Sentiment Analysis - A New Approach

Samahriti Mukherjee, Srijan Chattopadhyay, Rudrashis Bardhan

Indian Statistical Institute, Kolkata

Abstract

Sentiment Analysis, or Opinion Mining, or Emotion AI, is an approach to determine the emotional tone behind the data using NLP (Natural Language Processing), Text Analyzing etc. This paper focuses on the different processes and some new algorithms.

Introduction

Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text,as mentioned in the **Dictionary of Oxford**.A basic task in sentiment analysis is to determine the polarity of a document/sentence/word,etc.This task is commonly defined as the classification problem.This problem can sometimes be very difficult to judge mathematically because of the subjective nature of the words/phrases may depend on the context **(Source:Wikipedia)**.

This problem may get more difficult when we move towards sentiment analysis of words to sentences and then to documents.For example,say there are some opinions about a new scheme of government,so there will be positive,negative and neutral opinions about the same.So as a whole,we cannot judge the polarity of the document.Even,if some sentence tells partially about the good sides and the bad sides simultaneously,then judging the sentence on the basis of sentiment from each word will be controversial.So,it is better to treat such sentences as neutral.

So,we cannot always judge properly whether a word/sentence/document belongs to a class of opinions. But,the process of this sentiment analysis can be made more efficient by implementing better algorithms.Here,we will be briefing some existing algorithms and will be proposing some new algorithms.

There are certain steps of Sentiment Analysis.The primary step is data cleaning followed by Feature selection, classification techniques and analysis.

In the first section(Section **1**)),there is discussion about the detailed process of data cleaning and data preparation.

In the second section(Section **2**)),there is discussion about the process of Feature Selection, which is the process of selecting those features only which are the most important for the sentimental data. That is without losing much data, we want to review the whole data by removing some less important features for computational purpose. So, it is basically the dimensionality reduction of the dataset. We have discussed some existing methods for that, **eg. Term Frequency of words based feature selection, Mutual Information based feature selection, Chi Square(χ^2), LSI(PCA based method)**. Then we have proposed a new unsupervised method that uses **SVD** to find the singular values of the variance-covariance matrix and then only taking the important ones. Lastly, we proposed a method on information theory, that is the **entropy based method**. There are also different existing methods for feature selection. It is an active area of research nowadays. We have just presented some of those and proposed some new ones.

In the third section(Section **3**)),there is discussion about the process of Classification Techniques.After selecting the important features,we now perform classification using different algorithms to classify the datas into different clusters or different classes. Some of these are probabilistic methods, i.e. depending on probabilistic model, some are statistical, some takes theories from both information theory and probability and also there are some unsupervised clustering methods. Some of the known probabilistic methods are **Multinomial Naive Bayes', Gaussian Naive Bayes'**. We have proposed a new probabilistic method named **SSR-Distribution Classification Algorithm($\tilde{\alpha}$)**. Some statistical or ML Algorithm based classification are **SVM(Support Vector Machine), Random Forest(Decision Tree included)**. Then we have proposed a method which is based on both information theory and probability, or more specifically mutual information and conditional probability.

At the end, we have gone through the implementation of some of these algorithms, mainly the new proposed ones, and noted the results we got.

Methodology

We will be describing first the general step to follow for sentiment analysis.

Indian Statistical Institute, Kolkata

1)Data Cleaning and Data Preparation:

- First we convert all the upper cases letters into lower cases.
- Then we remove all the punctuation.
- After that, we remove all the digits from the sentences.
- Next we remove all the very common words from the sentences.(eg.i, me, my, myself, we, our, yours, ourselves etc.)
- Now we remove all the URLs from our data.
- And at the end we removed all the extra white spaces from the document.
- Hence, our document is now ready for the analysis.

