parts-of-speech) tags
2)Features derived from a polarity lexicon VADER

1.1 Assumptions and Dataset used

All naive bayes classifiers implemented in this project use the feature of "binarization", i.e, only unique words in each test document are considered for

overall word counts during testing. This is because we observed that naive bayes classifiers with binarization consistently gave higher accuracies and better performance than those without binarization.
The primary dataset used for this project is the IMDB movie reviews dataset from the website kaggle.com. The dataset is not biased (i.e., it has an equal number of positive and negative reviews). Although this dataset contains 50000 movie reviews, for our purposes, we have used only 5000 reviews of these for training and testing, keeping in mind the amount of time it takes to run our classifiers on a dataset of this size. Additionally, we have used half of the chosen reviews for training and the other half for testing.

2 Naive bayes approach with POS(parts-of-speech) tags

Y. Wang(2017) proposed the idea of optimizing the traditional naïve bayes classifier by only considering frequencies of words of certain parts-of-speech for both training and test data in order to save computational resources that were being wasted by the traditional approach. He considered using only verbs and adjectives, only adjectives and adverbs, and all three. For this project, we will look at use of each of verbs, adjectives, and adverbs individually; the aim is to figure out which part-of-speech works best as a feature for the naïve bayes classifier.

2.1 Observations

	Actual Positive	Actual Negative
Predicted Positive	915	184
Predicted Negative	335	1066

Table 1: Confusion matrix of classifier using only adjectives

Metric	**Value**
Precision	0.83257506
Recall	0.732
Accuracy	0.7924
F1-Score	0.779054916

Table 2: Summary of Classification Metrics

	Actual Positive	Actual Negative
Predicted Positive	957	595
Predicted Negative	293	655

Table 3: Confusion matrix of classifier using only adverbs

Metric	**Value**
Precision	0.616623711
Recall	0.7656
Accuracy	0.6448
F1-Score	0.68308351

Table 4: Summary of Classification Metrics

	Actual Positive	Actual Negative
Predicted Positive	779	297
Predicted Negative	471	953

Table 5: Confusion matrix of classifier using only verbs

Metric	**Value**
Precision	0.72397769
Recall	0.6232
Accuracy	0.6928
F1-Score	0.6698194

Table 6: Summary of Classification Metrics

We can see that, out of the three parts-of-speech, the naive bayes classifier performs best using adjectives. This implies that adjectives carry the most sentiment out of the three.
Our claim is further substantiated by the following fact: We implemented the same classifier on a different dataset (the movie reviews dataset for assignment 5) and reached the same result; here also, adjective was the POS that had the highest accuracy. Additionally, we computed the frequency of different parts-of-speech for both the training data and the test data (in sorted order) and observed that, for both training and test sets, the order of frequency of the three parts-of-speech was: $verb > adjective > adverb$ which does not match the order of accuracies that we got from our analysis ($adjective > verb > adverb$); hence this is not a conclusion based off of biased data.

3 Naive bayes approach with additional features from polarity lexicon VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. To each word it assigns a dictionary consisting of information about the word: whether the word has a positive, negative or neutral sentiment, and the compound score of the word. The positive, negative and neutral keys of the dictionary are marked 1.0 if the word is positive/negative/neutral and 0.0 otherwise. The compound score is between -1 (most extreme negative) and +1 (most extreme positive).
As per the documentation of VADER, positive sentiment corresponds to a compound score $>= 0.5$, neutral sentiment corresponds to a compound score > -0.5 and < 0.5, and negative sentiment corresponds to a compound score $<= -0.5$. However, due to our training dataset having reviews labelled as only "positive" and "negative", in our use of VADER for sentiment analysis, we have assumed that positive sentiment corresponds to a compound score of above 0 and negative sentiment corresponds to a compound score of less than 0. Additionally, we found that omitting the "neutral" category alone, and assuming positive sentiment: compound score $>= 0.5$ and negative sentiment: compound score $<= -0.5$ made little to no difference in the output.

3.1 Incorporating features from VADER

The main aim is the following: To combine the existing features of the naive bayes classifier and the features in VADER in such a way that they complement eachother and increase the accuracy of our classifier. We achieved this by scaling the word frequencies of each word by parameters derived from VADER before computing the log likelihoods. We did this in 2 ways:

3.1.1 Scaling according to polarity score

Here, we scaled the word frequencies according to the polarity score assigned by VADER, i.e, the "compound" score between -1 and +1.
In our naive bayes classifier, each word in the vocabulary has a positive frequency (no. of occurrences of the word in positive training data) and a negative frequency (no. of occurrences of the word in negative training data). For each word, we first look at its negative frequency. If the word has a non-zero polarity score (i.e, if it is present in VADER), we check if the compound score is positive or negative. Let the compound score (polarity score) be denoted by x. If the polarity score is negative, we scale up the negative frequency of the word by a factor of x, i.e., we multiply the negative frequency by $1+|x|$. If the polarity score is positive, we scale down the negative frequency of the word by a factor of x, i.e., we multiply the negative frequency by $1-|x|$. Note that we used $|x|$ instead of x in this process because we are only interested in the magnitude of positivity or negativity.
A similar procedure is carried out for the positive frequency of the word; if the word has a non-zero polarity score , we check if the polarity score is positive or negative. If the polarity score is positive, we scale up the positive frequency of the word by a factor of x, i.e., we multiply the positive frequency by $1+|x|$. If the polarity score is negative, we scale down the positive frequency of the word by a factor of x, i.e., we multiply the positive frequency by $1-|x|$.
The idea is to adjust the impact of the words in positive and negative contexts by their meaning as per VADER, i.e., we want to make positive words in positive contexts more positive, negative words in positive contexts less positive, etc. On running this modified classifier, we observed the following results:

	Actual Positive	Actual Negative
Predicted Positive	1039	174
Predicted Negative	211	1076

Table 7: Confusion matrix of classifier using only adjectives

Metric	**Value**
Precision	0.85655399
Recall	0.8312
Accuracy	0.846
F1-Score	0.84368656

Table 8: Summary of Classification Metrics

This is a slightly better result than our original naive bayes classifier, which gave the following result:

	Actual Positive	Actual Negative
Predicted Positive	962	141
Predicted Negative	288	1109

Table 9: Confusion matrix of original classifier

Metric	**Value**
Precision	0.87216681
Recall	0.7696
Accuracy	0.8284
F1-Score	0.81767955

Table 10: Summary of Classification Metrics

3.1.2 Scaling according to sentiment

Here, we scaled the word frequencies according to the sentiment assigned by VADER. These were binary values, i.e, 1.0 if the word was positive/negative/neutral and 0.0 otherwise. Essentially, we followed the same procedure as we did when we scaled the word frequency according to polarity score, except that we used a constant factor for scaling instead of the polarity score. Let us call this factor k.

For words that were labelled as positive in VADER, we multiplied their positive frequency by k and divided their negative frequency by k. For words that were labelled as negative in VADER, we multiplied their negative frequency by k and divided their positive frequency by k.

At k=2, we observed the following results:

	Actual Positive	Actual Negative
Predicted Positive	1035	191
Predicted Negative	215	1059

Table 11: Confusion matrix of classifier with sentiment scaling (k-factor)

Metric	**Value**
Precision	0.84208880
Recall	0.828
Accuracy	0.8376
F1-Score	0.83602584

Table 12: Summary of Classification Metrics

We can see that this classifier also performs slightly better than the original naive bayes classifier.
On increasing the value of k, we observed an accuracy of 0.8308 for k=3, 0.8244 for k=4 and so on. We can hence conclude that there is an inverse relation between k and the accuracy, i.e.,

$$k \propto \frac{1}{accuracy}$$

3.2 Analysis of outputs

We analyzed the outputs of the original naive bayes classifier and the classifier using word frequencies scaled by polarity. We noted the following:

- A sarcastic sentence from the assignment-5 movie reviews dataset was given as input to both classifiers to test their ability to detect sarcasm. The sentence is as follows: "i guess i could say that i admire how filmmakers have become so much more devious in their product placement strategies . . . oops , did i say " admire " , i meant " am disgusted ". On running the classifiers, we noted that both classifiers classified the sentence as negative with nearly equal total positive and total negative log likelihood scores. Hence, both classifiers can detect sarcasm equally well.
- Additionally, we looked at the reviews that both classifiers misclassified. We noticed that, reviews that were classified differently by both classifiers were reviews that originally had either slightly more positive log likelihood or slightly more negative log likelihood, and the change in log likelihoods of certain words due to the polarity lexicon was enough to shift the result from one class to another. For example, one particular negative review was given the following positive and negative log likelihood sums respectively by the original classifier: (-298.87835353204036, -299.61194606946884). Clearly, the positive likelihood is more, so this was a misclassification. However, the modified classifier outputted (-300.0669902458598 -299.40244568597205) as positive and negative log likelihood sums respectively. Since the negative likelihood is higher than the positive likelihood, this was a correct classification. On looking at the log likelihoods of each word, it was found that words such as "odd" and "wrong" had their negative likelihoods increased and their positive likelihoods decreased as a result of the modification in the classifier and that this was enough to tip the scales the other way.

4 Conclusion

We can conclude that, although using specific POS tags did not increase our accuracy and performance, incorporating features from the polarity lexicon VADER did increase our accuracy and performance.

5 References/Citations

Y. Wang, "Advanced Naïve Bayes Algorithm Design with Part-of-Speech Tagger on Sentiment Analysis," 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), Dalian, China, 2017, pp. 1382-1385, doi: 10.1109/ICCSEC.2017.8446798.
https://vadersentiment.readthedocs.io/en/latest/index.html
Dataset used: https://www.kaggle.com/datasets/mejbahahammad/imdb-movie-reviews-dataset