Sentiment Analysis Using Tuned Ensemble Machine Learning Approach

Pradeep Singh

Abstract With the recent emergence of Web-based applications and use of social networking sites, number of people are eager in expressing their views and opinions online. The sentimental analysis also referred to as opinion mining aims at processing user reviews (about products, movies, services, books, places, etc.). These reviews are often unstructured and need processing to evolve into the productive knowledge. Majority of the sentiment analysis works on the classification of opinion polarity with the use of simple classifiers. Handling diverse data distribution is one of the major issues that simple classifiers suffer. To cope up with the issue in this paper, we utilized the ensemble learners on the polarity prediction of the movie reviews. The proposed work processes the review data through some elementary steps that are conducted for the feature extraction in sentiment analysis. In addition to the feature extraction, we further perform the feature selection for the sake of dimensionality reduction. However, in contrast to the conventional simple learner, we applied the ensemble learner in the proposed model and evaluated its performance. To compare the ensemble model competence, we conducted the experiment on both individual as well as ensemble learner (random forest, AdaBoost, extra trees) and computed classification measures on both the model. IMDB dataset is used, and the polarity of a review, i.e., whether it is positive or negative, is predicted. With an extensive experimentation, it is found that results of ensemble classifiers are outperforming than individual learner in the classification of sentiment polarity.

Keywords Sentiment analysis · Ensemble learner · Tuning of parameter

1 Introduction

In our daily life, the opinions of customers and users of a product have a great influence on our decision making. These decisions may range from buying electronic

P. Singh (✉)
Department of Computer Science & Engineering, National Institute of Technology, Raipur, India
e-mail: psingh.cs@nitrr.ac.in

M. L. Kolhe et al. (eds.), *Advances in Data and Information Sciences*, Lecture Notes in Networks and Systems 38, https://doi.org/10.1007/978-981-10-8360-0_27

appliances or jewelry to taking review about the schools for children. Before the advent of Internet, opinions on products and services are taken from friends, relatives, or consumer reports. Now in the Internet era, it is much easier to collect diverse opinions from various people across the world. The review sites (CNET, Epinions, etc.), e-commerce sites (Flipkart, Amazon, Snapdeal, eBay, etc.), online opinion sites (TripAdvisor, Rotten Tomatoes), and social networking sites (Facebook, Twitter, etc.) are referred to get opinion about how a particular product or service is provided to them. Similarly, most of the organizations use opinion polls, surveys, and social media as a mode to obtain feedback on their products and services [1]. Sentimental analysis is the branch of text mining which processes these reviews computationally for identifying and categorizing opinions, sentiments, attitudes, subjectivity, views, evaluations, appraisal, emotions, etc., stated in a textual form as positive, negative, or neutral.

In sentimental analysis, classification is done according to different criteria such as polarity of the sentiment (negative, positive, or neutral), whether the opinion is in support or opposition of a service, number of pros and cons in the reviews, whether the user agrees or disagrees with some particular topic. According to [2] sentiment, classification is of three levels. Document level: The sentiment is evaluated by taking the whole document as one information unit. Sentence level: The sentiment is evaluated by taking each sentence as an individual unit. Aspect level: According to [3], in this level, the sentiment is evaluated by taking the polarity of each aspect of the review such as screenplay, acting skills, and direction for a movie. The sentiment analysis of movie reviews can be at document or aspect level [4].

