International Journal of Engineering & Technology, 7 (2.33) (2018) 1064-1067

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Design, Improvement, Development, and Performance Analysis of a Collection of Model Developed From Naïve Bayes and Maximum Entropy Opinion Mining Classifiers for Movie Reviews

Dr. Lokesh A[1], Mr.Yerriswamy T[2], Mr.Venkatagiri J[3], Mr.Pradeep. M [4],

[1]Associate Professor, Dept. of ISE, Sri Venkateshwara College of Engineering, Bengaluru, Karnataka, India.,
[2]Assistant Professor, Dept. of ISE., Sri Venkateshwara College of Engineering, Bengaluru, Karnataka, India.,
[3]Assistant Professor, Dept. of CSE., Sri Venkateshwara College of Engineering, Bengaluru, Karnataka, India.,
[4]Assistant Professor, Dept. of CSE., M S Engineering College, Bengaluru, Karnataka, India.,
** Corresponding author E-mail: lokeshyadav.ka@gmail.com [1], ys_2123@yahoo.com[2] , venkatajgiri@yahoo.co.in[3], pradeep.pradeemb@gmail.com[4]*

Abstract

The internet is a basic platform for people from every one walks of life to interconnect and convey opinions on the topic of their choice. Almost every website asks for comments, suggestions and reviews. Exploring opinion and determining a person's views is itself a large subject in computer science, known as Opinion Mining, also called Sentiment Analysis There are different sentiment classifiers, the most admired of which are the Naïve Bayes classifier, maintain Vector Machines (SVM), Maximum Entropy classifier, to name a few. In this paper, here we are analyzing the efficient performance of the Naïve Bayes also about the Maximum Entropy classifiers. Here we analyze and examine how bigrams perform better than unigrams in sentiment analysis. We further propose a serialized ensemble model of the two as a hybrid algorithm and analyze its performance as well.

Keywords: *Reviews, measure,Sentiment analysis, Naïve Bayes, Maximum Entropy, bigrams, ensemble model, hybrid algorithm*

1. Introduction

The internet today has many public opinion and discussion forums. One such platform is IMDb, which is short for Internet Movie Database. Some research examples defines IMDb of data related to films, TV programs, and video games to be in online database. It holds ratings and reviews by users, which can be a mine of information.

Sentiment analysis deals with exactly this─ extracting relevant information from source materials and being able to judge whether the opinion is broadly positive or negative [2]. Maximum Entropy and Naïve Bayes classifiers are two of the most elementary and effective ones in sentiment analysis. The performance of these two algorithms is to be analyzed.

The task is to develop a hybrid algorithm along the lines of ensemble modeling that is based on the analysis of the two abovementioned sentiment classifiers. The hybrid algorithm uses the ideas of eliminating or discarding low-information features and using bigrams rather than unigrams. The ensemble used is based on the idea of nesting classifiers, i.e., feeding the output of one as input to another to aid in classification. Implementations of both the sentiment classifiers are done using the NLTK package [3]. IMDb is the source of the movie reviews.

2. Procedure

2.1 Data Collection And Pre-Processing

Data collection can be done using the corpus from the NLTK package. Real-world data is usually "noisy", i.e., it may include whitespaces, and punctuation, which can be removed before text classification. Also, converting the text to lower case makes it simpler to work with. Words of length lesser than three are removed too.

2.2 Feature Extraction

Feature extraction is done next, where the words that have meanings irrelevant to sentiment are excluded. For e.g., words such as "the" do not contribute to positive or negative sentiment and can be excluded. A list of such words is made. The document is checked against this and words in the document present in the list are excluded.

2.3 Training/Testing the Algorithm

First, the algorithm is trained with training data in Conformance with machine learning techniques. Here, features extracted from the training data are fed to the algorithm along with the respective labels, positive or negative. Training the

algorithm makes it function as a classifier. Now, features extracted from the test data are passed to the classifier and it makes a prediction about the sentiment of the review.

