Naive Bayes model with improved negation handling and n-gram method for Sentiment classification

Sumit Kumar Garg
Ronak Kumar Meher

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Naive Bayes model with improved negation handling and n-gram method for Sentiment classification

Thesis submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Sumit Kumar Garg (111CS0068)

Ronak Kumar Meher (111CS0138)

under the guidance of

Prof. Korra Sathya Babu

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.
May 2015

DECLARATION

This thesis is a presentation of our original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgment of collaborative research and discussions. We hereby declare that this thesis is our own work and effort and that it has not been submitted anywhere for any award. The interpretations put forth are based on our reading and understanding of the original texts and they are not published anywhere in the form of books, monographs or articles. The other books, articles and websites, which we have made use of are acknowledged at the respective place in the text. For the present thesis, which we are submitting to NIT Rourkela, no degree or diploma or distinction has been conferred on us before, either in this or in any other University. We bear all responsibility and prosecution for any of the unfair means adopted by us in submitting this thesis.

Date **Signature**

Acknowledgment

We owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, We would also like to express our gratitude towards our project advisor, Prof. Korra Sathya Babu, whose mentor-ship has been paramount, not only in carrying out the research for this thesis, but also in developing long-term goals for our career. His guidance has been unique and delightful. He provided his able guidance whenever We needed it. Yet he always inspired us to be an independent thinker, and to choose and work with independence.

We would also like to extend special thanks to our project review panel for their time and attention to detail. The constructive feedback received has been keenly instrumental in improvising our work further.

We would like to thank other researchers in our lab and our friends for their encouragement and understanding.

Sumit Kumar Garg

Ronak Kumar Meher

Abstract

Sentiment classification is turning into one of the most fundamental research areas for prediction and classification. In Sentiment mining, we basically try to analyse the results and predict outcomes that are based on customer feedback or opinions. Some work has been done to increase the accuracy of the Naive Bayes classifier. In this project we have examined different methods of improvising the accuracy and space of a Naive Bayes classifier for sentiment classification. We have used a modified negation handling method using POS tagging to decrease the number of feature in the feature set and also discovered that combining these with n-gram method results in improvement in the accuracy. So, a more accurate sentiment classifier with less space complexity can be built from Naive Bayes Classifier.

Keywords: Sentiment analysis, nave Bayes classifier, n-gram, negation handling, POS tagging

Contents

Certificate ii

Acknowledgement iii

Abstract iv

List of Figures vii

1 Introduction 1

1.1 Sentiment analysis 2

1.2 Opinion Mining 2

2 Literature Survey 3

2.1 Opinion Mining Terminology 3

2.2 Steps in Opinion Mining 4

2.2.1 Document-Level Opinion Mining 5

2.2.2 Sentence-Level Opinion Mining 6

2.2.3 Feature-Level Opinion Mining 6

2.3 Previous works 6

3 Proposed Work 9

3.1 Extraction of Dataset 9

3.2 Negation Handling 9

3.3 Application of bigrams, tri-grams and four-grams 11

3.4 Find the semantic orientation of the document 12

3.4.1 Algorithms for Training and testing dataset 13

3.5 Use of Confusion Matrix 13

4 Implementation and Results 15

5 Conclusion **18**

Bibliography **19**

List of Figures

3.1 Confusion Matrix 14

4.1 Evolution on accuracy on different methods 16

Chapter 1

Introduction

With the quick development of e-commerce, more items are sold on the Web, and more individuals are likewise purchasing items on the web. To upgrade consumer loyalty and shopping background, it has turned into a typical practice for online shippers to empower their clients to to express opinions on the items that they have acquired. With more and more regular clients getting to be agreeable with the Web, an expanding number of individuals are composing surveys. Subsequently, the number of surveys that an item gets grows quickly. Besides, numerous surveys are long and have just a couple of sentences containing conclusions on the item. This makes it difficult for a client to peruse them to make an educated choice on whether to buy the item. On the off chance that he/she just peruses a couple of audits, he/she may get an one-sided perspective. The expansive number of audits additionally makes it hard for item makers to stay informed concerning client assessments of their items. For an item maker, there are extra challenges in light of the fact that numerous dealer destinations may offer its items, and the maker may produce numerous sorts of items.