2)Feature Selection:

Sentiment Analysis primarily is a classification problem which is done by judging the sentiment behind a sentence.If some training data is available where the sentiments behind a sentence are mentioned,then we work with that available data.In other cases,there are different methods we can follow(eg.We can count the frequency of a particular word and by dictionary, we can judge the sentiment of the words and then combining the words, the sentiment behind the sentence. The primary target of feature selection method is to reduce the dimension of the data by removing words which are too rare in the given dataset, which increases the accuracy of the analysis also in some cases. There are various ways of selecting the features-

- **Term Frequency of Words** : In this method,we just count the frequency of each word in the total dataset. Then we set a threshhold after viewing the range of the frequency of the words and only take those words which crosses the threshhold. Hence, we can consider only the words which are very important and not those which are rare, i.e. less important. So, one can reduce the dimensionality of the dataset without losing any important information.
- **Mutual Information**: Mutual information is a very useful concept of Information Theory. For 2 random varaiables X and Y, the Mutual Information between X and Y is defined as the following:

$$I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)} p(y) = E(\log \frac{p(x,y)}{p(x)p(y)}) \tag{1}$$

Or, it can also be written as

$$I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} = E(\log \frac{p(x|y)}{p(x)}) \tag{2}$$

where, $p(x,y)$ is the joint probability mass function of X and Y, $p(x)$, $p(y)$ are probability mass functions of X and Y respectively and $p(x|y)$ is the conditional probability of X given Y.
We will use the mutual information to select the most important words for a specific class.We first calculate the number of times a word occurs in a positive sentence or a negative sentence,etc.Then we create a table for each word and the number of times it occurs in different types of sentences.Then we use (2) in the following way:
Let a randomly selected word be x and we want to find how important the word w is for (say) the positive class. Define, $X = \mathbf{1}$(x is the word w) and $Y = \mathbf{1}$(x belongs to the positive class) [where, $\mathbf{1}$ is the indicator function] And, then we calculate the mutual information between these random variables X and Y which denotes the importance of the word w in the positive class. An important thing is to note that, the mutual information is always non negative. So, we just select the top k words with the highest mutual information with the positive class, where k is predetermined by the user. This reduces the data for the positive class to only the most important words needed for positive sentiment to be justified. The same thing we can apply for each data set and at the end, we will get the different sets of words for each class, there maybe overlap also. That means, if a word is important to determine a positive sentiment, it can also be justifying for a negative sentiment. So, in this way we can deduce the dimensionality using 'mutual information'
- **Chi Square (χ^2)**: χ^2 feature selection is very similar to the Mutual information based feature selection technique. Let there be n total number of documents in the dataset and say, we wish to see the important words for the positive class.So,then the χ^2 statistic of a word w and the positive class is defined as follows:

$$\chi^2_w(positive) = \frac{n \cdot P(w)^2 (P(w|positive) - P(positive))^2}{P(w)(1-P(w))P(positive)(1-P(positive))} \tag{3}$$

where $P(w)$ is the probability that a random selected word from the document is w, $P(positive)$ is the probability that a randomly selected word chosen from the document falls in the positive class and $P(w|positive)$ is the conditional probability of positive class for a randomly selected word to be w.
Now the next process of selecting those features based on the χ^2 statistic values is similar to that of the Mutual Information based feature selection.
- **Some unsupervised learning methods**:As mentioned before,the main goal of feature selection is to reduce the dimensionality of the dataset for computational purpose.As mentioned in the paper **[Sentiment analysis algorithms and applications: A survey](Walaa Medhat, Ahmed Hassan, Hoda Korashy)** [1], LSI (Latent Semantic Indexing) is a popular method of unsupervised feature selection process. LSI method thinks the text space to a vector-space and it transforms the axis system to a linear combination of the documents.PCA (Principal Component Analysis) is used for this process.Basically, it measures the variance-covariance matrix and then performs its eigen-decomposition with the eigenvalues arranged in a decreasing order.So the first eigenvalue will be of the highest importance and so on.The span of the text-space will be maximum along the eigenvector corresponding to the maximum eigenvalue and so on.So in this order,we just choose those eigenvalues which are above

a certain threshold and then try to represent the actual data as an approximate linear combination of those significant eigenvalues.Inspired by this idea,we are hereby proposing an Unsupervised Learning Method which is very similar to that of PCA.