2.4 Analysis of Algorithm

Each of the reviews can be classified as true and false positives and negatives. Based on this classification, the algorithm's accuracy, precision, recall and F-measure are tested [4].

- accurateness is a measure or estimate of the correctness of predictions. A higher accurateness means better performance
- Precision measures or calculates the accuracy of the classifier. A superior precision specify the less number of false positives, while a minor precision means more number of false positives.
- F-measure provides a single measurement to precision recall.

3. Literature Survey

Machine learning, or supervised learning, is concerned with algorithms that learn and predict outcomes based on data given to it beforehand. So, there are essentially two parts in classifying data using the abovementioned algorithms-
Training the classifier, i.e., "teaching" the computer
Testing the algorithm, i.e., predicting the sentiment based on what it learnt

3.1 Bag-Of-Words Feature Extraction

In a bag-of-words model, a text document is viewed as a large group of its words, without preserving ordering as a sentence or grammar rules [5]. In this technique, every word is assigned a value of True by default and is made to False if it is specifically negative in meaning.

3.2 Naïve Bayes Classifier

The Naïve Bayes classifier [6] is a family of the simplest text classifiers based on the Bayes' theorem
This algorithm makes assumptions about all the features being independent, which accounts for the presence of the word "naïve". Bayes' theorem states that, given if it is given that event *x* has occurred, then the probability of an independent event *c* occurring is given by the formula-

$$P(c|d,\lambda) \stackrel{def}{=} \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c' \in C} \exp \sum_i \lambda_i f_i(c',d)}$$

$$P(c \mid x) = \frac{P(c) \times P(x \mid c)}{P(x)}$$

3.3 Maximum Entropy Classifier

In the framework of natural languages, entropy is a calculating of unpredictability or disorderliness of information content. [8] Consider the following examples-

- A poll on a political issue- The degree of unpredictability of the poll is a measure of the entropy.
- Predictions are specified in the language of English Even though we cannot predict the next word that would appear, quite predictably, the number of e's would be more than the number of z's.

highest Entropy classifiers are based on the Principle of Maximum Entropy [9], which specifies that, constarint to precisely stated prior information or data (in this case, training data), the probability distribution best representing the current state of knowledge is the one with the largest entropy. The various advantages [10] of the stated principle are "…in construction inferences on the source of one-sided information or data we must use that probability distribution which has maximum entropy issue to whatever is known or specied. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we do not have." This means that the features that have the highest unpredictability are the ones that must be considered while making any predictions about that data.

This classifier keeps track of how often a particular feature occurs for each class, i.e., positive or negative, and gives corresponding weightage to that feature. The ones that are highly biased towards a particular class are given a lower weightage. It then takes the ratio of the exponentiation of the product of weight and number of occurrences of each feature for that particular class to the summation of the same quantity for all classes. [11] The probability that the class *c* occurs for a given document *d* is given by-

TRAINMULTINOMIALNB($\mathbb{C}, \mathbb{D}$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}, c$)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

APPLYMULTINOMIALNB($\mathbb{C}, V, prior, condprob, d$)
1 $W \leftarrow$ EXTRACTTOKENSFROMDOC(V, d)
2 **for each** $c \in \mathbb{C}$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in W$
5 **do** $score[c] += \log condprob[t][c]$
6 **return** $\arg\max_{c \in \mathbb{C}} score[c]$

Fig 1: shows the algorithm according to the Stanford University NLP group [7].

Here, λ is the weight assigned to that feature; fi (c, d) corresponds to the number of occurrences of the feature f i in document d where fi is in a context of class c; and C is the set of all classes.

