Classification of Sentimental Reviews using Natural Language Processing Concepts and Machine Learning Techniques

Ankit Agrawal

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

Classification of Sentimental Reviews using Natural Language Processing Concepts and Machine Learning Techniques

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Software Engineering)

by

Ankit Agrawal

(Roll Number - 213CS3178)

Under the supervision of

Prof. S. K. Rath

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

May 2015

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

Certificate

This is to certify that the work done in the thesis entitled ***Classification of Sentimental Reviews Using Natural Language Processing Concepts and Machine Learning Techniques*** by ***Ankit Agrawal*** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Software Engineering in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: 23 May 2015

Prof. Santanu Ku. Rath
HOD, CSE Department
NIT Rourkela, Odisha

Acknowledgment

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Prof. Santanu Ku. Rath for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction to the research and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I am also thankful to all the Professors at the department for their support.

I would like to thank Mr. Abinash Tripathy for his encouragement and support. His help can never be penned with words.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their help can never be penned with words.

I must acknowledge the academic resources that I have got from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

Ankit Agrawal

Roll-213CS3178

Abstract

Natural language processing (NLP) is the hypothetically motivated scope of computational strategies for representing and analyzing naturally occurring text at many levels of textual analysis for the goal of attaining automatic language processing system for multiple tasks and applications. One of the most import application of natural language processing from industry perspective is sentiment analysis.

Sentiment analysis is the most eminent branch of NLP because of its capability to classify any textual document to either as positive or negative polarity. With the proliferation of world wide web, huge textual unstructured data in form of tweets, messages, articles, social networking discussions, reviews of products and movies are available so as to extract right information from the large pool. Thus, a need is felt to analyze this data to bring out some hidden facts based on the intention of the author of the text. The intention can be either criticism(negative) of product and movie review or it can be admiration(positive). Although, The intention can vary from strongly positive to positive and strongly negative to negative.

This thesis completely focuses on classification of movie reviews in either as positive or negative review using machine learning techniques like Support Vector Machine(SVM), K-Nearest Neighbor(KNN) and Naive Bayes (NB) classifier. Further, a N-gram Model has been proposed where the documents are classified based on unigram, bigram and trigram composition of words in a sentence. Two dataset are considered for this study; one is a labeled polarity dataset where each movie review is either labeled as positive or negative and other one is IMDb movie reviews dataset. Finally, the prediction accuracy of above mentioned machine learning algorithms in different manipulations of same dataset is studied and a comparative analysis has been made for critical examination.

Contents

Certificate **ii**

Acknowledgement **iii**

Abstract **iv**

List of Figures **vii**

List of Tables **viii**

Abbreviations **x**

1 Introduction **1**

1.1 Branches of Natural Language Processing 2

1.2 Dataset used for classification of documents 5

1.3 Problem Definition 5

1.4 Motivation 6

1.5 Thesis Organization 6

2 Literature Survey **7**

2.1 Phase-Wise Evolution of Natural Language Processing Concept . . 7

2.2 Sentiment Analysis: Tools and Techniques 8

3 Classification of Sentimental Reviews **11**

3.1 Introduction 11

3.2 Methodology 12

3.3 Proposed Approach 17

3.3.1 Steps Followed For Classification: 18

3.4 Implementation and Results 20

3.4.1 Results of Naive Bayes Algorithm 20

3.4.2 Results of Support Vector Machine Algorithm 21

3.4.3 Results of K-nearest Neighbor Algorithm 23

3.5 Comparison of Results Obtained 24

4 N-Gram Model for Classification 26

4.1 Introduction 26

4.2 Results and Analysis 27

4.2.1 Naive Bayes Classifier 27

4.2.2 KNN Classifier 29

4.2.3 Support Vector Machine Classifier 31

4.2.4 Comparative Analysis 33

5 Conclusions and Future Work 34

Bibliography 35

Dissemination of Work 39

List of Figures

1.1 Sentiment analysis pictorial representation 2

1.2 Data flow diagram for Text categorization 4

3.1 Distance Functions used in KNN 15

3.2 *Diagrammatic view of the proposed approach* 17

4.1 Graphical Presentation of Confusion Matrix for Naive Bayes Classifier 28

4.2 Precision Recall Curve of Naive Bayes Classifier 28

4.3 Graphical Presentation of Confusion Matrix for KNN Classifier . . . 30

4.4 Precision Recall Curve of KNN Classifier 30

4.5 Graphical Presentation of Confusion Matrix for SVM Classifier . . . 32

4.6 Precision Recall Curve of SVM Classifier 32

List of Tables

2.1 Comparison of Sentiment techniques literature 10

3.1 Confusion Matrix 16

3.2 Confusion matrix for Naive Bayes classifier on Polarity dataset . . . 20

3.3 Evaluation parameters for Naive Bayes classifier on Polarity dataset 20

3.4 Confusion matrix for Naive Bayes classifier on IMDb Dataset 21

3.5 Evaluation parameter for Naive Bayes classifier on IMDb Dataset . 21

3.6 Confusion matrix for Support Vector Machine classifier on polarity dataset 22

3.7 Evaluation parameters for Support Vector Machine classifier on Polarity dataset 22

3.8 Confusion matrix for Support Vector Machine classifier on IMDb Dataset 22

3.9 Evaluation Parameter for Support Vector Machine on IMDb Dataset 23

3.10 Confusion matrix for K-Nearest Neighbor classifier on IMDb dataset 23

3.11 Evaluation parameter for K-Nearest Neighbor classifier on IMDb dataset 23

3.12 Confusion matrix for K-Nearest Neighbor classifier on IMDb dataset 24

3.13 Evaluation parameter for K-Nearest Neighbor classifier on IMDb dataset 24

3.14 Comparison of different existing literatures with proposed approach on Polarity dataset 24

3.15 Comparison of different existing literatures with proposed approach on IMDb Dataset 25

4.1 Confusion Matrix, Evaluation Parameter and Accuracy for Naive Bayes n-gram classifier 27

4.2 Confusion Matrix, Evaluation Parameter and Accuracy for k-nearest neighbor n-gram classifier 29

4.3 Confusion Matrix, Evaluation Parameter and Accuracy for Support Vector Machine n-gram classifier 31

4.4 Comparative result obtained with different literature using IMDb dataset and ngram approach 33

List Of Abbreviation

NLP	Natural Language Processing
POS	Parts-of-Speech
IMDb	Internet Movie Database
TF-IDF	Term Frequency-Inverse Document Frequency
SVM	Support Vector Machine
NB	Naive Bayes
KNN	K-Nearest Neighbor
CM	Confusion Matrix
SO	Semantic Orientation
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Chapter 1

Introduction

Natural language processing (NLP) is the investigation of scientific and computational modelling of different parts of language and the advancement of an extensive variety of frameworks. These contain speech recognition frameworks that amalgamate natural language and speech; agreeable interfaces to databases and learning bases that model parts of human-human association; multilingual interfaces; machine interpretation; and message-understanding frameworks. Research in NLP is exceptionally interdisciplinary, including ideas in software engineering, etymology, rationale, and psychology. NLP has an extraordinary part in software engineering on the grounds that numerous parts of the field manage semantic features of reckoning and NLP looks to model language computationally [1].

At the center of any NLP assignment, there is the critical issue of natural language understanding. The methodology of building computer programs that comprehend natural language includes three noteworthy issues: the first identifies with the point of view, the second addresses to the representation and importance of the phonetic info, and lastly the third and final one addresses to the world information. Therefore, a NLP framework may start at the word level to focus the morphological structure, nature, (for example part-of-speech). Next, It proceeds onward to the sentence level to focus the word request, language structure, importance of the whole sentence, etc. and afterward to the setting and the general environment or area. A given word or a sentence may have a particular importance or essence in a given connection or space, and may be identified with numerous different sentences and/or words in the given context.

NLP is a vast research area and hence consist of different sub areas where stress is given on a particular issue. Some of the research area under NLP are listed below.

- Sentiment analysis and opinion mining
- Text Summarization
- Text Categorization
- Parts-Of-Speech(POS) tagging

1.1 Branches of Natural Language Processing

- **Sentiment Analysis**: Sentiment mainly refers to emotions, feelings, attitude or opinion. With the proliferation of World Wide Web, Individuals tends to do everything on-line which include discussions on social media like twitter and Facebook, expressing views by writing blogs and ratings and reviews of movie or any item. The textual data over internet has grown to more than 20 billion pages. This huge text contains lot of information about a particular topic and needs to be examined to understand the response of individuals towards any item. Sentiment analysis is used to extract the polarity of any textual data. Polarity can be positive, neutral and negative. Product companies apply the concept of sentiment analysis to understand the standing of product in market and customer satisfaction by analyzing comments and reviews available on-line for their product. The result provides great insights to understand customer needs and can help in taking future marketing decisions.

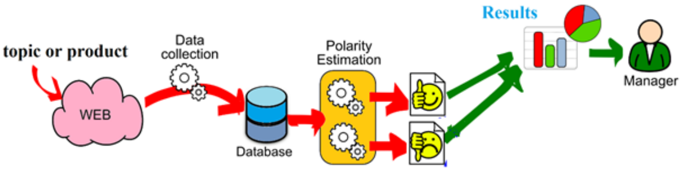

Figure 1.1: Sentiment analysis pictorial representation

The diagram in figure 1.1 shows the complete process involved in sentiment analysis of issues or product. The process starts from searching the issue or product reviews on the web with the help of data collector. All the relevant data is captured and stored at a physical location or large databases for analysis. Stored data is fed to the polarity estimator which is applied in any classification algorithm or machine learning algorithm. These algorithms label each collected document from web in to either positive sentiment or negative sentiment. The results obtained and key performance index are forwarded to the appropriate authority for understanding.

Sentiment analysis is performed under three levels, namely *sentence level, document level and aspect level* [2].

- **Sentence level**: It aims at classifying each sentence at a time and finally aggregate the sentiments found in each sentence to estimate the sentiments present in a particular document.
- **Document level**: Classify the whole document at once, considering the polarity of all the words appeared in the document.
- **Aspect level**: This level of sentiment classification is more advanced than other two because it converges the sentiment classification to a particular aspect or attribute of any product.

- **Text Summarization**: As the measure of textual information accessible electronically develops quickly because of proliferation in World wide web, it becomes more difficult for a client to adapt to all the contents that are important. Text summarization tools are hence becoming crucial in industry to convert large chunk of data in to smaller amount [3].

  According to the definition of **"Summary"** given by Radev [4] ,The three important principles are:

  - One or more document can be summarized in to single document.
  - Important data must be saved in summaries.

- Summary should be less than 50 percent of the original text.

Following are the advantages of Text summarization:

- Summary decreases reading time.
- Summaries help in reducing the searching time of literatures.
- Summaries ameliorate efficiency of indexing.

- **Text Categorization**: Text categorization is simply classifying any document in to a per-defined category. Web contains documents irrespective of class or category. Hence it becomes very difficult to identify the category of document in some cases. So, in order to sort the unstructured set of documents, an automated tool is required which can easily classify any document in to a pre-defined category. Different supervised and unsupervised machine learning techniques are utilized to anticipate the category of any document [5].

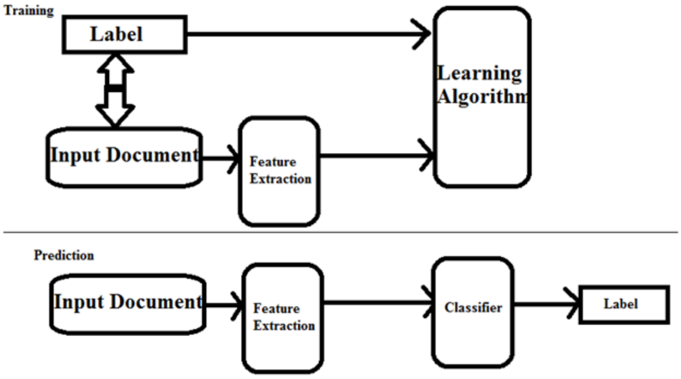

Figure 1.2: Data flow diagram for Text categorization

The figure 1.2 represents the process of classifying a document using supervised machine learning algorithm. Firstly, using labeled documents, the features are extracted and a model is trained for classification. This model is given any input document to classify, it then extracts the features and classify it in a pre-defined category.

- **Parts-of-speech Tagging**: The tagger lives up to expectations via consequently perceiving and helping its shortcomings, accordingly incrementally moving forward its execution. POS tagging is harder than simply having a database of words and their corresponding parts of speech, because many words can act as more than one part of speech at distinctive times. The tagger at first labels by allocating every word by its undoubtedly tag, estimated by inspecting a substantial labeled corpus, irrespective of context. This is not rare in natural languages. Most of the known word-forms are ambiguous. For example

- Sentence 1: Bait as Noun: You will need bait for fishing.
- Sentence 2: Bait as Verb: When you get an earthworm, it can be used to bait your hook.

Same word acting as noun in sentence 1 and verb in sentence 2. A naive tagger can simply ignore the context of the word and can mark the word Bait as either noun or verb in both sentences depending on the corpus. So, automatic detection of such ambiguous nature of parts-of-speech tagging invites further research in natural language processing.

1.2 Dataset used for classification of documents

- **IMDb Dataset** [**6**]: This Dataset consist of 50,000 movie reviews out of which 25,000 reviews are given for training and rest for testing of model. This dataset is a labeled dataset and hence the training set is equally divided in to positive and negative reviews i.e it contains 12500 documents marked as positive and 12500 documents are marked as negative.
- **Polarity Dataset** [**7**]: This dataset is also a labeled dataset and contains 1000 positive reviews and 1000 negative marked sentiment reviews.

1.3 Problem Definition

For classification of any message in to either positive or negative sentiment in a message presenting mixed emotions, the stronger sentiment should be chosen as polarity for that message.

1.4 Motivation

In recent years, there has been an enduring increment in enthusiasm from brands, organizations and analysts in the region of opinion investigation and its application to business examination. The business world today, just like the case in numerous information investigation streams, are searching for business understanding for future development. In relation to sentiment analysis, insights into consumer behavior, what customer's want, what are customer's likes and dislike about any product, what their buying signals are, what their decision process look like etc are important for companies to take business decision easily. Understanding behavior of customers is an important business strategy of any company. This business need is of great motivation to research in sentiment analysis and comes up with high and improved accuracy in prediction using various machine learning techniques.

1.5 Thesis Organization

- **Chapter-2**: This chapter summarizes the existed work done in sentiment analysis along with different dataset, tools and techniques used for analysis in different literatures.

- **Chapter-3**: In this chapter different machine learning techniques like support vector machine, k-nearest neighbor classifier and naive Bayes classifier are implemented and tested across both dataset and results are compared with results available in literature for critical examination.

- **Chapter-4**: This chapter illustrates the proposed N-gram model for classification of sentimental reviews using machine learning techniques. Results are compared with existing results available in literature for critical examination.

Chapter 2

Literature Survey

2.1 Phase-Wise Evolution of Natural Language Processing Concept

- **Phase 1**: This phase begins from late 1940s and continued up to initial years of 1960s and concentrated basically in Machine Translation (MT). During this phase, remarkable measure of work was studied in USSR, Europe ,USA and Japan also. Accordingly the language considered for examination in this period are essentially English and Russian [8]. Sentence structure is essentially the territory of exploration in this period as syntactic handling was clearly vital, and partly through implied or unequivocal support of the thought of linguistic syntax driven processing. In spite of the fact that amid this period utilization of PCs for linguistic and literary study had started, but much computation work not carried out on natural language processing.

- **Phase 2**: This phase begins from late 1960s to end of 1970s and the work for the most part concentrates on utilization of articial intelligence(AI) in natural language processing, with significantly higher emphasis on word learning and on its usage in the development and control of semantic understanding. AI is specifically used during this period for development. In late 1960s, the predominant hypothesis of phonetic is transformational punctuation, which gives the semantic data about NLP.

- **Phase 3**: This phase essentially concerned to the time of late 1970s to late

1980s. This phase can be portrayed as grammaticological phase. The prerequisite of improvement of syntactic hypothesis and development towards joining of logic in representation of logic and thinking activated amid 70s. In the middle of this phase, planned endeavors made to convert the available dictionary to machine understandable structure which further aides in content approval and manipulating lexical data.

- **Phase 4**: The last phase is from late 1980s onward. Amid this stage, the principle territory of examination is based on computational data processing. The recognition of linguistic events and examples in the huge available language resources for both semantic and syntactic investigation is the main attraction in this phase. The present consideration on vocabulary, recovering measurable data, and restore enthusiasm for MT.

2.2 Sentiment Analysis: Tools and Techniques

Pang *et.al.*, have considered sentiment classification taking into account classification perspective with positive and negative sentiments [9]. They have embraced the examination with three different machine learning calculations i.e., Support Vector machine, Maximum Entropy and Naive Bayes classification are being connected over the n-gram techniques.

Turney presents unsupervised calculation to order survey as either prescribed i.e., Thumbs up or Thumbs down [10].The creator has utilized Part of Speech (POS) tagger to distinguish phrases which contain modifiers or intensifiers.

Dave *et. al.* had utilized organized survey for testing and training, recognizing features and score strategies to figure out if the reviews are of positive or negative extremity [11]. They have utilized the idea of classifier to arrange the sentences retrived from web search through search crawlers using name of the product as a search quety in crawler program.

Pang and Lee mark sentences in the report as subjective or goal [12]. They have connected machine learning classifier to the subjective gathering, which avoids polarity classification from considering pointless and misdirecting information. They have investigated extraction of strategies on the premise of minimum cut.

Whitelaw *et. al.* have introduced a sentiment classification technique on the basis of analysis and extraction of appraisal groups [13]. Evaluation group corresponds to an arrangement of attribute values in semantic classification.

Li *et. al.* have proposed different semi-supervised strategies to tackle the issue of deficiency of marked information for sentiment classification [14] . They utilized sampling technique to manage the issue of sentiment classication i.e., imbalance problem.

Wang and Wang have proposed a variance mean based feature filtering method that reduces the feature for representational phrase of text classification [15]. The performance of the method was observed to be quite comparitive as it only considered the best feature and also the computation time got decreased as incoming text was classified automatically.

The table 2.1 provides a comparative study of approaches used by different authors.

Table 2.1: Comparison of Sentiment techniques literature

Authors	Approach	Algorithm Used	Obtained result	Dataset used
Pang et.al [9]	Classify the dataset using machine learning techniques and n-gram model	Naive Baye (NB), Maximum Entropy (ME), Support Vector Machine (SVM)	Unigram: **SVM (82.9)**, Bigram: **ME (77.4)**, Unigram + Bigram : **SVM (82.7)**	Internet Movie Database (IMDb)
Turney [10]	Semantic orientation (SO) of phrases calculated and classification done on average SO	Unsupervised learning algorithm and Point wise Mutual Information and Information Retrieval (PMI-IR)	**Accuracy for dataset Automobiles : 84, Banks : 80, Movie : 65.83, Travel destinations : 70.53**	410 reviews from Epinions
Dave et.al [11]	information retrieval techniques used for feature retrieval and result of various metrics are tested	SVM^{lite}, Machine learning using Rainbow, Naive Bayes	**Naive Bayes : 87.0 (t=2.486)**	Dataset from Cnet and Amazon site
Pang et.al. [12]	Text categorization technique applied to subjective portion of text that obtained using minimum cut in graph technique	Naive Bayes (NB), Support Vector Machine (SVM)	**NB: 86.4 , SVM: 86.15**	5000 sentences from plot summaries available from the Internet Movie Database (www.imdb.com)
Casey Whitelaw et.al. [13]	Semi-automated methods were used to build a lexicon of appraising adjectives and classification done on obtained list	Appraisal Group by Attitude and Orientation and Force, Bag of Words	**Standard bag-of-words (BOW) classification : 87.6, BOW + appraisal group classification: 90.2**	1000 positive and 1000 negative reviews, taken from the IMDb movie review archives.
Shoushan li et.al [14]	Under sampling technique used to handle the imbalance in sentiment classification	Semi-supervised Learning with Dynamic Subspace Generation and Semi-supervised Learning for Imbalanced Sentiment Classification	NA	1,000 positive and 1,000 negative documents on four domains available aaaaat multi-domain sentiment dataset v2.0.
Yi Wang et.al [15]	Variance mean based feature method fro text classification	General SVM classifier SVM^{lite}, Predefined Feature Count (PFC), Mladenic Vector Size (MVS)	**F1 value MVS Strategy: 0.9339, PFC Strategy : 0.866**	Chinese text classification corpus

Chapter 3

Classification of Sentimental Reviews

3.1 Introduction

Sentiment mainly refers to feelings, emotions, opinion or attitude [16]. With the rapid increase of world wide web, people often express their sentiments over internet through social media, blogs, rating and reviews. Due to this increase in the textual data, companies feel the need to analyze this text and calculate the insights for business. Business owners and advertising companies often employ sentiment analysis to discover new business strategies and advertising campaign.

Machine leaning algorithms are used to classify and predict whether a document represents positive or negative sentiment. Supervised and unsupervised are two categories of Machine learning algorithm. Supervised algorithm uses a labeled dataset where each document of training set is labeled with appropriate sentiment. Whereas, unsupervised learning include unlabeled dataset [17]. Unsupervised algorithm are more complex and require additional clustering algorithm in initial phase of implementation. This study mainly concerns with supervised learning techniques on a labeled dataset.

Sentiment analysis can be exercised on three levels namely *document level*, *aspect level* and *sentence level* [18]. Document Level sentiment classification aims at classifying the entire document or topic as positive or negative. In Sentence level classification of sentiments, the polarity of individual sentence of a document whereas aspect level sentiment classification first identifies the different aspects of a corpus and then for each document, the polarity is calculated with respect to all obtained aspects.

In this work, an attempt has been made to transform the textual movie reviews in to numerical matrix where each column represents the identified features and each row represents a particular review. This matrix is given input to machine learning algorithm in order to train the model. This model is then tested and different performance parameters are calculated. Finally a comparative table is shown which compares the results obtained in this study with the results of other researchers.

3.2 Methodology

Two different approaches of sentiment classification are often used i.e. binary classification of sentiments and multi-class classification of sentiments [19]. In binary sentiment classification each document or review of the corpus is classified in to two classes (positive, negative). Whereas, in multi-class sentiment classification, each review can be classified in more than two classes. Strong sentiment can further be classified in to strong positive and weak positive sentiment. Negative sentiment can also be further classified in to strong negative and weak negative sentiment. Just to add more complexity to problem, another category of sentiment is considered called neutral sentiment or no sentiment. Generally, the binary classification is useful when two products need to be compared. In this study, implementation is done with respect to binary sentiment classification.

The repository of movie reviews is stored in unstructured textual format. This unstructured data need to be converted in to meaningful data for machine learning algorithms. The processing of unstructured data includes removal of vague information, removal of unnecessary blank spaces. This processed data is required to be converted in to numerical vectors where each vector corresponds to a review and entries of each vector represent the presence of feature in that particular review.

The vectorization of textual data in to numerical vector is done using following methodologies.

- **CountVectorizer**: Based on the frequency of a feature in the review, a sparse matrix is created [20].
- **Term Frequency - Inverse Document frequency (TF-IDF)**: The TF-IDF score is helpful in balancing the weight between most frequent or general words and less commonly used words. Term frequency calculates the frequency of each token in the review but this frequency is offset by frequency of that token in the whole corpus [20]. TF-IDF value shows the importance of a token to a document in the corpora.

  Calculation of TF-IDF value: Suppose a movie review contains 100 words wherein the word *Awesome* shows up 5 times. The term frequency (i.e., TF) for *Awesome* can be calculated as (5 / 100) = 0.05. Again if we assume that there are 1 million reviews in the corpus and the word *Awesome* shows up 1000 times in whole corpus. The inverse document frequency (i.e., IDF) is computed as log(1,000,000 / 1,000) = 3. Thus, the TF-IDF value is calculated as: 0.05 * 3 = 0.15.

The supervised machine learning algorithm is applicable where the labeled dataset is available. The dataset used in this study is labeled dataset and each review in the corpus is either labeled as positive or negative. The description of different machine learning algorithms implemented in this work is as follows:

1. **Naive Bayes (NB)**: It is a probabilistic classifier that uses the properties of Bayes theorem assuming the strong independence between the features [21]. Despite the naive design of this classifier, it still manages to perform well in many situations. One of the advantage of this classifier is that it demands very little measure of training data to calculate the parameters for prediction. Instead of calculating the complete covariance matrix only variance of the feature is required to be computed because of independence of features.

For a given textual review d and for a class c (positive, negative), the conditional probability for each class given a review is $P(c|d)$. According to Bayes theorem this quantity can be computed by equation 3.1

$$P(c|d) = \frac{P(d|c) * P(c)}{P(d)} \tag{3.1}$$

To further compute the term P(d|c), it is decomposed by assuming that if the d's class is known, then f_i's are conditionally independent. This decomposition of P(d|c) is expressed in equation 3.2

$$P_{NB}(c|d) = \frac{P(c)(\prod_{i=1}^{m} P(f_i)|c)^{n_i(d)})}{P(d)} \tag{3.2}$$

2. **Support Vector Machine (SVM)**: SVM is a non-probabilistic binary linear classifier. In this study, SVM model interprets each review in vectorized form as a data point in the space. This method is used to analyze the complete vectorized data and training of SVM model facilitates to find a hyperplane which is represented by $\vec{w}$ in equation 3.3. The set of textual data vectors are said to be optimally distinguished by hyperplane only when it is separated with least possible error and the distance between closest points of each class and hyperplane is maximum. After training of the model, the testing reviews are projected in to same space and on the grounds of point of projection, the class for a particular review is predicted. [10]

   let $c_j \epsilon \{1,-1\}$ be the class (positive , negative) for a document d_j, the equation for $\vec{w}$ is given by

$$\vec{w} = \sum_j \alpha_j c_j \vec{d_j}, \alpha_j \geq 0, \tag{3.3}$$

   Dual optimization problem gives the α_j's. The vector $\vec{d_j}$ are computed in a way that all α_j becomes greater than zero. Such vectors are termed as support vectors because they have the advantage to impact the value of $\vec{w}$.

3. **K-Nearest Neighbor**: K-nearest neighbors uses a basic calculation that stores every single accessible case and groups new cases in accordance with similarity score (e.g., distance functions) [22]. KNN has been utilized as a part of measurable estimation in early 1970's as non-parametric technique. Following are the different distance function applied to check the nearest neighbor.

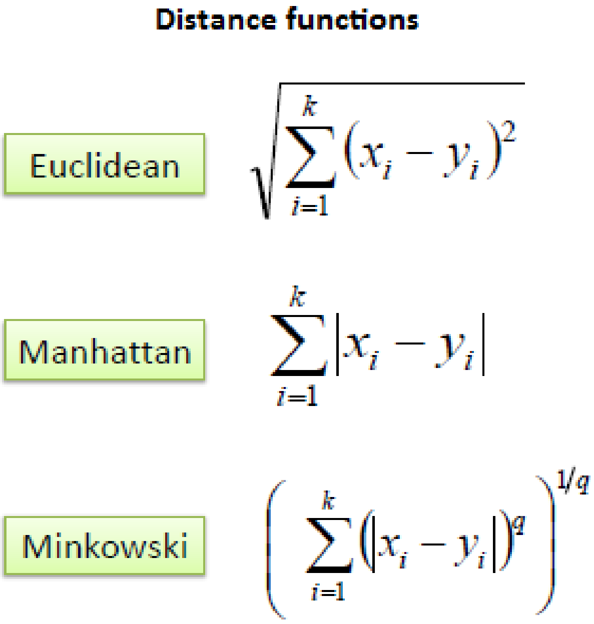
Distance functions
Euclidean
$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
Manhattan
$\sum_{i=1}^{k}|x_i - y_i|$
Minkowski
$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Figure 3.1: Distance Functions used in KNN

Confusion matrix is generated to tabulate the performance of any classifier. This matrix shows the relation between correctly and wrongly predicted reviews. In the confusion matrix shown in table 3.1, TP represents the number of positive movie reviews that are correctly predicted where as FP gives the value for number of positive movie reviews that are predicted as negative by the classifier. Similarly, TN is number of negative reviews correctly predicted and FN is number of negative reviews predicted as positive by the classifier. [23]

From this confusion matrix, performance evaluation parameter such as precision, F-measure, recall and accuracy are calculated. The tabulation of confusion matrix is shown in table 3.1

Precision : It gives the exactness of the classifier. It is the proportion of number

	Correct Labels	
	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Table 3.1: Confusion Matrix

of rightly predicted positive reviews to the total number of reviews predicted as positive.

$$precision = \frac{TP}{TP + FP} \tag{3.4}$$

Recall: It measures the completeness of the classifier. It is the proportion of number of rightly predicted positive reviews to the actual number of positive reviews present in the corpus.

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

F-measure: It is the harmonic mean of precision and recall. F-measure can have best value as 1 and worst value as 0. The formula for calculating F-measure is given in equation 3.6

$$FMeasure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.6}$$

Accuracy: It is one of the most common performance evaluation parameter and it is calculated as the ratio of number of correctly predicted reviews to the number of total number of reviews present in the corpus. The formula for calculating accuracy is given in equation 3.7

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.7}$$

The dataset considered in this study is the polarity movie review dataset which consist of 2000 movie reviews divided equally in to negative and positive reviews [24]. This dataset does not contain separate reviews for training and testing purpose. Therefore, cross validation technique is used which randomly selects the training and testing set. The other dataset considered is IMDb dataset which contains separate reviews for training and testing.

3.3 Proposed Approach

In this study, labeled polarity movie dataset has been considered which consist of 2000 review, divided equally in to negative and positive reviews [24] and IMDb movie review dataset which consist of 25000 movie reviews for training and same for testing [6]. Each movie review first undergoes through a preprocessing step, where all the vague information is removed. From the cleaned dataset, potential features are extracted. These features are words in the documents and they need to be converted to numerical format. The vectorization techniques are used to convert textual data to numerical format. Using vectorization, a matrix is created where each column represents a feature and each row represents an individual review. This matrix is used as input to classification algorithm. For Polarity dataset, we do not have separate reviews for training and testing so in order to resolve this issue, cross validation technique is applied to choose the training and testing set for each fold. Step-wise presentation of proposed approach is shown in the following block diagram 3.2.

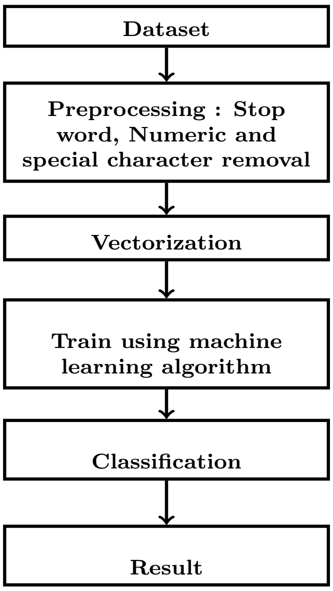
Dataset
Preprocessing : Stop word, Numeric and special character removal
Vectorization
Train using machine learning algorithm
Classification
Result

Figure 3.2: *Diagrammatic view of the proposed approach*

3.3.1 Steps Followed For Classification:

Step 1. The polarity movie review dataset is considered for analysis which consist of total 2000 reviews, divided equally in to negative and positive reviews. For every single review a separate text file is maintained. All 2000 files are first considered to be available in main memory for preprocessing of the reviews. Similarly, IMDb movie review dataset is considered which contains predefined training and testing set of 25000 movie reviews each.

Step 2. The reviews contain a large amount of vague information which are needed to be eliminated. In preprocessing step, firstly, all the special characters used like (!@) and the unnecessary blank spaces are removed. It is observed that reviewers often repeat a particular character of a word to give more emphasis to an expression or to make the review trendy [25]. Words like *"wooowwwwwww, oohhhhhh"* falls in this category. The repetition of characters are also tried to be eliminated in this step.

Most of the words that do not contribute to any sentiment used in English language are termed as stopwords. So, second step in preprocessing involves the removal of all the stopwords of English language.

Step 3. Both the datasets are in its cleanest form. So, features can be extracted from it. The features are tokenized word of a review. These words need to be converted to numerical vectors so that each review can be represented in the form of numerical data. The vectorization of features are done using the following two methods.

- **CountVectorizer**: It transforms the review to token count matrix. First, it tokenizes the review and according to number of occurrence of each token, a sparse matrix is created.
- **TF-IDF**: Its value represents the importance of a word to a document in a corpus. TF-IDF value is proportional to the absolute frequency of a word in a textual document; but it is limited by the occurrences of the word in the corpora.

Step 4. The numeric vectors can be given as input to the classification algorithms. The different classification algorithm used are as follows:

- Naive Bayes(NB) algorithm: It uses probabilistic analysis where features are extracted from numeric vectors. These features help in training of the Naive Bayes classifier model [21].
- Support Vector Machine (SVM) algorithm: SVM plots all the numeric vectors in space and defines decision boundaries by hyperplanes. This hyperplane separates the vectors in two categories such that, the distance from the closest point of each category to the hyperplane is maximum [10].
- K-Nearest Neighbor: K nearest neighbors uses a basic calculation that stores every single accessible case and groups new cases in accordance with similarity score (e.g., distance functions) [22].

Initially, the polarity dataset was not divided between testing and training subsets. So, k-fold cross validation technique is used and k-1 folds of reviews are used for training. Number of folds used are 10.

Step 5. After training of model, each model is tested against the reviews of dataset and a confusion matrix is generated which shows the number of positive and negative reviews that are correctly predicted and number of positive and negative reviews that are wrongly predicted. For each fold, prediction accuracy is calculated based on this confusion matrix and final accuracy is given by taking the highest accuracy obtained among all the folds.

Step 6. The result of each model consists of precision, recall and F-measure as performance evaluation parameters. The confusion matrix and a table containing performance evaluation parameter is generated. Finally, these obtained results are compared with other literatures.

3.4 Implementation and Results

The implementation of above mentioned algorithms are carried out on polarity movie review dataset and IMDb movie review dataset. For polarity dataset K-fold cross validation algorithm is implemented and training set is defined by k-1 fold of dataset respectively for each fold. For each algorithm different performance evaluation parameters are found out on the basis of elements of confusion matrix.

3.4.1 Results of Naive Bayes Algorithm

a) Polarity Dataset

- **Confusion matrix**: The confusion matrix of the fold in which maximum accuracy obtained after implementation of Naive Bayes classification algorithm is shown in table 3.2.

	Correct Labels	
	Positive	Negative
Positive	891	109
Negative	100	900

Table 3.2: Confusion matrix for Naive Bayes classifier on Polarity dataset

- The performance evaluation parameters obtained for Naive Bayes classifier for polarity dataset is shown in table 3.3

	Precision	Recall	F-Measure
Negative	0.90	0.89	0.89
Positive	0.89	0.90	0.89

Table 3.3: Evaluation parameters for Naive Bayes classifier on Polarity dataset

Maximum accuracy achieved in one of the fold of cross validation analysis of Naive Bayes classifier on **polarity Dataset is 0.8953**

b) IMDb Dataset

- Confusion Matrix: The confusion matrix (CM) obtained after classification is shown in table 3.4 as follows:

	Correct Labels	
	Positive	Negative
Positive	11107	1393
Negative	2834	9666

Table 3.4: Confusion matrix for Naive Bayes classifier on IMDb Dataset

- The evaluation parameters precision, F-measure and recall are shown in table 3.5 as follows

	Precision	Recall	F-Measure
Negative	0.80	0.89	0.84
Positive	0.87	0.77	0.82

Table 3.5: Evaluation parameter for Naive Bayes classifier on IMDb Dataset

The accuracy for Naive Bayes Classifier is **0.83092**

3.4.2 Results of Support Vector Machine Algorithm

a) Polarity Dataset

- **Confusion Matrix**: The CM of the fold in which maximum accuracy obtained after implementation of Support Vector Machine algorithm is shown in table 3.6
- The performance evaluation parameters obtained for Support Vector Machine classifier is shown in table 3.7

  Maximum accuracy achieved in one of the fold during cross validation analysis of Support Vector Machine classifier is **0.9406**

	Correct Labels	
	Positive	Negative
Positive	930	70
Negative	49	951

Table 3.6: Confusion matrix for Support Vector Machine classifier on polarity dataset

	Precision	Recall	F-Measure
Negative	0.95	0.93	0.94
Positive	0.93	0.95	0.94

Table 3.7: Evaluation parameters for Support Vector Machine classifier on Polarity dataset

b) IMDb Dataset

- **Confusion Matrix:** The CM obtained after classification is shown in table 3.8 as follows

	Correct Labels	
	Positive	Negative
Positive	11102	1398
Negative	1688	10812

Table 3.8: Confusion matrix for Support Vector Machine classifier on IMDb Dataset

- The evaluation parameters precision, f-measure and recall are shown in table 3.9 as follows

  The accuracy for Support Vector Machine Classifier is **0.884**

	Precision	Recall	F-Measure
Negative	0.87	0.89	0.88
Positive	0.89	0.86	0.88

Table 3.9: Evaluation Parameter for Support Vector Machine on IMDb Dataset

3.4.3 Results of K-nearest Neighbor Algorithm

a) Polarity Dataset

- **Confusion Matrix:** The CM obtained after classification is shown in table 3.12 as follows

	Correct Labels	
	Positive	Negative
Positive	890	110
Negative	125	875

Table 3.10: Confusion matrix for K-Nearest Neighbor classifier on IMDb dataset

- The evaluation parameters precision, F-measure and recall are shown in table 3.13 as follows

	Precision	Recall	F-Measure
Negative	0.90	0.89	0.90
Positive	0.89	0.90	0.90

Table 3.11: Evaluation parameter for K-Nearest Neighbor classifier on IMDb dataset

Maximum accuracy achieved in one of the fold of cross validation analysis of K-Nearest Neighbor classifier on polarity dataset is **0.8993**

b) IMDb Datset

- **Confusion Matrix:** The CM obtained after classification is shown in table 3.12 as follows

	Correct Labels	
	Positive	Negative
Positive	11058	1442
Negative	1482	11018

Table 3.12: Confusion matrix for K-Nearest Neighbor classifier on IMDb dataset

- The evaluation parameters precision, F-measure and recall are shown in table 3.13 as follows

	Precision	Recall	F-Measure
Negative	0.88	0.88	0.88
Positive	0.88	0.88	0.88

Table 3.13: Evaluation parameter for K-Nearest Neighbor classifier on IMDb dataset

The accuracy for K-nearest neighbor classifier for IMDb dataset is **0.88304**

3.5 Comparison of Results Obtained

The table 3.14 shows the comparison of the proposed approach with others literature using Polarity dataset.

Classifier	Classification Accuracy					
	Pang and Lee [12]	**Whitelaw et al.** [13]	**Matsumoto et al.** [26]	**Aue and Gamon** [27]	**Read et al.** [28]	**Proposed approach**
Naive Bayes	0.864	*	*	*	0.789	0.895
Suppport Vector Machine	0.872	0.902	0.937	0.905	0.815	**0.940**
K-nearest neighbor	*	*	*	*	*	0.883

The '*' mark indicates that the corresponding classifier not been considered by author

Table 3.14: Comparison of different existing literatures with proposed approach on Polarity dataset

It can be inferred from the comparison that SVM still performs the best among other classifiers and accuracy of Naive Bayes classifier is slightly improved compared to Read et al. [28] due to extensive text processing of dataset.

The table 3.15 shows the comparison of the proposed approach with others literature using IMDb dataset.

Classifier	Classification Accuracy					
	Pang et al. [9]	**Salvetti et al.** [29]	**Mullen and Collier** [30]	**Beineke** [31]	**Matsumoto** [26]	**Proposed approach**
Naive Bayes	0.815	0.796	*	0.659	*	0.83
Suppport Vector Machine	0.659	*	0.86	*	0.883	**0.884**
K-nearest neighbor	*	*	*	*	*	0.883

The '*' mark indicates that the corresponding classifier has not been considered by author

Table 3.15: Comparison of different existing literatures with proposed approach on IMDb Dataset

The table 3.15 shows that SVM and K-nearest neighbor are equivalent in classification, but SVM has upper hand because of little higher accuracy among others.

Chapter 4

N-Gram Model for Classification

4.1 Introduction

N-Gram Model: It is a strategy for checking of n continuous things from a given grouping of content or speech. The items in sequence can be syllables, phonemes, words or letters depending on the application. This model assists to predict the next item in a sentence or sequence. In sentiment analysis, the n-gram model assists to estimate the sentiment of the document or text. Unigram refers to n-gram of size one, Bigram refers to n-gram of size two, Trigram refers to n-gram of size three, higher n-gram refers to four-gram, five-gram and so on. If the items of sequence are words, n-grams may also be named as **shingles**. The n-gram can be explained using following example:

A typical example of sentiment may be considered as “The show is not a fabulous one”.

Its unigram: “‘The’,‘show’,‘is’, ‘not’, ‘a’, ‘fabulous’,‘one’ ”.

Its bigram: “‘The show’,‘show is’, ‘is not’, ‘not a’, ‘a fabulous’, ‘fabulous one’ ”.

Its trigram: “ ‘The show is’, ‘show is not’, ‘is not a’, ‘not a fabulous’,‘a fabulous one’ ” .

4.2 Results and Analysis

4.2.1 Naive Bayes Classifier

The confusion matrix and evaluation parameters such as precision, F-measure, recall and accuracy obtained after classification using Naive Bayes classifier using n-gram techniques are shown in table 4.1.

Method	Confusion Matrix			Evalution Parameter				Accuracy
Unigram		Correct Level			Precision	Recall	F-measure	83.652
		Positive	Negative					
	Positive	11025	1475	Negative	0.81	0.88	0.84	
	Negative	2612	9888	Positive	0.87	0.79	0.83	
Bigram		Correct Level			Precision	Recall	F-measure	84.064
		Positive	Negative					
	Positive	11156	1344	Negative	0.81	0.89	0.85	
	Negative	2640	9860	Positive	0.88	0.79	0.83	
Trigram		Correct Level			Precision	Recall	F-measure	70.532
		Positive	Negative					
	Positive	10156	2344	Negative	0.67	0.81	0.73	
	Negative	5023	7477	Positive	0.76	0.60	0.67	
Unigram + Bigram		Correct Level			Precision	Recall	F-measure	86.004
		Positive	Negative					
	Positive	11114	1386	Negative	0.84	0.89	0.86	
	Negative	2113	10387	Positive	0.88	0.83	0.86	
Bigram + Trigram		Correct Level			Precision	Recall	F-measure	83.828
		Positive	Negative					
	Positive	11123	1377	Negative	0.81	0.89	0.85	
	Negative	2666	9834	Positive	0.88	0.79	0.83	
Unigram + Bigram + Trigram		Correct Level			Precision	Recall	F-measure	86.232
		Positive	Negative					
	Positive	11088	1412	Negative	0.85	0.89	0.87	
	Negative	2030	10470	Positive	0.88	0.84	0.86	

Table 4.1: Confusion Matrix, Evaluation Parameter and Accuracy for Naive Bayes n-gram classifier

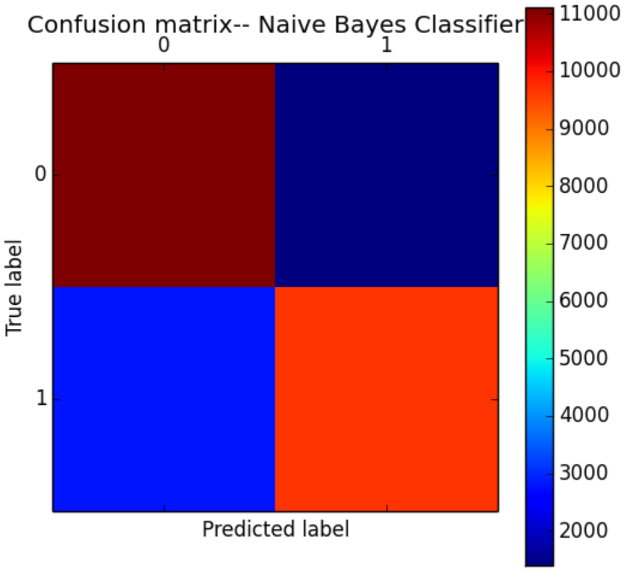

Figure 4.1: Graphical Presentation of Confusion Matrix for Naive Bayes Classifier

Precision-Recall Curve: The Precision-Recall Curve of Naive Bayes (NB) classifier is shown in figure 4.2. The area of curve computed is **0.91**

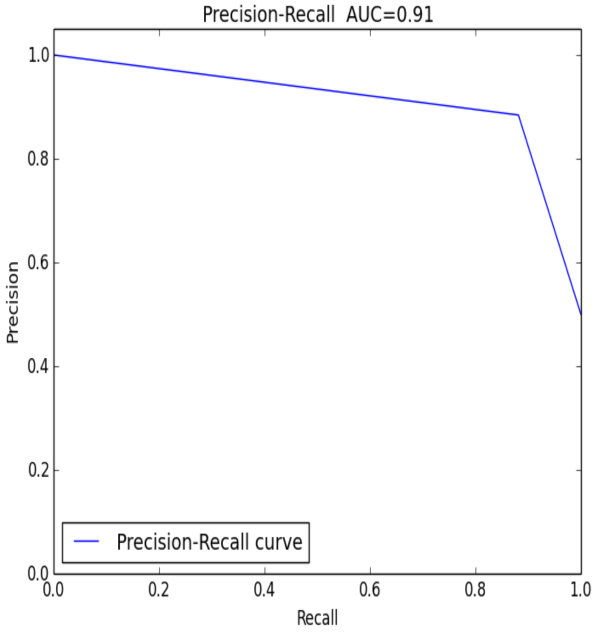

Figure 4.2: Precision Recall Curve of Naive Bayes Classifier

4.2.2 KNN Classifier

The confusion matrix and evaluation parameters such as precision, F-measure, recall and accuracy obtained after classification using KNN classifier using n-gram techniques are shown in table 4.2.

Method	Confusion Matrix			Evalution Parameter				Accuracy
Unigram		Correct Level			Precision	Recall	F-measure	88.48
		Positive	Negative					
	Positive	11011	1489	Negative	0.89	0.88	0.88	
	Negative	1391	11109	Positive	0.88	0.89	0.89	
Bigram		Correct Level			Precision	Recall	F-measure	83.228
		Positive	Negative					
	Positive	10330	2170	Negative	0.84	0.83	0.83	
	Negative	2023	10477	Positive	0.83	0.84	0.83	
Trigram		Correct Level			Precision	Recall	F-measure	71.38
		Positive	Negative					
	Positive	8404	4096	Negative	0.73	0.67	0.70	
	Negative	3059	9441	Positive	0.70	0.76	0.73	
Unigram + Bigram		Correct Level			Precision	Recall	F-measure	88.42
		Positive	Negative					
	Positive	11018	1482	Negative	0.89	0.88	0.88	
	Negative	1413	11087	Positive	0.88	0.89	0.88	
Bigram + Trigram		Correct Level			Precision	Recall	F-measure	82.948
		Positive	Negative					
	Positive	10304	2196	Negative	0.83	0.82	0.83	
	Negative	2067	10433	Positive	0.83	0.83	0.83	
Unigram + Bigram + Trigram		Correct Level			Precision	Recall	F-measure	86.232
		Positive	Negative					
	Positive	11006	1494	Negative	0.89	0.88	0.88	
	Negative	1429	11071	Positive	0.88	0.89	0.88	

Table 4.2: Confusion Matrix, Evaluation Parameter and Accuracy for k-nearest neighbor n-gram classifier

Precision-Recall Curve: The Precision-Recall Curve of k-nearest neighbor classifier is shown in figure 4.4. The area of curve computed is **0.88**

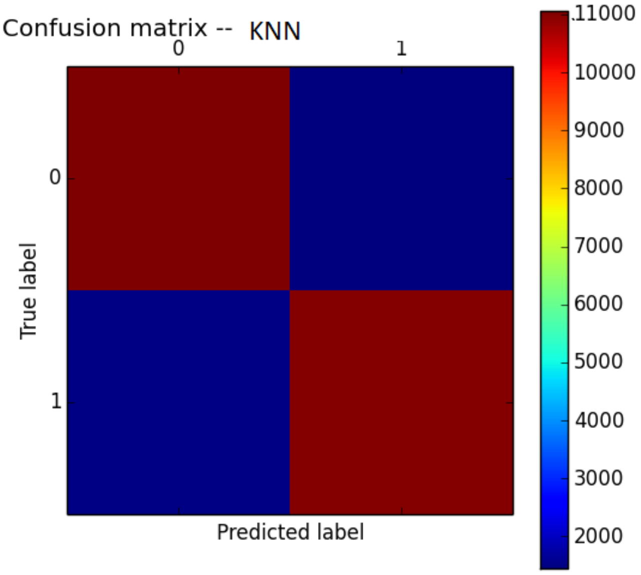

Figure 4.3: Graphical Presentation of Confusion Matrix for KNN Classifier

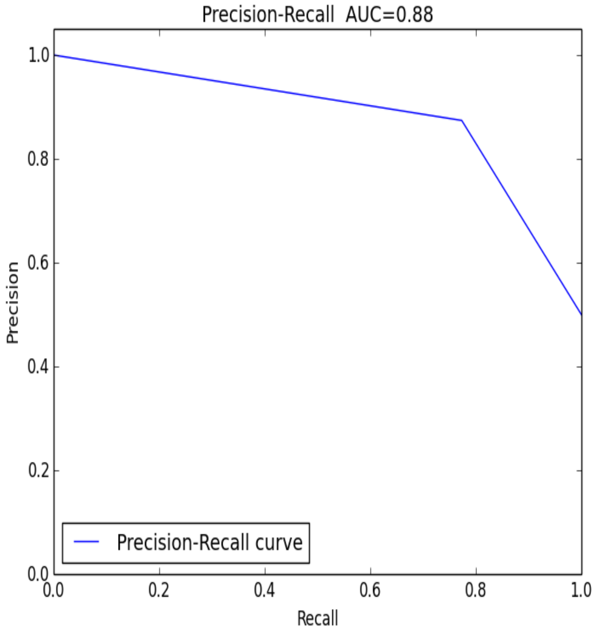

Figure 4.4: Precision Recall Curve of KNN Classifier

4.2.3 Support Vector Machine Classifier

The confusion matrix and evaluation parameters such as precision, F-measure, recall and accuracy obtained after classification using SVM classifier using n-gram techniques are shown in table 4.3.

Method	Confusion Matrix			Evalution Parameter				Accuracy
Unigram		Correct Level			Precision	Recall	F-measure	86.976
		Positive	Negative					
	Positive	10993	1507	Negative	0.86	0.88	0.87	
	Negative	1749	10751	Positive	0.88	0.86	0.87	
Bigram		Correct Level			Precision	Recall	F-measure	83.872
		Positive	Negative					
	Positive	10584	1916	Negative	0.83	0.85	0.84	
	Negative	2116	10384	Positive	0.84	0.83	0.84	
Trigram		Correct Level			Precision	Recall	F-measure	70.164
		Positive	Negative					
	Positive	8410	4090	Negative	0.71	0.67	0.69	
	Negative	3359	9131	Positive	0.69	0.73	0.71	
Unigram + Bigram		Correct Level			Precision	Recall	F-measure	88.884
		Positive	Negative					
	Positive	11161	1339	Negative	0.89	0.89	0.89	
	Negative	1440	11060	Positive	0.89	0.88	0.89	
Bigram + Trigram		Correct Level			Precision	Recall	F-measure	83.636
		Positive	Negative					
	Positive	10548	1952	Negative	0.83	0.84	0.84	
	Negative	2139	10361	Positive	0.84	0.83	0.84	
Unigram + Bigram + Trigram		Correct Level			Precision	Recall	F-measure	88.944
		Positive	Negative					
	Positive	11159	1341	Negative	0.89	0.89	0.89	
	Negative	1423	11077	Positive	0.88	0.89	0.88	

Table 4.3: Confusion Matrix, Evaluation Parameter and Accuracy for Support Vector Machine n-gram classifier

Precision-Recall Curve: The Precision-Recall Curve of support vector machine classifier is shown in figure 4.6. The area of curve computed is **0.91**

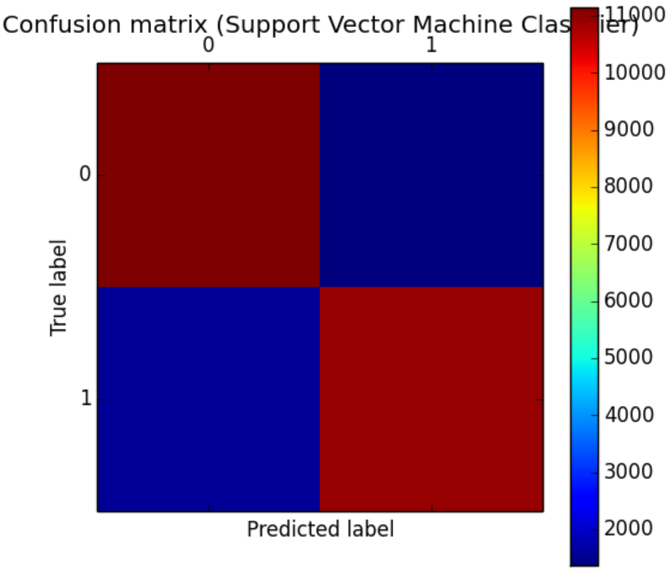

Figure 4.5: Graphical Presentation of Confusion Matrix for SVM Classifier

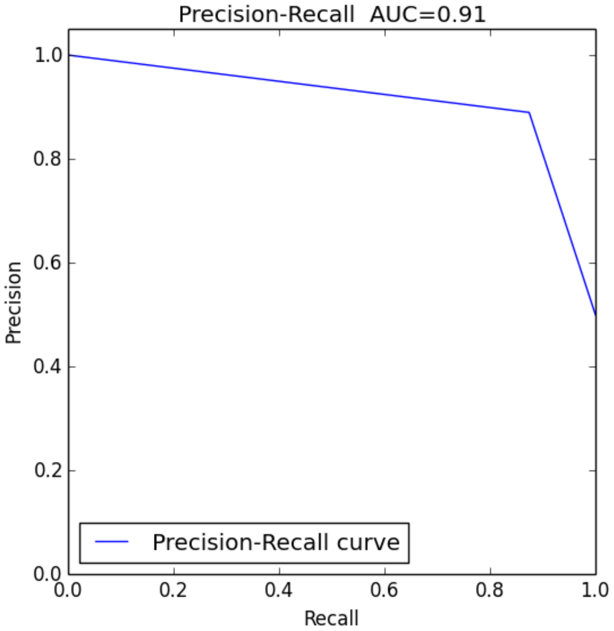

Figure 4.6: Precision Recall Curve of SVM Classifier

4.2.4 Comparative Analysis

The table 4.4 shows the comparison of the result obtained using proposed approach with results available in literature using IMDb dataset and n-gram approach.

Method used		Pang et al. [9]	**Present approach**
Naive Bayes	unigram	81.0	83.6
	bigram	77.3	84.06
	trigram	⊛	70.5
	unigram +bigram	80.6	86.0
	bigram+trigram	⊛	83.8
	unigram + bigram + trigram	⊛	86.2
K-Nearest Neighbor	unigram	⊛	88.4
	bigram	⊛	83.2
	trigram	⊛	71.3
	unigram +bigram	⊛	88.4
	bigram+trigram	⊛	82.9
	unigram + bigram + trigram	⊛	88.3
Support Vector Machine	unigram	72.9	86.9
	bigram	77.1	83.8
	trigram	⊛	70.1
	unigram +bigram	82.7	88.8
	bigram+trigram	⊛	83.6
	unigram + bigram + trigram	⊛	88.9

Table 4.4: Comparative result obtained with different literature using IMDb dataset and ngram approach

The '⊛' mark indicates that the algorithm is not considered by the author in their respective articles.

Pang et. al. [9], have used machine learning techniques viz., Naive Bayes, Support Vector Machine and Maximum Entropy method using n-gram approach of unigram, bigram and combination of unigram and bigram. Here, three different algorithms viz., Naive Bayes, KNN, and Support Vector Machine using n-gram approaches like unigram, bigram, trigram, and their possible combinations like unigram+bigram, bigram+trigram, and unigram+bigram+trigram are carried out. The result obtained in present approach is observed to be better than the result obtained by Pang et.al.

Chapter 5

Conclusions and Future Work

This thesis work makes an effort to classify sentiment reviews using supervised machine learning techniques. In this work, three different supervised machine learning algorithms such as K-nearest Neighbor(KNN), Support Vector machine (SVM) and Naive Bayes (NB) are first implemented to check the prediction behavior in classifying the sentimental reviews. Further, using n-gram Model with the application of above mentioned classifying algorithm, the effect of n-gram in classification is studied. These algorithms are implemented on polarity dataset and IMDb dataset and show better result in comparison with the result published in literatures. It is found out that as the value of ‘n’ in n-gram increases the classification accuracy decreases i.e., for unigram and bigram, The result obtained using the algorithm is remarkably better but when trigram, four-gram, five-gram classification techniques are carried out the accuracy decreases.

Supervised machine learning techniques are studied in this work for classification. In future, it is intended to use unsupervised machine learning methods like neural networks and deep learning methods to check the quality of performance. Only three supervised learning methods have been used in this work; so, other supervised learning techniques such as Artificial Neural network, Random forest, Decision Tree may also be applied to examine the quality of performance.

Bibliography

[1] A. K. Joshi, "Natural language processing," *Science*, vol. 253, no. 5025, pp. 1242–1249, 1991.

[2] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[3] M. Mitray, A. Singhalz, and C. Buckleyyy, "Automatic text summarization by paragraph extraction," *Compare*, vol. 22215, no. 22215, p. 26, 1997.

[4] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Computational linguistics*, vol. 28, no. 4, pp. 399–408, 2002.

[5] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features.* Springer, 1998.

[6] d. IMDb, "Imdb, internet movie database sentiment analysis dataset," 2011.

[7] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the ACL*, 2004.

[8] A. D. Booth, *Machine translation.* North-Holland, 1967.

[9] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[10] P. D. Turney, “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews,” in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424, Association for Computational Linguistics, 2002.

[11] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, ACM, 2003.

[12] B. Pang and L. 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*, p. 271, Association for Computational Linguistics, 2004.

[13] C. Whitelaw, N. Garg, and S. Argamon, “Using appraisal groups for sentiment analysis,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 625–631, ACM, 2005.

[14] S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee, “Semi-supervised learning for imbalanced sentiment classification,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 1826, 2011.

[15] Y. Wang and X. J. Wang, “A new approach to feature selection in text classification,” in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 6, pp. 3814–3819, IEEE, 2005.

[16] S. Argamon, K. Bloom, A. Esuli, and F. Sebastiani, “Automatically determining attitude type and force for sentiment analysis,” in *Human Language Technology. Challenges of the Information Society*, pp. 218–231, Springer, 2009.

[17] Y. Singh, P. K. Bhatia, and O. Sangwan, “A review of studies on machine learning techniques,” *International Journal of Computer Science and Security*, vol. 1, no. 1, pp. 70–84, 2007.

[18] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[19] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10760–10773, 2009.

[20] R. Garreta and G. Moncecchi, *Learning scikit-learn: Machine Learning in Python*. Packt Publishing Ltd, 2013.

[21] A. McCallum, K. Nigam, *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Citeseer, 1998.

[22] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 4, pp. 580–585, 1985.

[23] K. Mouthami, K. N. Devi, and V. M. Bhaskaran, "Sentiment analysis and classification based on textual reviews," in *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*, pp. 271–276, IEEE, 2013.

[24] P. dataset, "Polarity dataset version 2.0, sentiment anaysis dataset."

[25] S. Amir, M. Almeida, B. Martins, J. Filgueiras, and M. J. Silva, "Tugas: Exploiting unlabelled data for twitter sentiment analysis," *SemEval 2014*, p. 673, 2014.

[26] S. Matsumoto, H. Takamura, and M. Okumura, "Sentiment classification using word sub-sequences and dependency sub-trees," in *Advances in Knowledge Discovery and Data Mining*, pp. 301–311, Springer, 2005.

[27] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," in *Proceedings of recent advances in natural language processing (RANLP)*, vol. 1, pp. 2–1, Citeseer, 2005.

[28] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL Student Research Workshop*, pp. 43–48, Association for Computational Linguistics, 2005.

[29] F. Salvetti, S. Lewis, and C. Reichenbach, "Automatic opinion polarity classification of movie," *Colorado research in linguistics*, vol. 17, p. 2, 2004.

[30] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources.," in *EMNLP*, vol. 4, pp. 412–418, 2004.

[31] P. Beineke, T. Hastie, and S. Vaithyanathan, "The sentimental factor: Improving review classification via human-provided information," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 263, Association for Computational Linguistics, 2004.

Dissemination of Work

Accepted

1. Abinash Tripathy, Ankit Agrawal, Santanu Kumar Rath. Requirement Analysis Using Natural Language Processing, *5th Internatoinal Conference on Advances in Computer Engineering* , Kochi, India, 2014.

2. Abinash Tripathy, Ankit Agrawal, Santanu Kumar Rath. Classification of Sentimental Reviews Using Machine Learning Techniques, *3rd International Conference on Recent Trends In Computing,* New Delhi, India, 2015.

Communicated

1. Abinash Tripathy, Ankit Agrawal, Santanu Kumar Rath. Classification of Sentimental Reviews Using N-gram Machine Learning Approach, *Expert System with Application Journal Elsevier*

2. Abinash Tripathy, Ankit Agrawal, Santanu Kumar Rath. Classification of Sentimental Reviews Using Supervised Machine Learning Techniques, *Expert System with Application Journal Elsevier*