Comparison of Lexicon based and Naïve Bayes Classifier in Sentiment Analysis

Rohini.V[1] Merin Thomas[2]
[1]M.Tech. Student [2]Assistant Professor
[1,2]Department of Computer Science and Engineering
[1,2]Don Bosco Institute of Technology, Affiliated by VTU

Abstract— In the recent years Sentiment analysis (SA) has gained momentum by the increase of social networking sites. Sentiment analysis has been an important topic for data mining, social media for classifying reviews and thereby rating the entities such as products, movies etc. This paper represents a comparative study of sentiment classification of lexicon based approach and naive bayes classifier of machine learning in sentiment analysis.
Key words: Sentiment Words, Machine Learning

I. INTRODUCTION

People have started to share information about entities such as products, movies etc, through different kinds of social media. This information plays an important role in finding whether a entity is good or bad. Sentiment analysis is also known as opinion mining.

It is a task of identifying the orientation of opinion or sentiment words in a text. Sentiment analysis can be of three level document level (such as blog), sentence level (such as comments) and word level. In this paper we compare the two methods of sentiment analysis lexicon based approach and machine learning approach

II. DEFINITION

Sentiment analysis is the task of finding sentiment words in a given piece of text. It Comprises three functions i) dividing sentences into words, ii) identification of sentiment in sentence using a sentiment analysis tool, iii) finding positive and negative polarities of the sentiment words and rating the reviews as positive or negative based on their polarities or score. Sentiment classification looks, for words or emotional states such as sweet, happy, angry, and sad. Let us consider an example for each type of reviews, such as "the movie was simply awesome", and "it was a horrible movie is positive review and negative review respectively for the movie.

III. CLASSIFICATION OF SENTIMENT ANALYSIS

Sentiment analysis can be classified into two approach lexicon based approach and machine learning approach. Lexicon based approach deals with searching the axioms or sentiment words form the sentence and comparing with seed words, it has two branches dictionary and corpus based approach.

Corpus based in turn into semantic classification algorithm. Machine learning deal with sample review for rating the sentiment words, it is split into two forms first, unsupervised approach - this compares each word of the text with maximum valued positive word and negative word for rating. Second, supervised approach - this uses equations to obtain the sentiment. Under supervised is the Naive Bayes classifier algorithm.

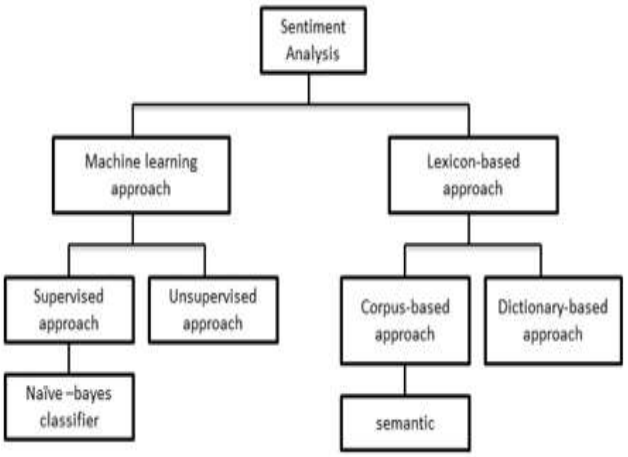

Fig. 1: Sentiment Analysis Hierarchy

IV. LEXICON BASED APPROACH

Sentiment words are used in many sentiment classification tasks. Positive and negative sentiment words are used to express some desired and undesired states respectively Lexicon based approach deals with searching the axioms such as adjective, adverb, noun etc form the sentence and comparing with seed words or training data set with its corresponding polarity in the database of words There are three main approaches to collect the sentiment word list. Manual approach, it is very time consuming and automated approach. The two types of automated approaches are presented in the following subsections.

A. Dictionary-based approach

Sentiment words are collected manually to form a small list, which is later developed by searching more words from a known corpora wordnet. Wordnet is a corpora the produces synonyms and antonyms for a word. The new words found excluding the seed words are included to the list. The process continues until no new words are found from the corpora. The major drawback of dictionary-based approach its inability of finding sentiment words with respect to a domain or a feature.

B. Corpus-based approach

The Corpus-based approach helps to overcome the drawback of dictionary based approach in finding sentiment words with feature specific orientations. It depends on certain patterns that occur together along with a seed list of sentiment words to find similar sentiment words in a large corpus.

Corpus-based approach has a disadvantage that corpus-based alone is not as effective as the dictionary-based approach because it is hard to cover all English words, but this approach has a major advantage that it can help to find domain and feature specific sentiment words using a

domain corpus. The corpus-based approach is performed using statistical approach or semantic approach as illustrated in the following subsections:

C. Semantic approach

This method of finding co-occurrence seed sentiment words using semantic technique is done by deriving polarities using the co-occurrence of axioms as adjectives, adverbs in a corpus, It is also possible to use a document on the web as the corpus for the construction of seed sentiment words and covers the maximum unavailable words, if the corpus with small list of seed words is used as training data set.

The Semantic approach gives sentiment values directly and gives similar sentiment values words that are close semantically. Tool for computing the similarity between words is WordNet; it provides different kinds of semantic relationships between words used to calculate sentiment polarities. WordNet could be used too for obtaining a list of sentiment words by iteratively expanding the initial set with synonyms and antonyms and then determining the sentiment polarity for an unknown word by the relative count of positive and negative synonyms of this word.

1) Requirements

The training data set or seed words in the database consists of the following elements

- Sentiment words, these are the words that are a collection of adjective, adverbs that present a semantic meaning to the sentence or text ,words such as "good, bad, happy, worst" are sentiment words
- Negation words, if a sentiment word is preceded by a negation word such as not and if it is located at a distance of almost 2 places from the sentiment word, then the polarity of the sentiment word is reversed to its complete opposite polarity. Here in this example "the movie is not good" though the word has positive score (4), since it is preceded by negation word "not", it changes to (-4) resulting in negative polarity
- Blind Negation Words, these are the words that present some absence or incompleteness of some desired state. Here in this example "TV remote needs a better remote" the blind negation word "needs" make the sentence negative irrespective of its location in the sentence.

2) Preprocessing

- Tokenization, in this method quotes, punctuations are removed from sentence.
- Remove stop words, the words such as "a" , "is"," the"," this" and so on are consider as stop words. Stop words do not provide any meaning to the text or sentence hence they are removed from the sentence.
- Stemming is used for reducing words to their word stem. For example, stemming reduces the word running, runs to run.
- exaggerated word shortening, words which have repeated letter more than once are reduced to the word, for example, "gooood" is reduced to "good"
- POS Tagging, the words of sentence are presented along with part of speech. There are nine parts of speech. They are articles, nouns, pronouns, adjective, verb, adverb, conjunction, prepositions, and interjection. The part-of-speech words noun, adjective, verb, adverb are represented as NN, JJ, VBZ for noun, adjective, adverb respectively. This example shows the sentence after pos tagging "this_DT is_VBZ good_JJ book_NN ",

The algorithm is referred from "Serendio: Simple and Practical lexicon based approach to Sentiment Analysis" Prabu Palanisamy, Vineet Yadav and Harsha Elchuri

Algorithm: Sentiment Calculation
Data: Preprocessed Data
Output: Positive, Negative

```
If BlindNegat then
    return negativity
else
    if SentiList and NegatList then
        foreach word int SentiList do
            if word is almost the distance of 2
                from SentiNegat then
                Revert the Polarity of the word
            end
        end
    else
        if SentiList then
            add the sentiList to the
            poslist
        end
    end
end
SeniSum=0;
foreach word in SentiList do
        SenitSum = SentiSum+ sentiment of word;
end
If SentiSum > 0 then
        SentiType = "positive";
else
        SentiType="negative";
end
return SentiType;
```

In the above algorithm, the following three conditions are handled – First, if blind negation is found, then the sentence is considered as negative review. Second, If positive sentiment (such as good) word is preceded by a negation then the polarity is reversed, for negative sentiment word it remains unchanged. Third, if the only sentiment words then, their corresponding polarities are retrieved from training data set then, perform sum of polarities to get the final value for the review

V. MACHINE LEARNING APPROACH

Machine learning is a field of artificial intelligence, it explores the study of algorithms that can learn from and make certain predictions on the data. Machine Learning Approach implements the concept of probability for sentiment words in classifying them into two classes, positive or negative classes of reviews. Machine Learning Approach is subdivided into two branches Supervised and unsupervised learning.

A. Unsupervised learning

The supervised learning methods depend on comparing the words of sentence with the maximum positive value word and the maximum negative value using point wise Mutual Information equation

B. Supervised learning

The supervised learning methods depend on training data. Probabilistic classifiers use mixture models for classification. The mixture model imagines that each class is a component of the mixture. Each mixture component is a generative model that gives the probability of sampling a particular term for that component. These kinds of classifiers are also called generative classifiers. Three of the most famous probabilistic classifiers are discussed in the next subsections.

C. Naïve Bayes Classifier (NB)

The Naïve Bayes classifier is the simplest and most commonly used classifier, it is also known as baseline algorithm. Naive Bayes Classifier technique is based on the Bayesian theorem. This method computes the posterior probability of a class, based on the distribution of the words in the sentence. This model ignores the position of the word in the sentence. The words of the sentence are collected together called Bag of words. Each words positive and negative polarity is calculated by the equation given below

$$P(label|features) = \frac{P(label) * P(features|label)}{P(features)} \quad \text{-----(1)}$$

Using a set of sample reviews as training data, the words from the sample reviews collected together results the bag of words.

A posterior probability is the probability of assigning the observations into groups of data after relevant evidence is taken into account. A prior probability is the probability that an observation will fall into a group before you collect the data. Prior probability would express the belief before collecting the evidence.

P (label |feature) is the posterior probability of a label. P(features |label) is the likelihood that a random feature .P(features) is the prior probability that a given feature set is occurred. Given the Naïve assumption which states that all features are independent, the equation could be rewritten as follows:

$$P(label|features) = \frac{P(label) * P(f1|label) * \ldots.. * P(fn|label)}{P(features)} \quad \text{------(2)}$$

1) Requirement

A set of sample review ranging from 100 to 200 reviews based on certain domain such as movie, product etc and its corresponding value. The values are either 0 or 1. The value one represents positive and zero as negative

The algorithm is referred from "Sentiment Analysis Tool using Machine Learning Algorithms", I.Hemalatha1, Dr. G. P Saradhi Varma2, Dr. A.Govardhan3

Algorithm : Naïve Bayes Classifier

Input: Data M= {m1, m2, m3.....}

Output: Positive, Negative

1) Step 1: Divide a message into words
   mi ={w1, w2, w3.....}
2) Step 2: If wi belongs to NT
   Retrieve +ve and -ve polarity
3) Step 3: Calculate probability of +ve polarity of wi With total +ve polarity of M
   a. $$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$
4) Step 4: Calculate overall polarity of a Word= (+ve polarity) + (-ve polarity)
5) Step 5: Repeat step 2 until end of words
6) Step 6: Add the polarities of all words of a message
7) Step 7: Message can be positive or negative
   Repeat step 1 until M NULL

In the above algorithm, first divide sentence into words, and collect them to form Bag of words. Retrieve the probability value of positive review and negative review from sample review data set, for each of the word in sentence. For example, "the movie is good", has both positive review and negative review for words movie, good and so on. Calculate the posterior probability of both positivity and negativity for each word in the review. If the word has positive posterior probability then, it is a positive review else negative. Repeat the process until probability all the words of the sentence are compute and add the polarities to get the result, positive or negative.

VI. APPLICATIONS

- Product review: The review based on certain product is obtained from the customers in the form of feedback of the product, these review are processed in sentiment analysis
- Social Networking Sites: Networking sites such as twitter, face book and so on use comment as the main source for sentimental analysis
- Politics during elections: The role of a politician can be judged by analyzing the complaints of people during elections
- Movie review: Analyzing the review of the movie to decide whether the movie is a hit or not. This can even help us in assigning the rate for the movie tickets.
- Health care: Sentiment analysis is used to analyze biomedical text related to patient medical data
- Improve Customer Service: Sentiment Analysis gives useful insights about your current and future customers' purchase preferences, brand affiliations, topics of interests, opinions, point of views on discussions, likes and dislikes in products/ services and much more. This useful information lets organizations to drastically improve their customer service and engagement strategies by building on the positive sentiments and formulating methods to combat negative sentiments.
- Defence Services: The software for Twitter, called the Dynamic Twitter Network Analysis (DTNA), is now being field-tested by three Defense Department units overseas to help gauge public opinion in some of the world's hot spots. The software pulls in data from the public Twitter feed, then sorts it, live, by phrases, keywords or hash tags. The program is continuously updated, integrating a mapping feature and geo-tagged information. Intelligence officers could use DTNA to understand people's moods about a topic, or hopefully prevent or simply respond faster in any future U.S. embassy attacks.

VII. IMPLEMENTATION

A. Lexicon – based Implementation

The implementation of lexicon based algorithm produces the following output, the words in database for the input and its scores are shown in the seed table (2) below. The steps of processing and tagging are shown in processing table (1).

Fig. 1: Lexicon-based Positive review

sentence	The movie is awesome I like watching it	Direction was bad movie story line is not catchy
Stop words	Movie awesome like watching	Direction bad movie story line not catchy
Pos Tagger	movie_NN awesome_JJ like_JJ watching_VBG	direction_NN bad_JJ movie_NN story_NN line_NN not_RB catchy_JJ
Score	3+2 = 5	-2+(-2) = -4
Type	Positive Review	Negative Review

Table 1: Processing Table

Seed words here are few sentiment words from the database of the sentiword.net, along with their corresponding positive and negative score. Input is preprocessed and this processed data is input to lexicon- based algorithm known as sentiment calculation. In the preprocessing step, stop words, punctuations and stemming words removed. Tagging of the sentence is done by pos tagger it transform the sentence into tokens by detecting axioms such as adjectives, nouns, adverb, which are the key essentials to find sentiment words in the text. In the above input for positive review, "the movie was awesome and I like watching it" stop words, punctuations removed and in the sentence an adjective (awesome) with the tag (_JJ) is retrieved by POS Tagger. The adjective is compared with the database and its corresponding score is retrieved. Similarly the scores of adverbs, nouns are obtained and of all scores in the text are performed, if it is a positive value then, it as positive review or as negative review, Since it is a positive valued sentence the input is a positive review. Similarly in the above input for negative review, the words bad, catchy which are adjectives in the sentence produce a negative and positive value as shown in the seed table(2). Though the word catchy has a positive value 2, its value is reverted to -2, because it is preceded by the negation word 'not'

Fig. 2: Lexicon-based Negative review

Word	score
adventure	1
awesome	3
awful	-3
awkward	-1
bad	-2
catchy	2
good	4
like	2

Table 2: seed words Table

B. Naïve Classifier Implementation

Words in the review form the bag of words. Each word is compared with the words of sample review collected in the database and retrieves its rating. A word such as movie is present in 4 of 12 sample positive reviews then likelihood P(x/c) of positive review for the word movie is 4/12 similarly likelihood of negative review is 2/8..

Fig. 3.1: Naïve Bayes Positive review

Bags of word	pos	Neg	P(X)
movie	4/12	2/8	6/20
awesome	3/12	0/8	3/20
like	2/12	4/8	6/20
watching	3/12	2/8	5/20
P(C)	12/20	8/20	

Table 3 Probability of words in positive review

Class prior probability P(c) is the probability of total positive review to the total number of review in the database. Class prior probability for negative review is calculated in the similar way. Predictor prior probability P(x) is probability of total review for a word such as movie to the total sample review in the database. The posterior

probability of each word is calculated, if the value is positive then it is positive review else negative

Fig. 3.2: Naïve Bayes Negative review

Bag of words	pos	Neg	P(X)
direction	4/18	2/20	6/38
bad	0/18	6/20	6/38
movie	4/18	2/20	8/38
story	3/18	2/20	5/38
not	0/18	4/20	3/38
line	4/18	3/20	7/38
catchy	3/18	0/20	3/38
P(C)	18/38	20/38	

Table 4 Probability of words in negative review

VIII. Comparison of Lexicon and Naïve Bayes Classifier

Fig. 5(a): Comparison lexicon and naïve classifier

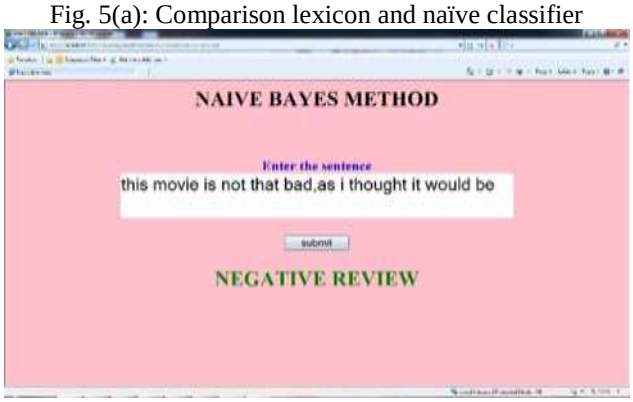

Fig 5(b): Comparison on lexicon and naïve classifier

IX. Results

	POSITIVE	NEGATIVE
Precision	0.9361	0.9245
Recall	0.9351	0.8215
Accuracy	100%	86%

Table 5: Results

X. Conclusion

Lexicon–based method is accurate than Naïve bayes classifier when sentence is processed completely with training set data and retrieve their respective scores. Naïve bayes classifier inefficient than Lexicon–based method algorithm in accuracy but gives better results in cases where data is incomplete or uncertain and has a wide application

References

[1] Prabu Palanisamy, Vineet Yadav and Harsha Elchuri approach to Sentiment Analysis, "Serendio: Simple and Practical lexicon based"
[2] I.Hemalatha1, Dr. G. P Saradhi Varma, Dr. A.Govardhan "Sentiment Analysis Tool using Machine Learning Algorithms".
[3] B. Pang and L. Lee. "Opinion mining and sentiment analysis" Foundations and Trends in Information Retrieval, 2(1-2):1{135, 2008.}
[4] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning Techniques. [2002]
[5] Peter Turney. 2002. "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews". In Proc *of the ACL*.
[6] Comparative Study of Classification Algorithms used in Sentiment Analysis, Amit Gupte, Sourabh Joshi, Pratik Gadgul, Akshay Kadam,
[7] Machine Learning with Naïve Bayes Available:http://blog.datumbox.com/machine-learning-tutorial-thenaive-bayes-text-classifier
[8] Aditya Joshi, Balamurali Pushpak Bhattacharyya, "A Fall-back Strategy for Sentiment Analysis in Hindi: a Case Study", Department of computer science, IIT Bombay
[9] Bas Heerschop, Frank Goossen, Alexander Hogenboom, "Polarity Analysis of Texts using Discourse Structure", Erasmus University
[10] Minqin Hu and Bing Liu, "Minig and Summarizing Customer reviews", Department of CS, University of illinios at chicag.