3.4 Bigrams

A bigram [12] is every sequence of two adjacent elements in a string of tokens, which are typically letters, syllables, or words; they are n-grams for n = 2
So, the probability that Wn occurs, given that Wn-1 has occurred is given by-

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

3.5 Elimination of Low-Information Features

In larger reviews, words with the highest probability of determining polarity are selected and the others are ignored. A

frequency distribution for overall frequency of words and a conditional frequency distribution for calculating frequency in each class are plotted. The chi square function is calculated from this, and the top few words are chosen. This is very effective in picking only the most important features and dropping the ones which are not so important [13].

3.6 Ensemble Modeling

Ensemble modeling is a technique in which machine learning here various learning algorithms are used to achieve better analytical performance it could be obtained from a number of of the essential learning algorithms. [14][15] also used when the two algorithms use vastly different techniques. The Naïve Bayes classifier uses the most fundamental laws of probability whereas Maximum Entropy uses logarithms and exponents to predict the outcome statistically using weighted features.

In the hybrid algorithm we used proposes of (shown in Fig. 2) a sequential ordering of classifiers and the output of one is appended to the feature list and fed as input to the next classifier. The output of the last classifier is the final output. It involves training both Naïve Bayes and Maximum Entropy classifiers using the same training data. On the testing data, first Maximum Entropy classifier is applied and the output of it is passed as an extra feature to the Naïve Bayes classifier, which gives the final output.

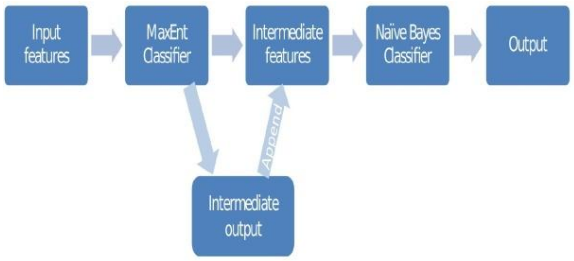

Fig 2: Ensemble model

3.7 Standard Criteria for Performance Evaluation

Here, *tpos* refers to true positives, *tneg* refers to true negatives, *fpos* refers to false positives and *fneg* refers to false negatives.

- Accuracy is the ratio of the true predictions to the total population of movie reviews.

$$Accuracy = \frac{tpos + tneg}{tpos + tneg + fpos + fneg}$$

- Precision is the ratio of true positives to all the reviews classified as positive.

$$\text{Precision} = \frac{tpos}{tpos + fpos}$$

- Recall is the ratio of true positives to all the reviews that were actually positive.

$$\text{Recall} = \frac{tpos}{tpos + fneg}$$

- F measure or F1 score is the harmonic mean of both.

$$\text{F1 score} = \frac{2 \times precision \times recall}{precision + recall}$$

- Conversely, negative precision, negative recall and negative F measure can be calculated

$$Accuracy = \frac{tpos + tneg}{tpos + tneg + fpos + fneg}$$

4. Results and Discussion

The Bag-Of-Words model is the easiest to work with for simple sentences. For e.g., the sentences “The movie was amazing”, and “What a nice movie!” are used to train the classifier. The list of positive words is, [“a”, “amazing”, “movie”, “nice”, “the”, “was”, “what”]. Then the sentence “Amazing movie!” is assigned to positive based on this model. It proposes that each single feature, or unigram, is evaluated and used for prediction.

Clearly, implementing the bag-of-words model is not as efficient as using bigrams. The reason for this can be summarized with the following example. Consider the phrase “not good”; the Bag-Of-Words model splits the phrase into two words, “not” and “good”, and evaluates their sentiments to be negative and positive respectively. Intuitively, the sentiment must be evaluated to be negative for the entire phrase. In the bigrams implementation, the sentiment of every set of two consecutive features is evaluated. So, in the analysis of “not good”, the entire phrase would be identified as an overall negative, reducing the ambiguity. The presence of such phrases in the movie reviews improves performance of the algorithm.

The Naïve Bayes classifier works on the Bayes’ theorem, making the assumption that all features are independent of each other. The document is broken up into feature bigrams and passed to the trained Naïve Bayes classifier. Despite the fact that the assumption made by this technique is difficult to implement in real life, an accuracy of about 81.6% is observed. This observation supports [16], where it is stated that Naïve Bayes can obtain high accuracies even though the assumption is ignored.