Our proposed method:First of all,any $A_{m \times n}$ matrix over real Field can be factorised as

$$A_{m \times n} = U_{m \times m} \Lambda_{m \times n} V^T_{n \times n} \quad (4)$$

where, U, V are orthogonal matrices, i.e. $UU^T = U^TU = I$(identity matrix), V^T is the transpose of V, and Λ is a diagonal matrix with non negative diagonal entries $\lambda_1 \geq \lambda_2 \geq ..... \geq \lambda_r > 0$ These are called the singular values of A. This is called the Singular Value Decomposition(SVD) in Real field. Now, we first find the variance covariance matrix of the given dataset and then find the SVD of that to find the singular values.
Now, by Ekhart-Young theorem, if $A = \lambda_1 u_1 v_1^T + ... + \lambda_r u_r v_r^T$, then for $k \leq r$, the closest approximation of the matrix A will be $A_k = \lambda_1 u_1 v_1^T + ...\lambda_k u_k v_k^T$, which can be intuitively explained because of the ordering of the singular values with respect to their priority, so the maximum information is carried by the first singular value, the second maximum is by the second singular value and so on. So, to reduce the dimensionality of the variance-covariance matrix, this method also can be used and to find the closest approximate matrix for a pre-defined k and can work with that. Then, can calculate the score of the words with respect to that matrix and then can perform sum of square of distance of the score for each word and arrange them accordingly to find the maximum information carrying words with respect to SVD.

- **Our proposed 2nd method of Feature Selection:** Entropy is a very important concept of Information Theory. Entropy measures the uncertainty of a random variable. The analogy we have found here is that, if we count each word as a random variable and negative, positive, anger, irrelevant, neutral etc. as the different measurement of the words, then the entropy will determine that how random is the word for a class, i.e. if the entropy is close to 0, that means that the word is important for exactly one class and hence can determine accurately the sentiment of that class, i.e. if that word is present in some sentence, then we can surely say something about the sentiment of the sentence. Keeping this idea in mind, we are hereby proposing this method to find the entropy of each word and only take those values which are under a certain threshold.
The entropy $H(X)$ of a random variable X is defined as $H(X) = -\sum_x p(x) \log p(x)$, where the summation is over the support of X[i.e. where $p(x) > 0$], which can be also written as $E(-log(p(x)))$.
So, we just find the entropy of each word and then fix a threshold α and only take the words having entropy less than α. Then, for each sorted word, see the maximum component and put that in that sentiment class. Here, clearly there will be no overlapping between any two classes. But, a certain probability value of word to be in that class will be there implicitly, on which we will be classifying the new data or the testing data.
There are many feature selection algorithms also and there will be many, since it is an active field of research (NLP, Natural Language Processing). So, let us now go to the next step of the Sentiment Analysis.

3)Classification Techniques:

Classification is performed on structured or unstructured data.Here the main goal is to categorize data into a number of classes and to identify which class it falls into based on the features.An algorithm that does the classification process methodically, is called a classifier.In most of the cases,the classifier represents some mathematical function.Classification algorithms can be generally of three types:

- **Probabilistic Classification**: That is classification based on probability of occurrence of a class.(for example algorithms like Naive Bayes(both multinomial and Gaussian divides into classes based on conditional probabilities of a class).