Machine learning algorithms play a critical role in the document-level polarity prediction. The choice of supervised learning algorithm in the classification problem is cumbersome due to the wide availability of the candidate. Single classifiers are producing the good classification rate, but the presence of differences in the data distribution between train and test instances makes the simple learner to perform poorly. Thus, ensemble learner has the ability to handle the data distribution efficiently and with superior performance delivered by multi-classifier model in the other application makes an appropriate choice to be elected as alternative for the single classifier. This paper aims at classifying the document-level sentiments using machine learning algorithms and compares the performances of simple and ensemble learners. In addition to it, we also aid the joint contribution of unigram and bigrams features along with the parts of speech (POS tagging) for feature extraction, whereas feature selection algorithm is used for the efficient prediction of polarity with minimum data handling complexity. We considered random forest, multinomial naive Bayes, Bernoulli naive Bayes, SGD, SVC, and NuSVC for the sake of performance comparison between simple and ensemble learner. The contribution of the paper can be summarized as follows:

- Use of unigrams features leads to false analysis. Hence, we focus on the combination of uni- and bigrams which increase the efficiency.
- Use of feature selection reduces the total number of features (words), thereby decreasing the time for overall computation.

- Use of POS tagging eliminates many useless pronouns, propositions.

The remaining part of the paper is organized as follows: Sect. 2 presents contextual information related to the past work done. Section 3 presents the proposed sentiment analysis method covering common problems listed during the study of related work. Comprehensive discussion on the supervised learning on both simple and ensemble learners is presented in Sect. 4. In the last Sect. 5, we conclude the overall performance of the learners in sentiment analysis.

2 Related Work

Pang et al. considered aspect of sentiment analysis using categorization with positive and negative sentiments [5]. They used different machine learning algorithms (classifiers) like support vector machine, Naïve Bayes. They classified it using unigram features, bigram features, and by combining both unigram and bigram features. To realize the algorithm, they use bag-of-words (BOW) in their algorithms and found SVM with good classification rate.

Salvetti [6] discussed an overall opinion polarity (OvOP). Here hypernym given by word net and parts of speech tagging acts as lexical filter. The results from word net are less accurate than POS filter. Their work has shown good result in Web data. Mullen [7] applied SVM where values are given to selected words and are pooled to make a model for the classification. Features which are close to the topic are allocated with higher values. They gave comparison of their approach and hand annotation. Their approach gave better results.

Matsumoto [8] used syntactic relation among words in document-level sentiment analysis. The frequent word subsequence and dependency sub trees are extracted from the sentences and used as input features for support vector machines (SVM) machine learning algorithm. They performed their experiment on IMDB and polarity dataset. Liu [9] proposed multi-label classification. They used 11 methods compared on 2 micro-blog dataset and 8 evaluation matrices. Lin et al. [10] performed an empirical study of sentiment categorization on Chinese hotel review. A Chinese corpus, MioChnCorp, with a million Chinese hotel reviews is collected. A word2vec model is trained using MioChnCrop to represent words and phrases in Chinese hotel domain. Their experimental results indicate that the more data produces the better performance. They also used word embeddings which represent each comment as input in different machine learning methods like SVM, Logistic Regression, Convolutional Neural Network (CNN) and ensemble methods for sentiment classification.

The literature review identifies the vague issues that remained untouched during the problem solving. From the review, we find the following issues:

- Most of the work done in the field of sentimental analysis is done considering unigram approach which may not include all the features accurately.

- Much work has been done on the sentiment analysis, but they have used a lot of features for the feature extraction part. A lot of time is wasted during the processing of the whole review data. If there is a big dataset then it wastes a lot of time.
- Majority of the model uses single classifiers which performs poor in the case of diverse data.

In this paper, we aim at providing the solutions for the issues that were found during the survey. The objective is achieved by collecting sufficiently large amount of data consisting of reviews, preprocessing it applying feature selection in order to extract the features with high frequencies and reducing large number of words to a limited features and passing these inputs to the ensemble classifiers; additionally, we also performed the simulation with the simple classifiers for the sake of performance comparison.

3 Methodology

The process followed in this paper is shown in Fig. 1. In this model, we acquired the textual datasets and applied preprocessing and POS tagging on them. This preprocessed datasets is divided into train and test data (if not given explicitly) by tenfold cross-validation, for the further use of train data in learner. Later, we input the test data to predict the sentiment. The original sentiment of the reviews is compared with the obtained sentiment to calculate accuracy, precision, recall, and f-measure.

To evaluate the performance of simple and ensemble learner on sentiment analysis, we carried the elementary steps that are followed in the opinion mining process. Firstly, IMDB reviews used in this project were acquired from [11]. This dataset consists of 25,000 reviews. Apart from the text data, the dataset has numeric, acronym, and HTML tag contents in it. Thus, in order to have good classification it is necessary to eliminate these entries from the dataset.

To remove such entries, we apply preprocessing on the entire dataset. This preprocessing involves removal of acronyms, numeric letters, and HTML tag. In addition to this, we also eliminate the stop words (words which do not contribute to the sentiment analysis).

An acronym is an abbreviation, and it is generally used in content published on the Internet. Users are often drawn toward brief words, and using acronyms is one way of ensuring that the sentence still grabs the attention of the reader despite the fact that it is short. For example, DND stands for 'Do Not disturb' while OMG stands for 'Oh My God.'

Reviews contain numeric characters which do not affect the sentiment of words. Hence they are removed. Stop words such as 'comma,' 'full stop,' parenthesis, question marks, exclamations, and special characters such as *, @, $, # are eliminated. Also the reviews that contain references to contain links are useless. Hence they are eliminated.

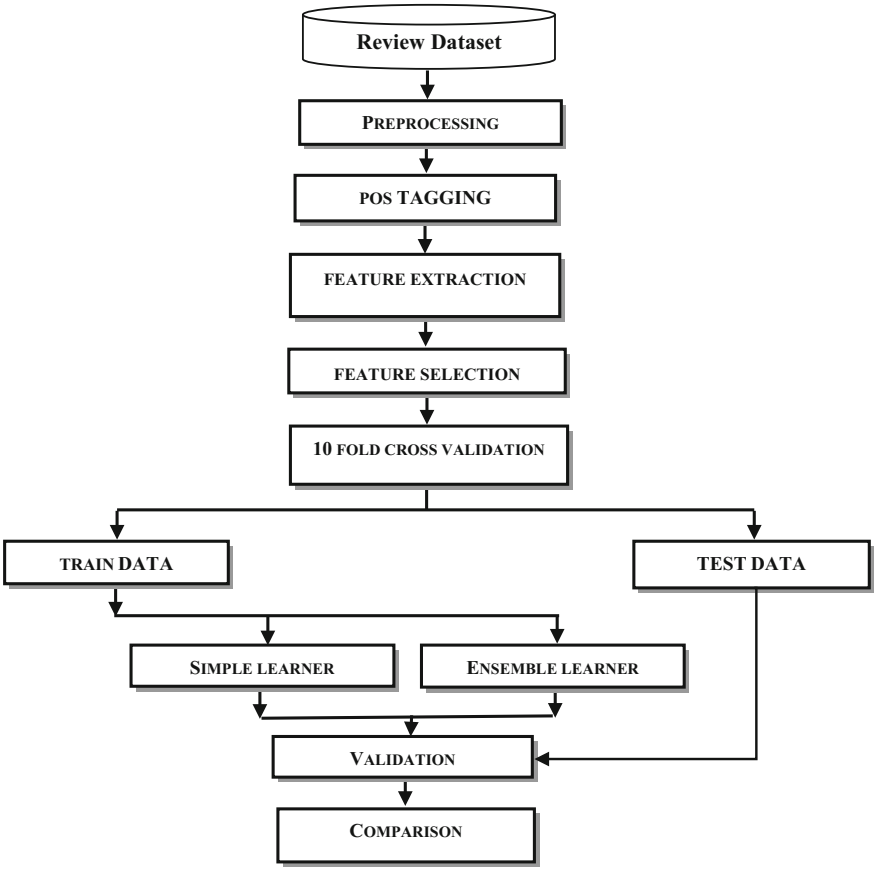

Fig. 1 Process flow diagram

After preprocessing, parts of speech (POS) tagging is performed in the processed data. The review carries the word, and tagging assigns parts of speech to each word in the dataset, i.e., whether the word is a noun, pronoun, verb, adjective, or adverb. POS tagging can be used as feature reduction by selecting only few elements in parts of speech; i.e., pronouns and prepositions can be eliminated from the data which reduce the number of features. In our work, we used a combination of verbs, nouns, adjectives, and adverbs and eliminated the remaining features.

Feature extraction is an attribute or dimensionality reduction process [12]. It is used to extract a subset of features from the original feature set by means of some functional mapping retaining as much info in the data as possible as mentioned in [13]. Feature extraction is used to transform the actual attributes. The transformed features are linear combinations from the original attributes. Models constructed based on extracted features tend to be of higher quality, because the information is

described by less, more meaningful attributes. Feature extraction techniques used in our project are:

Bigrams: Bigrams are nothing but a sequence of two adjacent words in the sentence. Examples of bigrams are 'not good,' 'very bad.' Bigrams are very useful in assigning correct polarity as the string of 2 words can increase the efficiency. Bigram is an N-gram for $N = 2$.

Term frequency-inverse document frequency (TF-IDF): Term frequency is the amount of times a specific word or term appears in the text. Inverse document frequency measures the existence of a particular word in all documents. According to [14], the values of TF-IDF are directly proportional to the term frequency; i.e., it increases as the frequency of a word in the document increases.

Count vectorizer: It implements both counting of occurrence and tokenization in a single class. It converts the entire text documents into a sparse matrix representation.

4 Supervised Learning Algorithms

We employed nine classifiers from the scikit-learn package [15] from Python, four from the simple learner, and four from the ensemble learners. The hyper-parameters of classifiers are tuned using randomized parameter optimization.

Classifier	Type	Description
Random forest	Ensemble	It uses a huge number of individual, unpruned decision trees
AdaBoost [16]	Ensemble	Amount of focus is quantified by a weight that is allocated to every pattern in the exercising set
Extra trees [17]	Ensemble	Randomizing tree building in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree
Gradient boosting [18]	Ensemble	'boosting' many weak predictive versions into a strong one, available as ensemble of weak types
Support vector machines [19]	Simple	Kernel-based method uses hyperplane that separates the classes and has the largest distance between border line data points
Naïve Bayes variants, Multinomial Naive Bayes (MNB) Bernoulli Naive Bayes [20]	Simple	The probabilistic model of Naïve Bayes is originated from Bayes theorem
Stochastic gradient descent [21]	Simple	Stochastic approximation of gradient descent optimization approach which is used to minimize an objective function
Logistic regression	Simple	Logistic regression is a linear model for classification also known as maximum entropy classification (MaxEnt) or the log-linear classifier

5 Experimental Results

When we use a classifier model, we always need to find the exactness of that model as the result obtained forecasts from all expected results. This is called classifier accuracy. When we have to choose whether it is a sufficient model to take care of, accuracy is not the only metric for assessing the viability of a classifier. Two other important measurements are precision and recall. A confusion matrix is a summary of prediction results of a classification problem (Table 1).

A false positive error, or in short false positive, commonly called a 'false alarm,' is a result that specifies a given condition has been satisfied; when it actually has not been satisfied, i.e., erroneously a positive effect has been assumed. A false negative error, or in short false negative, is where a test result indicates that a condition failed; while it actually was successful. True positives are relevant items that we correctly identified as relevant. True negatives are irrelevant items that we correctly identified as irrelevant. A confusion matrix C is such that $C_{(i,j)}$ is equal to the number of observations known to be in group i but predicted to be in group j. A confusion matrix is used to describe the performance of the classifier.

Accuracy is how close a measured value is to the actual (true) value. It is the proportion of instances whose class the classifier can correctly predict. It can be calculated as shown in Eq. 1.

$$\text{Accuracy} = \frac{T_\text{p} + T_\text{n}}{\text{Total number of samples}} \tag{1}$$

where T_P denotes the number of true positives and T_n is the number of true negatives.

Precision measures exactness of a classifier. High precision indicates that there are less number of false positives; likewise, a lower precision indicates more number of false positives. Precision is defined as the ratio of number of true positives over the number of true positives plus the number of false positives [22]. Its formula is shown in Eq. 2.

$$P = \frac{T_\text{p}}{T_\text{p} + F_\text{p}} \tag{2}$$

where F_p denotes the number of false positives.

Recall is used to measure completeness, or sensitivity, of a classifier. Increasing the value of recall often decreases precision because it gets increasingly harder to be more precise as sample space increases. Recall is defined as the number of true

Table 1 Confusion matrix

Actual class	Predicted class	
	Yes	No
Yes	T_P	F_N
No	F_P	T_N

Table 2 Performance metrics—random forest classifier

Feature type	IMDB dataset			
	Accuracy	Precision	Recall	*F*-measure
UNIGRAM	84.7	83.48	85.56	84.51
UNI + BI	84.73	83.72	85.45	84.57
Parameter tuning (UNI)	85.33	84.65	85.81	85.23
Parameter tuning (UNI + BI)	85.64	85.15	85.99	85.57

positives over the number of true positives plus the number of false negatives. Its formula is shown in Eq. 3.

$$R = \frac{T_{\mathrm{p}}}{T_{\mathrm{p}} + F_{\mathrm{n}}} \tag{3}$$

Precision and recall can be combined to produce a single metric known as *F*-measure. It is weighted harmonic mean of precision and recall. Its equation is shown in Eq. 4.

$$F1 = 2\frac{P \times R}{P + R} \tag{4}$$

where P is the precision and R is the recall.

We compare the performance of the classifiers that we used based on their precision, recall, *F*-measure, accuracy, and confusion matrices.

We train each of the classifiers using the two datasets individually. First, we test all the classifiers on each training set one at a time and then we test it on the test set. Table 2 summarizes the results of all the experiments performed on IMDB dataset.

The results of different approaches on the IMDB dataset are shown in Table 3. Initially, the analysis is done only by considering the unigram words and applying the random forest machine learning algorithm. We used noun, verb, adjective, and adverb from our data and removed all the unnecessary words. On applying single machine learning algorithms (Table 3) on the IMDB dataset, we observed good results in NuSVC and stochastic gradient descent method. On applying on ensemble classifiers (Table 4), extra trees classifier using (uni + bi) has performed outstanding.

After applying the parameter tuning on the random forest classifier (Table 2), the accuracy of the sentiment prediction has increased by 1%. The parameters used in our process are n_estimators = 100, max_features='sqrt', oob_score='true', n_jobs = −1, random_state = 50.

Table 3 Performance metrics—single classifiers

Classifier and feature type	IMDB dataset			
	Accuracy	Precision	Recall	*F*-measure
NuSVC UNIGRAM	87.452	89.032	86.305	87.647
NuSVC (UNI + BI)	87.342	88.804	86.281	87.524
SGD UNI	90.532	87.096	93.522	90.195
SGD UNI + BI	90.530	87.096	93.519	90.193
MNB UNI	83.968	81.304	85.879	83.529
MNB UNI + BI	84.728	83.648	85.495	84.561
BNB UNI	84.192	81.7	85.985	83.788
BNB UNI + BI	85.022	84.296	85.538	84.912
LR UNI	84.369	82.485	85.715	84.069
LR UNI + BI	85.084	84.4	85.571	84.981

Table 4 Performance metrics—ensemble classifiers

Classifier and feature type	IMDB dataset			
	Accuracy	Precision	Recall	*F*-measure
AdaBoost (UNI)	83.468	84.176	81.395	82.762
AdaBoost (UNI + BI)	83.8	84.28	81.86	83.05
ExtraTrees (UNI)	86.128	84.528	87.32	85.90
ExtraTrees (UNI + BI)	86.496	85.11	87.535	86.306
RandomForest (UNI)	84.7	83.48	85.568	84.511
RandomForest (UNI + BI)	84.736	83.72	85.456	84.579
Gradient boosting (UNI)	81.096	86.336	78.146	82.037
Gradient boosting (UNI + BI)	81.14	86.208	78.274	82.049

6 Conclusions

In this paper, intensive experiments were performed to predict movie reviews using different supervised machine learning algorithms like Naïve Bayes (NB), stochastic gradient descent (SGD), support vector machines (SVM), random forest, AdaBoost, extra trees, and gradient boosting. We applied both unigram, unigram + bigram approach. The learner performed better when using unigram + bigram approach than the unigram feature. Parameter tuning has been applied to improve the accuracies.

Processing of Twitter reviews may have some issues because they contain emojis and smileys (they hold important information whether it is a positive tweet or negative tweet) which are not processed in our approach. Some words like 'greatttt', 'fineee' are also processed using the stemmer because those features should not be missed. The accuracy of the prediction may increase with various preprocessing techniques and machine learning algorithms. Taking the above limitations into consideration, further work can be performed in order to improve the accuracy of sentiment prediction.

References

1. Fernández-gavilanes M, Álvarez-lópez T, Juncal-martínez J, Costa-montenegro E, González-castaño FJ (2016) Unsupervised method for sentiment analysis in online texts, vol 58, pp 57–75
2. Medhat, W., Hassan, A., Korashy H (2014) Sentiment analysis algorithms and applications: a survey. Ain Shams Eng 5(4):1093–1113
3. Parkhe V (2014) Aspect based sentiment analysis of movie reviews
4. Singh VK, Piryani R, Uddin A (2013) Sentiment analysis of movie reviews, pp 712–717
5. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up: sentiment classification using machine learning techniques. Proc Conf Empir Methods Nat Lang Process, 79–86
6. Salvetti F, Lewis S, Reichenbach C (2004) Automatic opinion polarity classification of movie. Color Res Linguist 17(1):2
7. Mullen T, Collier N (2004) Sentiment analysis using support vector machines with diverse information sources. Conf Empir Methods Nat Lang Process, 412–418
8. Matsumoto S, Takamura H, Okumura M (2005) Sentiment classification using word subsequences and dependency sub-trees. In: Proceedings of 9th Pacific-Asia conference advances in knowledge discovery and data mining, vol 059, pp 301–311
9. Liu SM, Chen J-H (2015) A multi-label classification based approach for sentiment classification. Expert Syst Appl 42(3):1083–1093
10. Lin Y, Lei H, Wu J, Li X (2015) An empirical study on sentiment classification of Chinese review using word embedding. In: 29th Pacific Asia conference on language information and computation, pp 258–266
11. http://ai.stanford.edu/~amaas/data/sentiment/
12. https://docs.oracle.com/database/121/DMCON/feature_extr.htm#DMCON268
13. Pechenizkiy M, Puuronen S, Tsymbal A (2001) Feature extraction for classification in the data mining process PCA-based feature extraction feature extraction for a classifier and dynamic integration of classifiers. Int J 10:271–278
14. Tripathy A, Agrawal A, Rath SK (2016) Classification of sentiment reviews using n-gram machine learning approach. Expert Syst Appl 57:117–126

15. Pedregosa F (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
16. McCallum A, Nigam K (1998) A comparison of event models for Naive Bayes text classification. AAAI/ICML-98 work learning for text categorization, pp 41–48
17. Kibriya AM (2004) Multinomial Naive Bayes for text categorization revisited. Adv Artif Intell, 488–499
18. Mason L, Baxter J, Bartlett P, Frean M (1999) Boosting algorithms as gradient descent. Nips, 512–518
19. Fradkin D, Muchnik I (2006) Support vector machines for classification. Discret Methods Epidemiol 70:13–20
20. http://sebastianraschka.com/Articles/2014_naive_bayes_1.html
21. https://books.google.co.in/books?id=48u5BQAAQBAJ&pg=PA369&lpg=PA369&dq=Stochastic+Gradient+Descent
22. http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/