The MaxEnt classifier calculates the probability of a document belonging to a particular class by maximizing entropy, i.e., by assuming that while making predictions about the sentiment of the document, the features which have the highest amount of unpredictability must be considered. This ensures that no biases are introduced or specified into the system. The model makes no assumptions of the autonomy of features. So, bigram features are fed to the classifier and the features with the highest amount of disorderliness are allotted highest weights. Calculation of weights for each feature is computationally expensive, making this algorithm slower than the Naïve Bayes classifier. The hypothesis made by the Principle of Maximum Entropy holds as this algorithm turns out to be perfect to 80.2%.

Now, we develop a hybrid algorithm. First, the low-information features are eliminated. Only the high-information features are broken up into bigrams and then the ensemble model is used. These bigrams are first passed to the MaxEnt classifier and the output generated is appended to the feature list and passed into the Naïve Bayes classifier. The output generated is the final sentiment of the document. 92.4% of the predictions made by this algorithm are correct and this can be attributed to the elimination of low-information features, and to the ensemble model.
In large movie reviews, the overall sentiment of the tweet can be judged without having to analyze the entire review. Many words in the tweet may cause redundancy. This causes the algorithm to do more work than required and hence use more memory space. For e.g., if the words “magnificent” and “nice” are present in the same document, the presence of “magnificent” is enough to judge the polarity of the tweet because it shows a much higher positive sentiment than “nice”.

In such a case, “nice” is a low-information feature and is not required for judgment and prediction. Therefore it will be eliminated.

Ensemble modeling is used in machine learning when there are multiple algorithms to perform the same task and a blend of multiple algorithms can perform improved than any single algorithm itself. Since the Naïve Bayes and MaxEnt classifiers are vastly different in implementation, this model is suitable to design a hybrid algorithm. The serial ordering turns out to be beneficial because in cases where the first classifier predicts correctly, it assists the second classifier in prediction. In cases where the first classifier falters, the second one can correct it. Many studies have showed that, overall, ensemble modeling is more efficient than choosing only one technique and our project supports this.

The analysis of each of these algorithms is made by calculating and comparing positive and negative precision, recall, and F measure. A plot of the accuracies is also made.

This can be made easier by constructing a confusion matrix. A table of contingency or known as confusion template of matrix, is a comparison of actual and predicted positive and negative values [17][18]. Columns of the matrix represent predicted values and rows of the matrix represent actual values. Fig 3 shows the layout of a confusion matrix.

	Predicted	Predicted
	Positive	Negative
Actual	Tpos	Fneg
Positive		
Actual	Fpos	Tneg
negative		

Fig 3:. Confusion matrix

Negative recall, for e.g., can be calculated as follows:

$$Negative\ recall = \frac{tneg}{tneg + fpos}$$

The shaded regions are required to calculate it.

A comparison of the three algorithms can be summarized by the graph in fig 4

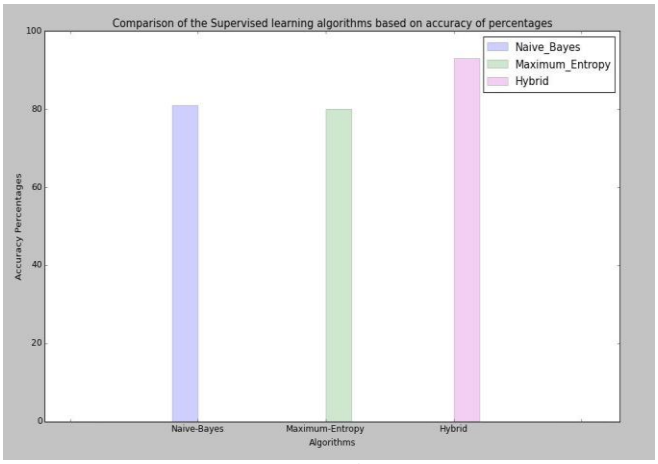

Fig 4:. Comparison of accuraccy

5. Conclusion

Sentiment analysis is the technique or method of predicting the polarity of a manuscript or document, usually with the help of machine learning algorithms. It comes under a broader field called Natural Language Processing, which deals with computers understanding raw data as given by human beings, for e.g., optical character recognition. Machine learning techniques are used when a prediction is to be made based on past knowledge and analysis, exactly as required by sentiment analysis.