  Naive Bayes: Naive Bayes is a machine learning model based on Bayes' Theorem. There are 2 types of Naive Bayes' classifications:

  Multinomial Naive Bayes: This algorithm uses the idea of bayesian statistics, i.e. the posterior and prior distribution, and the formula used is nothing but the baye's theorem of conditional probability. That is

  $$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (5)$$

  So, the question is, how will we use this for classification problem? So, we are given now the important features, and we have to determine the which class has the maximum chance of having that feature. Here the comes the probabilistic part. That is, to which class the feature belongs with the maximum probability. So, we just determine the conditional probability of the class given the feature. Here it is

  $$P(class|feature) = \frac{P(feature|class)P(class)}{P(feature)} \quad (6)$$

  Now, we know $P(feature)$ that is how many documents contains the feature divided by total no of documents. $P(feature|class)$ is the conditional probability of the documents containing the feature given a class, that is also known to us and $P(class)$ is the probability that a word belong to that class. So, we can easily find $P(class|feature)$.
  So, by this process we attach some probabilities to each word that with what chance it belongs to which class.
  Now, when we get a new sentence, we just find the words which are in the intersection of the sentence and our selected words.
  Now, the main point of multinomial naive bayes is that it assumes all the features to be independent.
  So, for judging the new sentence, $P(class|sentence)$ will be just the product of those words times $\frac{P(class)}{P(words)}$.

Hence, we now choose the maximum probability and conclude that the sentence belongs that class with corresponding probability.

Gaussian Naive Bayes: Gaussian Naive Bayes is another variant of Naive Bayes when we are working with continuous data instead of categories.Here the assumption is that the continuous values associated with each class are distributed according to a gaussian distribution.That is $x_i|y \sim N(\mu_y, \sigma_y^2)$, where μ_y is the mean of x_i associated with the class y and σ_y is the bessel corrected variance(i.e. replacing n by (n-1) in the denominator) of the x_i's associated with the class y. The likelihood of the features are assumed to be:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (7)$$

Now, when we are given a new sentence, say the common features are $x_1, x_2, .., x_k$. So, we have to determine now the conditional probability of a class y given that features. We can calculate it by the Bayes theorem of total probability.

Here also, the independence of the features is assumed to be true.

So, according to the theorem of total probability,

$$P(y|sentence) = P(y|x_1, x_2, .., x_k) \quad (8)$$

$$= \frac{P(y)\prod_{i=1}^{k} P(x_i|y)}{\sum_{allclasses} P(y)\prod_{i=1}^{k} P(x_i|y)} \quad (9)$$

So, we just find the class for which the above probability is maximum and then do the rest things similar to that of Multinomial Naive Bayes.

Maximum Entropy classifier: The Maximum entropy Classifier(known as conditional exponential classifier) converts labeled feature sets to vectors using encoding.The Max Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. The Max Entropy does not assume that the features are conditionally independent of each other. The Maximum Entropy is based on the Principle of Maximum Entropy and from all the models that fit our training data,we select the one which has the largest entropy.

Theoretical Background of Maximum Entropy:

Our target is to use the contextual information of the document (unigrams, bigrams, other characteristics like words within the text) in order to categorize it to a given class (positive/neutral/negative, etc). Following the standard bag-of-words framework that is commonly used in natural language processing and information retrieval, let w_1,...,w_m be the m words that can appear in a document. Then each document can be represented in a binary format of 1s and 0s that indicate whether a particular word w_i exists or not in the context of the document.Our target is to construct a stochastic model which accurately represents the behavior of the random process: that is to take as input the contextual information x of a document and produce the output value y.x and y are also known as the feature sets and labels of our document. The first step of constructing this model is to collect a large number of training data which consists of samples represented on the following format: (x_i,y_i) where the x_i includes the contextual information of the document (the array) and y_i its class. The second step is to summarize the training sample in terms of its empirical probability distribution:

p(x,y)=$\frac{\text{Number of times (x,y) occurs in the sample}}{N}$

where N is the size of the training dataset We will use the above empirical probability distribution in order to construct the statistical model of the random process which assigns texts to a particular class. The building blocks of our model will be the set of statistics that come from our training dataset i.e. the empirical probability distribution.