In this segment, we give a brief presentation about what is opinion mining, what is sentiment analysis and how it can be performed. We additionally give brief thought regarding the terminology utilized as a part of this paper that are needed for a superior comprehension the paper Opinion Mining or Sentiment classification includes building a framework to make utilization of surveys posted by the clients and conclusions that are communicated in websites and gatherings as remarks and surveys in e-commerce sites.

1.1 Sentiment analysis

What do different people think has constantly been a key segment in decision making philosophy. Assumption Analysis or Sentiment Classification is the technique to naturally focus the opinions conveyed in a touch of plain substance using some standard computerized get ready frameworks. To be particular, term Sentiment is astoundingly wide and it constitutes emotions, conclusions, manners, specific experiences etc. In this hypothesis, we talk pretty much the assessments conveyed in compositions which are written in human readable natural language, in e-commerce sites.

1.2 Opinion Mining

Opinion mining is the part of study that analyses individual sentiments, opinions, assessments, mentality, feelings, attitude, and emotions from written text. It has pulled in various examiners from unmistakable territories of investigation including data mining, Natural Language Processing, machine learning and also sociology. In the present section, we first analyse necessity for opinion mining and thereafter portray the phrasings used inside this study. In the advancing segments, we discuss general sentiment mining assignments and present a compact review of the present and related manages every one.

Chapter 2

Literature Survey

In this section, we give a brief introduction about the previous work done in the area of Sentiment analysis and opinion mining. We also give brief idea about the basic concepts and terminology used in this paper for a better understanding of the paper. What are the type of sentiment analysis, what are the methods that have been performed in classification of sentiments, what was their results will be discussed in this part.

2.1 Opinion Mining Terminology

In this segment we define the various terms used in the Sentiment Analysis.

Fact: A fact is that which has truly happened or which is really the case.

Opinion: An opinion is a view or judgment formed about something (like product or movie) not necessarily based on fact or knowledge.

Subjective Sentence: A sentence or a text is a subjective or opinionated if it actually indicate ones feelings or emotions.

Objective Sentence: An objective sentence indicates some facts and known information about the world. for example: universal truths.

Review: A review is texts that contain a particular combination of words that has opinions of customer a particular item or opinions of viewers for a movie. A review may be subjective or objective or even both.

Item: An individual article or unit, especially one that is part of a list, collection or set.

Known Aspects: Known aspects are default aspects provided by the certain website for which users separately give ratings.

Sentiment: Sentiment is a polarity term that implies to the direction in which a behavior or opinion is expressed. For example, excellent is a sentiment for the attribute camera in the sentence ”the camera of the iphone is excellent”.

Opinion Polarity: Opinion Polarity or Subjectivity Orientation denotes the polarity expressed by the user or customer or viewer in terms of numerical values.

Rating: Most of the people use star ratings for expressing polarity, represented by stars in the range from 5 to 1 which are called ratings.

Polarity: Polarity is a three way orientation scale. In this, a sentiment can be either negative or positive or neutral.

2.2 Steps in Opinion Mining

In this fragment we demonstrate a study of the present and related tasks in brief at opinion mining proposed in the current techniques. For an thorough study, we sort opinion mining system into three general classes, yet before we discuss our plan, we present current request developments experienced in the written work. Pang et al. [1] bunch the critical issues of opinion mining into three classes: sentiment polarity identification, joint topic-sentiment analysis, and subjectivity detection. Some experts also describes three mining systems for opinionated content in their books. He further develops this classification in his handbook as: sentiment and subjectivity characterization, aspect based opinion mining, opinion search and data retrieval, sentiment analysis of similar sentences, and opinion spam identification. Finally, in his most recent book he portrays three general classifications for sentiment mining tasks: document level, phrase level, and sentence level.

1. Document level opinion mining
2. Sentence level opinion mining
3. feature level opinion mining

2.2.1 Document-Level Opinion Mining

Document level tasks essentially concerns with grouping issues where the accessible report must be organized into an arrangement of predefined classes. In subjectivity examination an document is defined as subjective or objective. In sentiment analysis, a record can be positive or negative or unbiased (or neutral) relying on the polarity of subjective data that is exhibit in the report. Opinion quality and support evaluation settles on choice whether a sentiment is helpful or not and opinion spam identification groups and divides opinion as not a spam and a spam.

Subjectivity Analysis

Subjectivity Analysis refers to finding whether the given document makes an opinion or not. To be precise, whether a document or text is objective or subjective. We take this problem generally as a classification problem. Many of the current methods uses supervised learning, even though we have few unsupervised methods. One of the works in this area given by Wiebe et al [2]. does subjectivity analysis using the naive Bayesian classifier. Succeeding research uses other learning algorithms for finding subjective text. Future research has been mainly focused on developing automated process for subjectivity analysis. One of the tough tasks in subjectivity classification is the human effort involved in labeling training examples as subjective or objective.

2.2.2 Sentence-Level Opinion Mining

The issue at sentence level opinion mining is, it measures everything in reference to sentences. In information extraction and recovery and sentiment inquiry answering, sentences are for generally set and situated concentrated around some criteria. Sentiment plans to choose an arrangement of sentences which describes the feeling more precisely. Finally, sentiment mining in relative sentences fuses perceiving comparative sentences and concentrating information from them.

2.2.3 Feature-Level Opinion Mining

Feature level opinion mining comes into picture when a client or user searching for criticism of certain feature or quality or attribute of a product rather than total feedback of the complete product. We see numerous customers interested in only certain features of specific products rather than the whole product like a few people look for a mobile that has excellent battery life and they are not concerned with other features like camera resolution, music sound and so on. In circumstances like mentioned in this part, feature level opinion mining helps a considerable measure for extracting polarity information for a particular feature or attribute from a product.

2.3 Previous works

In the field of Sentiment analysis some work has been done by Hu and Lius [3]. Using association mining they looked for the features that have been talked about by the people frequently. Their proposed method was effective in discovering frequent features. They used Naive Bayes classifier to classify the extracted feature. And, some works on negation handling and n gram with Naive Bayes classifier has also been done by Vivek and Ishan [6]. But, Our work is slightly different from them as we have used bigram, trigram and four gram. We have also used confusion matrix, recall and precision along with accuracy to compare the outputs of the different methods.

Naive Bayes Model

A Naive Bayes classifier is a general probabilistic model which is based on the Bayes rule in addition of a assumption of independence. The Bayes Rule is given by :

$$p(c|d) = \frac{p(c \cap d)}{p(d)} \tag{2.1}$$

The Nave Bayes model includes a simplifying conditional independence assumption ie the position of different features are independent on their position. The accuracy of the review is not affected the assumption of independence. It also makes the model considerably fast for classification. Rennie et al [4] [5] discuss the performance of Nave Bayes on text classification tasks.

Negation Handling Model

Negation handling is one of the methods that usually increase the accuracy of the classifier. Since unigrams are used as features, the word bad in the phrase not so bad will be reflecting to negative sentiment though it is positive. Here if the word is present before any word say "bad", then it is not considered.

Previously some work in negation handling has been done by Vivek and Ishan [6]. In their work, they used the negation handling method described by Chen and Das [7], and considered the effect of negators till the end of the sentence or till another negator is encountered, which increases the number of unnecessary features.

Tag	Description	Tag	Description
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle VBN Verb, past participle
NN	Noun, singular or mass	VBP	Verb, non-3rd person singular present
NNS	Noun, plural	VBZ	Verb, 3rd person singular present
NNP	Proper noun, singular	WDT	Wh-determiner
NNPS	Proper noun, plural	WP	Wh-pronoun
PDT	Predeterminer	WP$	Possessive wh-pronoun
POS	Possessive ending	WRB	Wh-adverb
PRP	Personal pronoun	VBN	Verb, past participle

Table 2.1: Part of Speech codes

POS Tagging

The process of classifying words into their parts of speech and labeling them accordingly is known as part-of-speech tagging. The POS tags and the description are given in the table 2.1

Chapter 3

Proposed Work

In this section, We have discussed about proposed approach for performing sentiment classification and what techniques and algorithms we have used to determine the class of documents in test dataset for getting useful information from product reviews. Our methodology includes the following steps:

1. Extraction of Dataset (Both Training set and Testing set)
2. Preprocessing of dataset
3. Negation handling
4. Application of bigrams, tri-grams and four-grams
5. Find the semantic orientation of the document
6. Summarization

3.1 Extraction of Dataset

We have used a highly polar dataset of product reviews from the e-commerce site amazon.co.in (12,500 positive reviews and 12,500 negative reviews) for training and 20,000 for testing.

3.2 Negation Handling

Negation handling is one of the methods that usually increase the accuracy of the classifier. Since unigrams are used as features, the word bad in the phrase not so

bad will be reflecting to negative sentiment though it is positive. Here if the word is present before any word say "bad", then it is not considered. So to figure out the result we used a basic algorithm to handle the negated words.
Earlier, some work has been done by Vivek and Ishan [6], regarding negation handling. They have used the algorithm suggested by Chen and Das [7]. In that algorithm, whenever a negation word like "not" or "nt" or "no" was found, and the flag is true, the words following the negation word are considered as "not_" + word.

In their process, they continued the algorithm till the end of the sentence or till we encounter another negation word, which increase the unnecessary features in the feature set of opposite class. But, in this paper we have done a little modification to that. We continue the process till a particular word or phrase is encountered instead of going till the end. We have used POS tagging to find the word or words that are effected by the negator. So the proposed algorithm 1 is described below :

Algorithm 1: Negation Handler

```
Require: doc
    doc: given document
flag = false
for w in doc do
    if w = "not" or "n't" or "no" then
        flag= not flag
        continue
    end if
    if flag= True then
        if w= Adjective or Verb or Noun then
            Add "not_"+ w to feature set of opposite class
        end if
        if (w= Adverb or Determiner) and nextw= Adjective then
            Add "not_"+ nextw to feature set of opposite class
        end if
        if w= Determiner and nextw= Adverb and nextnextw = Adjective then
            Add "not_"+ nextnextw to feature set of opposite class
        end if
        flag= false
    end if
end for
```

We have used POS tagging in the above algorithm. POS tagging or the part of speech tagging refers to the classification of each word into its category. In the algorithm, adjective refers to JJ, JJR, JJS; adverb refers ro RB, RBR, RBS; verb refers to VB, VBD, VBG, VBN, VBP, VBZ; noun refers to NN, NNS, NNP; determiner refers to DT.

In our algorithm, the process stops after processing some words instead of continuing till the end. For instance, in the review "this is not a good phone and still i bought it", after finding the negator "not" our algorithm stops when it encounters the word "good", adding "not_good" to feature set of opposite class unlike the work of Vivek and Ishan [6] of going till end. Our algorithm prevents the addition of unnecessary phrases like "not_and", "not_still", "not_bought" in the opposite classifier and improves the space complexity.

3.3 Application of bigrams, tri-grams and four-grams

Generally, from the adjectives or from some combinations of adjectives with adjectives and other parts of a document, the information about sentiment is fetched [6] [4]. This data can be found by adding features like adjacent words (bigrams), or word triplets (trigrams) or even four consecutive words (four grams). For example the words like "so" or "too" don't give lot of information about sentiment on their own, but phrases like "so cute" or "too good" enhance the possibility of that document being positively or negatively classified.

In four-gram model, all the four-gram feature extracted by general method may not contribute towards the efficient classifier. So, to decrease the noise caused by four-gram method we have chosen some particular sequence of words instead of all four-grams. For example, sequence of "DT + RB + RB + JJ" (a very very good phone) or sequence of "CD + IN + DT + JJS" (one of the best phone) are considered.

Here we have taken the unigrams, bigrams, trigrams and four grams. By using n gram, we got 13,67,528 features from the 25,000 training data set. All the words or features of the feature set are not applicable for sentiment classification. So, we have to choose a particular number of useful feature from the feature set. We have taken top 50,000 features based on their frequency for further work.

3.4 Find the semantic orientation of the document

After applying the above methods, we have to find the semantic orientation of the testing data sets. So to calculate the positive and negative score we have used Naive Bayes classifier.

Naive Bayes Classifier

A Naive Bayes classifier is a general probabilistic model which is based on the Bayes rule in addition of a assumption of independence. The Bayes Rule is given by :

$$p(c|d) = \frac{p(c \cap d)}{p(d)} \tag{3.1}$$

The Nave Bayes model includes assumption that the effect of the features are independent on their position. Here, the probability of maximum likelihood of a word (feature) belonging to a particular class is calculated by the equation:

$$p(w|c) = \frac{count(w,c)}{totalcount(c)} \tag{3.2}$$

where *count(w,c)* is count of word w in documents with class c and *totalcount(c)* is total number of words in documents with class c.
Bayes Rule states, the probability of a particular document belonging to a class c_i is given by:

$$p(c_i|d) = \frac{p(d|c_i) * p(c_i)}{p(d)} \tag{3.3}$$

If we use the assumption of independence the above equation can be written as:

$$p(c_i|d) = \frac{\prod p(x_i|c_j) * p(c_j)}{p(d)} \tag{3.4}$$

Here the x_i s are the distinct words of the document. p(x_i—c_j) is the likelihood[x_i][c_j] and p(c_j) is the prior[c_j].

In Navie Bayes model if the classifier meets a feature for the first time in the training set, the probability of the feature in both the classes will be zero. So by using Laplacian smoothing this kind of problem can be solved.

$$p(w|c) = \frac{Count(w,c) + k}{(k+1) * Noofwordsinclassc} \tag{3.5}$$

Generally, k is selected as 1. So in this way, there is a same probability for the new word to be present in any of the classes.

3.4.1 Algorithms for Training and testing dataset

The algorithms 2 and 3 have been used for the training dataset and testing dataset respectively.

Algorithm 2: Training Algorithm

Require: D,C
D: Set of Documents (Training dataset)
C: class of the document positive, negative
V = Extract feature Vector(D)
N = Number of training documents
for c in C **do**
  N_c = Number of documents with class c
  prior[c] = N_c / N
  for w in V **do**
    likelihood[w][c] = (count(w,c)+k)/((k+1)*Number of words in class c)
  end for
end for
return prior, likelihood

.

3.5 Use of Confusion Matrix

In machine learning, a confusion matrix is a particular table format, that permits visualization of execution of a supervised learning algorithm. It holds information about predicted and actual classification done by the supervised classifier. The

---

Algorithm 3: Testing Algorithm

```
Require: d
    d: Document to test
W = Extract Feature Vector(d)
for c in C do
    score[c] = prior[c]
    for w in V do
        score[c]= score[c] * likelihood[w][c]
    end for
end for
return argmax(score[c])
```

---

following figure 3.1 shows elements of the matrix.

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Figure 3.1: Confusion Matrix

The formulas for precision, recall and accuracy are given below:

$$precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.6}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.7}$$

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{3.8}$$

Chapter 4

Implementation and Results

We have applied our classifier by the help of dictionary data structure of python to keep track of the frequency of the features. In training phase we performed the negation handling method and then applied the bigram, tri-gram and four-gram method to train our classifier.

We have used a highly polar, publicly accessible dataset of product reviews from the e-commerce site amazon.co.in. It is a database of 25,000 highly polar product reviews (12,500 positive reviews and 12,500 negative reviews) for training and 20,000 for testing.

Results

After performing all the above methods and algorithms to the testing data set, we got the results as shown in table 4.1. The histogram in the figure 4.1 shows the result of addition of different methods to the Naive Bayes model.

Comparison

The Original Naive Bayes model gave an accuracy of 74.12%, but after performing the above methods 87.93% of the testing data set were found to be classified accurately with decrease in space complexity. We got these results by applying a modified negation handling method and four-gram method. We have applied a modified negation handling algorithm, in which we consider the effect of negation word up to a

Different Methods	True positive	True Negative	False Positive	False Negative	Accuracy
NB Classifier	7448	7376	2624	2552	74.12
NB with negation handling	8237	8326	1672	1763	82.94
NB with negation handling and bigrams, trigrams	8789	8738	1262	1211	87.63
NB with negation handling and bigrams, trigrams and four grams	8845	8742	1258	1155	87.93

Table 4.1: Output of different Methods

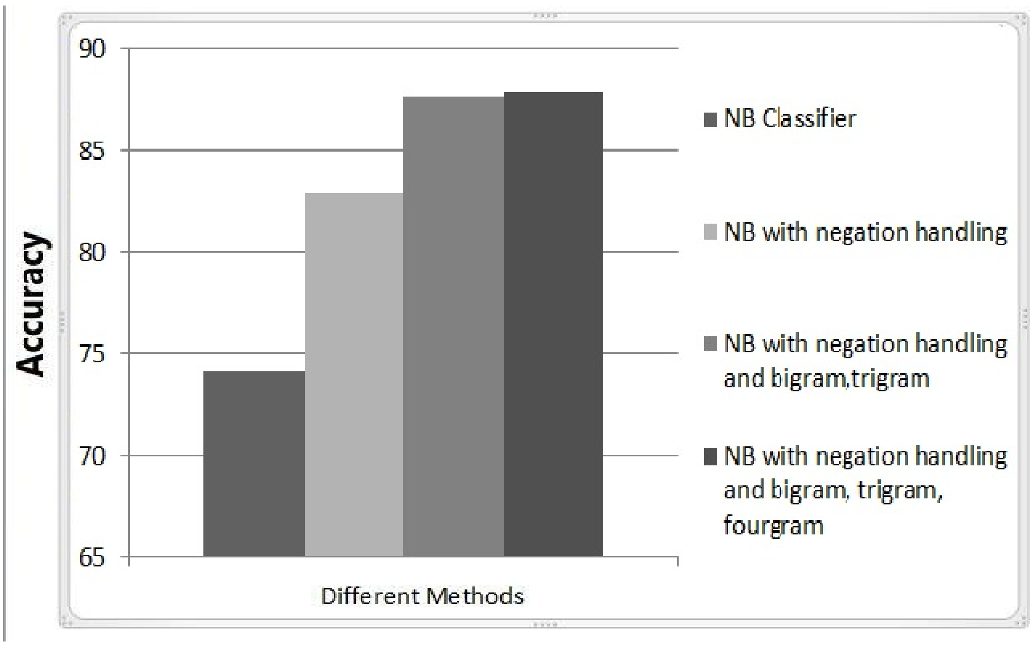

Figure 4.1: Evolution on accuracy on different methods

particular word in the sentence instead of effecting till the end of the sentence. This decreases the number of unnecessary features in the feature set and thus decreases the space complexity of the classifier as compared to the methods used by [6] [7]. The over fitting problem refers to the presence of ambiguity or noise in the classifier. It generally occurs when the model is quite complex. Like in four-gram model, all the four-gram feature extracted by general method may not contribute towards the efficient classifier. So, to overcome the over fitting problem caused by four-gram method we have chosen some particular sequence of words instead of all four-grams. For example, sequence of "DT + RB + RB + JJ" (a very very good phone) or sequence of "CD + IN + DT + JJS" (one of the best phone) are considered. By choosing some particular sequence we are making the classifier more space optimized.

Chapter 5

Conclusion

In this project our results showed that the accuracy and space complexity of the Naive Bayes classifier can be improved by adding different methods like modified negation handling and n-gram methods. Here we got an accuracy of 87.93% by using all the methods, whereas in the original Naive Bayes classifier with only unigrams as its feature set, we got only 74.12% of accuracy over the same set of test data set. The space complexity is also optimized by using the improved negation handling method and by using four-gram method using POS tagging. The different methods used in this project can easily be implemented on the Naive Bayes model unlike other models like SVM or maximum entropy model to optimize the time and space complexity over a large data set.

Bibliography

[1] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[2] Janyce M Wiebe, Rebecca F Bruce, and Thomas P O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics, 1999.

[3] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[4] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[5] Fuchun Peng and Dale Schuurmans. *Combining naive Bayes and n-gram language models for text classification*. Springer, 2003.

[6] Vivek Narayanan, Ishan Arora, and Arjun Bhatia. Fast and accurate sentiment classification using an enhanced naive bayes model. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013*, pages 194–201. Springer, 2013.

[7] Sanjiv Ranjan Das and Mike Y Chen. Yahoo! for amazon: Sentiment parsing from small talk on the web. In *EFA 2001 Barcelona Meetings*, 2001.

[8] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment classification using word sub-sequences and dependency sub-trees. In *Advances in Knowledge Discovery and Data Mining*, pages 301–311. Springer, 2005.