Sentiment analysis is of enormous use to the fields of marketing and customer service. Once the sentiment of people is mined correctly, it is quite helpful in predicting what kind of goods would appeal to the consumer and what would not. In the context of movie reviews, a movie producers and directors, for e.g., can analyze what genre of movies is popular among a particular age group. This also helps in predicting trends.

In other fields, say politics, the outcomes of polls can be predicted, or the stock market futures and trends can be predicted for making the most profitable investments. Public approval or disapproval on any current news story or popular trend can be judged

References

[1] Wikipedia.org ‘Intenet Movie Database’, 2015 [Onlne]. Available:https://en.wikipedia.org/wiki/Internet_Movie_Database

[2] Jayashri Khairnar, Mayura Kinikar, ‘Machine Learning Algorithms for Opinion Mining and Sentiment Classification’, International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013

[3] Nltk.org ‘NLTK 3.0 Documentation’, [Online]. Available: http://www.nltk.org/

[4] Wikipedia.org ‘Precision and recall’, 2015 [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall

[5] Wikipedia.org ‘Bag-of-words model’, 2015 [Online]. Available: https://en.wikipedia.org/wiki/Bag-of-words_model

[6] Wikipedia.org ‘Naive Bayes Classifier’, 2015, [Online]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[7] Nlp.stanford.edu ‘Naïve Bayes text classification’, 2008, [Online] Available: http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

[8] Wikipedia.org ‘Entropy (information theory)’, 2015 [Online]. Available: https://en.wikipedia.org/wiki/Entropy_(information_theory)

[9] Wikipedia.org ‘Principle of maximum entropy’, 2015 [Online]. Available: https://en.wikipedia.org/wiki/Principle_of_maximum_entropy

[10] Bayes.wustl.edu ‘Information Theory and Statistical Mechanics’ 1957, [Online] Available: http://bayes.wustl.edu/etj/articles/theory.1.pdf

[11] Sentiment.christopherpotts.net, ‘Sentiment Symposium Tutorial: Classifiers’ 2011, [Online], Available: http://sentiment.christopherpotts.net/classifiers.html#maxent

[12] Wikipedia.org ‘Bigram’, 2015, [Online] Available: https://en.wikipedia.org/wiki/Bigram

[13] Streamhacker.com ‘TEXT CLASSIFICATION FOR SENTIMENT ANALYSIS – ELIMINATE LOW INFORMATION FEATURES, 2010, [Online]. Available: http://streamhacker.com/tag/feature-extraction/

[14] Wikipedia.org ‘Ensemble learning’, 2015, [Online], Available: https://en.wikipedia.org/wiki/Ensemble_learning

[15] AnalyticsVidhya.com, ‘Basics of Ensemble Learning Explained in

[16] Simple English’, 2015,[Online],Available: http://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/

[17] ‘The Optimality of Naive Bayes’, Harry Zhang, [Online], Available: http://www.aaai.org/Papers/FLAIRS/2004/Flairs04-097.pdf

[18] Wikipedia.org ‘Confusion Matrix’, 2015, [Online], Available: https://en.wikipedia.org/wiki/Confusion_matrix

[19] Data School ‘Simple guide to confusion matrix terminology’, 2014,[Online], Available http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

[20] Christine Day, ‘The Importance of Sentiment Analysis in Social Media

Analysis’, [Online], Available: