International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Opinion Mining Classification using Naive Bayes Algorithm

Raghavendra Vijay Bhasker Vangara, Kailashnathan Thirupathur, Shiva Prasad Vangara

Abstract*: With the recent advancement in the field of online services, the importance of a review for a product has also gone up. In this paper we focus on the aspect of reducing the time and effort for the user by recommending the best product to him. For this to be achieved, this paper proposes a Naive Bayes Classifier which labels the reviews accurately and combines the reviews to give a final rating to the product. The amazon product review data consisting of both negative and positive reviews was used for training and testing purposes. The model's performance is evaluated, and results are analysed.

Keywords: Natural Language Processing, Sentiment Analysis, Opinion Mining, Machine Learning.

I. INTRODUCTION

Sentiment analysis is also known as opinion mining. Sentiment analysis in a broad sense is a way to know a user's emotion(sentiment) using his features. There are three kinds of emotions: positive, negative and neutral. Positive emotion can be happy, liking something etc. Negative emotions are angry, unhappy, etc. Neutral emotions are when a person shows no expression. In sentimental analysis we categorize these features in one of the two ways : 1) Image of user's face expressing his emotion, in this method an algorithm is trained using a large no. of images of facial expressions and is used to test new facial images to get their emotions. 2) A text written by the user expressing his emotion in it. It can be a text sent to someone or his search in the internet or it can be his review for a product or can be anything. Over the past few years, a lot of work has been done in this area of research. Many Researchers have already developed supervised machine learning algorithms to go through large datasets and learn to differentiate between different moods/emotions.

The main concern in this paper is regarding the second category mentioned above. There are many applications of sentiment analysis using the text of a user. One of the main applications is online review/feedback services for different products or services provided. Amazon, Flipkart are few of the online shopping service providers which uses sentiment analysis to differentiate between the positive and negative reviews. Nowadays, online shopping is preferred more than the traditional way of going to the store due to the lack of time for the employed users. And nowadays people are considering the reviews of other users. Therefore, accurate review is much more needed than ever before. For a user, reading through a thousand reviews is a very time consuming and is also a very tire job. Hence, analysing the reviews and combining them to give an overall rating for the product is also necessary. A positive rating i.e. a large no. of positive reviews means identifying the product as authentic. Conversely, a negative review identifies a products quality to be not so good and hence, can cause sales loss. Therefore, accuracy is all the more important because if a positive review is labelled as negative, this can be a serios problem for the product's owner and equally a negative review labelled label as positive will be a loss on user/customer side.

This paper uses automatically labelled datasets instead of manually labelled ones to reduce the time-consuming effort and it is also incorporated with the aspect-level sentiment analysis method for more accuracy. As in [3] & [4], this paper proposes the same feature extraction technique which considers the different aspects in the paper combined with the naive bayes algorithm to label the user reviews accurately. This method is expected to identify fake reviews automatically.

The rest of paper organises as follows: section 3 describes the related work and the section 4 gives a detailed word of methodology and section 5 explains implementation process and section 6 illustrates results.

1.2 OBJECTIVES

- To analyse the users or customer reviews and assessing the user's emotions on the specified user feedbacks
- Classify the buyer's reviews assessment (The clients can evaluate or the product level, whether is a great or not etc), and audit using machine learning model
- To analyse the distinct user feedback and mine the emotional words to obtain the user distinct importance about the brand
- To mine the customer reviews and emotions
- To implement a Naive Bayes algorithm for automatic classification of text into positive or negative or neutral
- To improve the accuracy of proposed model than the existing system.

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Raghavendra Vijay Bhasker Vangara*, Department of Mathematics and Computer Science, University of Missouri-St. Louis, Missouri, USA. Email: rv6dc@mail.umsl.edu
Kailashnathan Thirupathur, Department of Computer Science, University of Bridgeport, Connecticut, USA. Email: kailashanath@gmail.com
Shiva Prasad Vangara, Department of Information Systems, Indiana Tech University, Indianapolis, USA. Email: shiva27389@gmail.com

1.3 PROPOSED SYSTEM

The Proposed model is the Naive Bayes algorithm

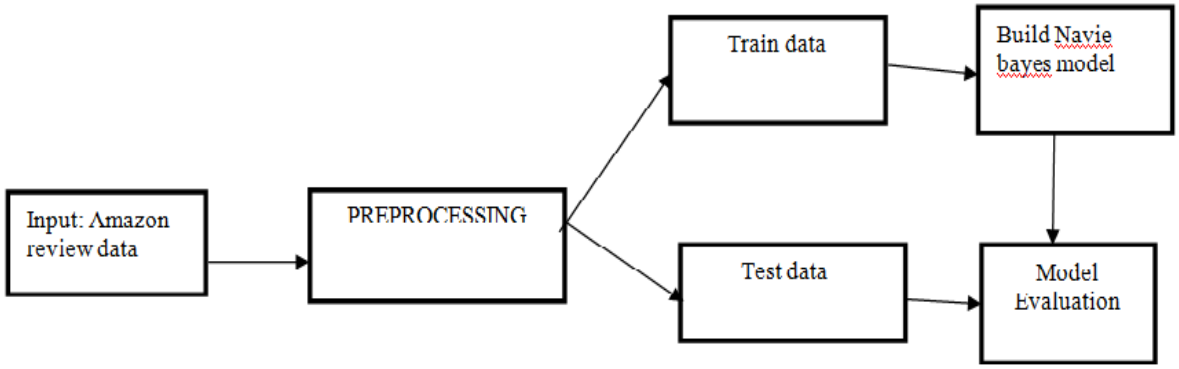

Figure 1: Block diagram of proposed model

II. LITERATURE SURVEY

There is a lot of work done already in this particular area of interest and as mentioned earlier there are a lot of service provider which are already using these techniques. But still, a lot of effort is being put in because of the increasing usage of the online services along with their reviews and the low accuracy of these machine learning algorithms.

In [1], three types of feature extraction classifiers were used. 1) Phrase level, where phrases/sentences were used as features. 2) single-word level, in this important context-based words are used as features. 3) multi-word level, group of words which collectively form a feature. They use Naive Bayes algorithm for learning. They observed that the phrase level feature extraction method gives more accurate results than the other two.

Haque et al [2] proposed a supervised learning model. Bag-of-word method and Chi-square methods were combined, and the results were obtained as they had expected. They got 90% accuracy with F1 score. Similarly, they got 90% precision and recall.

Xiao ma et al [3] were the first to propose a multi-aspect model for sentiment analysis. Different aspects in the same sentences were also consider getting more accurate emotion result. In the first part of the work, different parts of the same sentences were considered individually, and their emotions were obtained whether positive or negative. In the second part, emotion of each of those parts were processed with other parts of the same sentence were processed to get a better emotion result for the sentence. For e.g. 'I am happy that my friend is unhappy'. In this sentence the first half is positive and the second part is negative. This is obtained in the first stage of the algorithm. In the second half of the algorithm, they both are seen together which give us the final results which is a negative emotion.

Liu and shen [4] did a similar work as Xiao ma et al [3], RNN was used but couldn't capture local context. Similarly, CNN was used to overcome the drawbacks of RNN but had other issues. A new method was proposed which is named as Gated Alternate Neural Network (GANN). Similar to [3] the distance between the two words and their aspects is related to obtain better labelling of the sentences.

This paper [7] discusses the adaptation of simple MNB text classification for sentiment analysis. The main contribution is that with the help of MNB because of that the bit of have issues can work abnormal, slow and too much overfit while the data sets are very few, and multinomial performance is better than Bernoulli when it compared to Bernoulli model. In detail, multinomial is every time a preferable method for any type of text classification (spam detection, topic categorization, sentiment analysis) as taking the frequency of the word into consideration and get back better accuracy than just checking for word occurrence.

This study [9] describes a technique called Tree Fast K-Nearest-Neighbor (TFKNN), which allows one to quickly search for the nearest k neighbors. The existing KNN has a fatal error because the similar competitive computation time is applied to the KNN algorithm for classification which is too long for the large size of text and big samples. The author has introduced a new algorithm to sort out the computational cost, which is the TFKNN algorithm that enables one to immediately close the neighbor. Besides that, the SSR tree is developed to find for nearby neighbors, all child nodes of every non-leaf node are ranked as per the distance among their principal points and the focus of their parents.

III. METHODOLOGY

Naive Bayes algorithm is the supervised machine learning algorithm that uses Bayes theorem which relies on conditional probability. It is a very famous algorithm used for sentiment analysis. This algorithm predicts the tag of text and calculates the probability for every tag of a given text and then the highest one is the output.

Step 1: Combining the probability distribution of P with a fraction of documents belonging to each class. For class **m**, word **n** at a word frequency of **w**:

$$p(m) \propto \pi_m \prod_{n=1}^{|v|} p(n|m) w_n \quad (1)$$

Step 2: To avoid underflow, we will use the sum of logs

$$p(m) \propto \log\left(\pi_m \prod_{n=1}^{|v|} p(n|m) w_n\right) \quad (2)$$

$$p(m) = \log \pi_m + \sum_{n=1}^{|v|} w_n \log\left(p(n|m)\right) \quad (3)$$

Step 3: If a word appears again, the probability of it appearing again goes up. To smooth this, we take the log of the frequency:

$$p(m) = \log \pi_m + \sum_{n=1}^{|v|} \log (1 + w_n) \log (p(n|m)) \quad \text{_____(4)}$$

Step 4: To take stop words into account, we will add an Inverse Document Frequency (IDF) weight on each word:

$$t_j = \log \left(\frac{\sum_{k=1}^{K} doc_k}{doc_j} \right) \quad \text{_____(5)}$$

$$p(m) = \log \pi_m + \sum_{n=1}^{|v|} w_n \log (t_j \ (p(n|m)) \quad \text{__(6)}$$

Step 5: Even though the stop words have already been set to 0 for this specific use case, the IDF implementation is being added to generalize the function.

IV. IMPLEMENTATION

A. Dataset:
The amazon product review data is considered.

B. Pre-processing:
In this phase of opinion mining, raw data is considered and processed for feature extraction.
Then it is divided into following steps:
Tokenization: Here the sentences are categorized into words or tokens by ignoring whitespaces and other symbols
Stop Word Removal: It ignores articles like "a, an, the".
Stemming: Decreases the tokens or words to its root form.
Case Normalization: Changes the whole document either in lower case letters or upper-case letters.
Punctuation marks removal: Punctuation like commas, quotes, question marks are removed

C. Feature Engineering:
The transformation of raw data input to represent features which are learned by the machines. It deals with the identification of types of features used for opinion viz. term frequency, term co-occurrence, OS information, Opinion word, Negation, Syntactic Dependency). It is used to select good features for opinion classification in the following ways like Information gain, Odd ratio, Document frequency, and Mutual Information. It calculates weight for ranking the characteristics using Term presence and term frequency and Term Frequency and Inverse document frequency (TF-IDF). It lowers the vector size to optimize the performance efficiency of a classifier.

D. Machine learning algorithm:
The Naive Bayes model is built on the trained data. Before the model is built the data is divided into 80% train data and 20% test data randomly. The test data is applied on the trained model to predict results.

E. Evaluation:
The model is evaluated using the confusion matrix. The model is evaluated between actual data and predicted data.

V. RESULTS

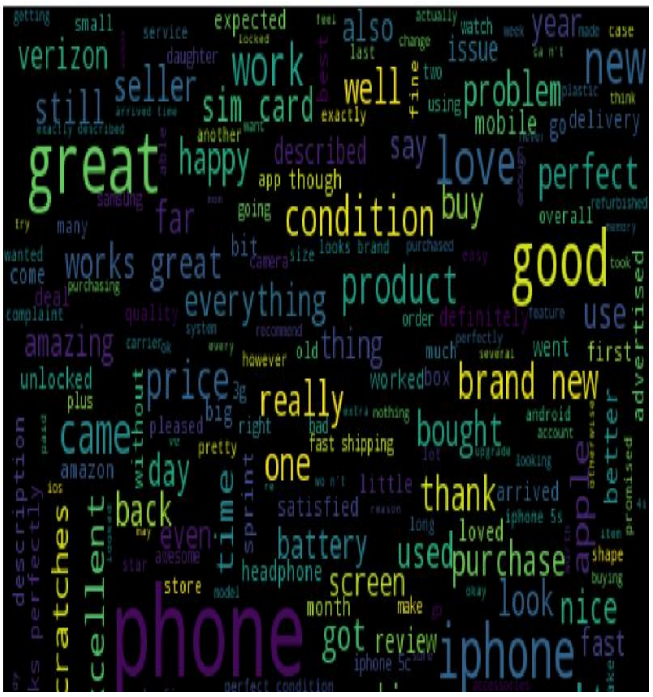

Figure 2: Positive word cloud

Figure 3: Negative word cloud

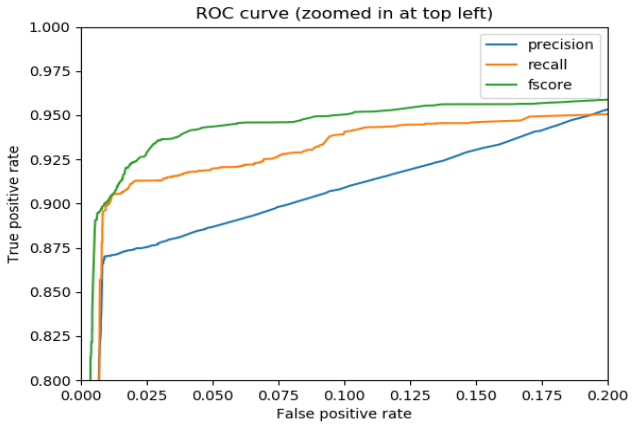

Figure 4: ROC Curve

Results comparison table

Strategy	Existing	Proposed	% change
Accuracy	65.4%	85.7%	+20.3%

VI. CONCLUSION

As the use of the internet is increasing day-by-day, the need for sentiment analysis is also increasing. In the present age where the world has become so fast paced that users have no time and they prefer to use online services. But, even in online services they need to go through the tons of reviews to finally decide on the product. By using this proposed algorithm, the Naive Bayes algorithm, users can make a decision more confidently as the reviews are precisely classified. The results show that this method performs more accurately than the existing models with 20.3% increased effectiveness.

REFERENCES

1. Tahura Shaikh.et.al "Feature Selection Methods in Sentiment Analysis and Sentiment Classification of Amazon Product Reviews" IJCTT – Volume 36 Number 4 - June 2016
2. Tanjim Ul Haque et.al. "Sentiment analysis on large scale Amazon product reviews" IEEE, 11 June 2018
3. E.-H. Han, G. Karypis, and V. Kumar, "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification," 2001.
4. F. E. T. Al, "Locally Weighted Naive Bayes," 2003.
5. J. M. Ponte and W. B. Croft, "A Language Modeling Approach to Information Retrieval," pp. 275–281.
6. D. E. Losada and L. Azzopardi, "Assessing Multi-variate Bernoulli models for Information Retrieval," no. February 2014.
7. L. Jiang, Z. Cai, D. Wang, and H. Zhang, "Knowledge-Based Systems Improving Tree augmented Naive Bayes for class probability estimation," Knowledge-Based Syst., vol. 26, pp. 239–245, 2012.
8. B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based framework for text categorization," Procedia Eng., vol. 69, pp. 1356–1364, 2014.
9. Y. Wang and Z. O. Wang, "A fast KNN algorithm for text categorization," Proc. Sixth Int. Conf. Mach. Learn. Cybern. ICMLC 2007, vol. 6, no. August, pp. 3436–3441, 2007.
10. K. Masuda and T. Matsuzaki, "Semantic Search based on the Online Integration of NLP Techniques," vol. 27, no. Pacling, pp. 281–290, 2011.
11. G. Toker and Ö. Kırmemiş, "Text Categorization Using K Nearest Neighbor Classification," Middle East Tech. Univ., 2013.
12. G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Using kNN Model-based Approach for Automatic Text Categorization," Soft Comput., vol. 10, no. 5, pp. 423–430, 2006.
13. T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization."
14. S. Manne, "A Query based Text Categorization using KNearest Neighbor Approach," vol. 32, no. 7, pp. 16–21, 2011.
15. A. Arnold, "Query Dependent Ranking Using K-Nearest Neighbor ∗," 2008.
16. H. Ug, "Knowledge-Based Systems A two-stage feature selection method for text categorization by using information gain , principal component analysis and genetic algorithm," vol. 24, pp. 1024–1032, 2011
17. S. V. Thirupathur Kailashnathan Vijay Vangara, "A Survey onNatural Language Processing in context with Machine Learning," IJAEMA, vol. XII, no. 1, pp. 1390–1395, 2020, doi: 18. 0002.IJAEMA. 2020.V12I1.200001.015103.

AUTHORS PROFILE

Raghavendra Vijay Vangara is a doctoral student in computer science department at the University of Missouri -St. Louis. He received bachelor's degree in Computer Science from Osmania University in Hyderabad, India and master's degree in Computer Science from University of Central Missouri in Missouri, USA. He is currently working for Missouri Institute of Mental Health as Research Assistant. His current research areas are Natural Language Processing, Machine Learning and Deep learning.

Kailashnathan Thirupathur is a Computer Engineer whose research is mainly in Natural Language Processing. He received bachelor's degree in Computer Science from James Cook University in Singapore and master's degree in Computer Science from University of Bridgeport in Connecticut, USA. His current research areas are Natural Language Processing, Machine Learning and Deep learning.

Shiva Prasad Vangara is a Computer Engineer in whose research is mainly in Natural Language Processing. He received bachelor's degree in Computer Science from Madras University in Chennai, India and master's degree in Computer Science from Indiana Tech University, Indianapolis, USA. His current research areas are Natural Language Processing, Machine Learning and Deep learning.