We introduce the following indicator function:

$$f_j(x, y) = \begin{cases} 1 & \text{if y = } c_i \text{ and x contains } w_k \\ 0 & \text{otherwise.} \end{cases}$$

We call the above indicator function as **"feature"**. This feature function returns 1 only when the class of a particular document is c_i and the document contains the word w_k. We express any statistic of the training dataset as the expected value of the appropriate binary-valued feature function f_j. Thus the expected value of feature f_j with respect to the empirical distribution is equal to:

$$p(f_j) = \sum_{x,y} p(x, y) f_j(x, y)$$

If each training sample (x,y) occurs once in training dataset then $p(x, y)$ is equal to 1/N.

Now we calculate the expected value of the feature f_j wrt the conditional emprircal probability distribution of class or label(y) given feature-set(x). To do so, we constrain the expected value that the model assigns to the expected value of the feature function f_j. The expected value of feature f_j with respect to the model $p(y|x)$ is equal to:

$$p(f_j) = \sum_{x,y} p(x)p(y|x)f_j(x, y)$$

where $p(x)$ is the empirical distribution of x in the training dataset and it is usually set equal to 1/N.

By constraining the expected value to be the equal to the empirical value and from the above equations, we have that:

$$\sum_{x,y} p(x)p(y|x)f_j(x, y) = \sum_{x,y} p(x, y)f_j(x, y)$$

This is known as the constrain equation and we have as many constrains as the number of j feature functions.

The above constrains can be satisfied by an infinite number of models. So in order to build our model, we need to select the best candidate based on a specific criterion. Now according to the principle of Maximum Entropy, we should select the model that is as close as possible to uniform. In order words, we should select the model p^* with Maximum Entropy:

$$p^* = \underset{p}{\operatorname{argmax}} \left(-p(f_j)\right) \text{ i.e}$$
$$p^* = \underset{p}{\operatorname{argmax}} \left(-\sum_{x,y} p(x)p(y|x)f_j(x,y)\right)$$

To solve the above optimization problem we introduce the Lagrangian multipliers, we focus on the unconstrained dual problem and we estimate the lamda free variables λ_1,...,λ_n with the Maximum Likelihood Estimation method,where λ_1,...,λ_n are also known as the weights.

It can be proven that if we find the λ_1,...,λ_n parameters which maximize the dual problem, the probability given a document x to be classified as y is equal to:

$$p^*(y|x) = \frac{\exp\left(\sum_i \ \lambda\ _i f_i(x,y)\right)}{\sum_i \ \exp\left(\sum_i \ \ \lambda_i \ f_i(x,y)\right)}$$

This is our desired probability of a particular word belonging to a particular class.

Advantages of MaxEnt algo over Naive Bayes Algo are that this algo makes no independence assumptions for its features unlike Naive Bayes.This means we can add bigrams and phrases to MaxEnt without worrying about feature overlapping.

Our Proposed Probabilistic method: Keeping the idea of Gaussian Naive Bayes on mind, we have thought of an algorithm, lets call it **SSR-Distribution Classification Algorithm**($\tilde{\alpha}$), where $\tilde{\alpha}$ is a vector. So, we first fit the actual smoothing density function to each class. So, say for class y, the density function is f_y. Now lets look at the word w. The word w has some scores of negative, positive etc which are the number of times the word w appears in the positive documents, negative documents etc respectively. Now, the main thing is that at the time of feature selection, we just take the important features. Now, most of the words will have a density very close to 0 and the number of words which has a high density will be very less compared to other. So, it is expected that $P(y|w)$ will be inversely proportional to some positive power of $f_y(w)$ and proportional to $P(y)$. So, our assumption of this model is

$$P(y|w) = \frac{1}{(f_y(w))^{\alpha_w}} \tag{10}$$

where, α_w is > 0 for each word w.

Now, when we get a new sentence, first of all, we try to find the synonyms or the actual words contained in that sentence which has an intersection with the selected features for class y. Then we proceed again considering all the features to be independent of each other. So,let say the new sentence got the common features $x_1, x_2, ..., x_k$ for the class y.

$$P(y|sentence) = P(y|x_1, x_2, .., x_k) \tag{11}$$
$$\propto \frac{P(y)}{\prod_{i=1}^{k}(f_y(x_i))^{\alpha_{x_i}})} \tag{12}$$

Now, $\alpha_1, \alpha_2, ..., \alpha_k$ are clearly dependent on the given dataset and hence for a given dataset, we are keeping this as future work to find theoretically the optimum α_i's for a given dataset. But for a first work, we can assume all the α_i's to be 1, that is first we can work with SSR-Distribution Classification Algorithm(1), where 1 is a vector with all elements 1.

So, after fixing an optimum α, we find for a sentence w, the maximum probability is for which class and then follow the similar process as of multinomial naive bayes. So our the final class to be assigned will be:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \frac{P(y)}{\prod_{i=1}^{k}\left(f_y(x_i)\right)^{\hat{\alpha_{x_i}}}} \tag{13}$$

where $\hat{\alpha}$ is the following:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}}\, P(y|sentence) \tag{14}$$

So, basically first finding the optimum alpha over classes we have to find the optimum class for a given new sentence. So, this is to find the supremum of the supremums. But, for now, we will proceed keeping $\hat{\alpha}$ to be 1, a random estimator and will implement how it works.

Then, we can use the idea of reinforcement learning to use the test data also. So, we will test the given test sentences one by one. So, say at the first sentence, we predicted it to be positive and got some new words. Then, we will repeat the whole process again starting from feature selection and then classification, and will repeat the process again and again. Now, the question is which sentence to choose first? For that, we can use the cross-validation and can take that one for which the algorithm gives the maximum accuracy at the end.

- **Some Statistical or ML Algorithm based classification**: Machine learning algorithms like SVM (Support Vector Machine) or Random Forest.

  SVM (Support Vector Machine): Support Vector Machine (SVM) is a supervised learning model which is used for classification. The goal of this algorithm is to find the best decision boundary (Hyperplane) in an n-dimensional (dimension of the hyperplane depends upon the no. of features)space that can distinctly classify the data points. Now suppose we have a feature space x. These belong to either of the classes $w = w_1, w_2, ..., w_n$. The goal is to define the hyperplane

  $$w^T x + w_0 = 0$$

  that classifies all the training vectors properly. The algorithm tries to find a hyperplane which has maximum

margin with the closest points of all the classes, i.e. find w^* such that

$$w^* = arg_w max[min_n d(x_n)],$$

where

$$d_H(x_n) = \frac{w^T x_n + w_0}{||w||_2}$$

Now in our experimental analysis, we make a new vector set corresponding to each sentence on the feature set, which contains numbers denoting each word is contained how many times. We also number the classes **Positive, negative, Irrelevant,** and **Neutral** as 1,2,3 and 4. Now we train our SVM Model according to which vector from the new vector set is classified to which class. after that we create the confusion matrix and check the accuracy.

Random Forest: Random Forest or Random Decision is a Supervised Machine Learning Algorithm used for classification which is done by constructing a multitude of decision trees at the time of training the model. The output of the random forest will be the class selected by most of the trees. Now **Decision Tree** is also a supervised machine learning algorithm used for classification, which is basically used in Random Forest. The steps of the Algorithm of the are as follows:

- Begin the tree with the root node containing the feature set (say S).
- Using Attribute Selection Measure (ASM) we have to identify the best attribute in the feature set.
- Divide the set S into subsets containing possible values for the best attribute.
- Repeat steps 2 and 3.

Now to select the best attribute, we should use **Gini Index** which is defined as $P_j = 1 - \sum_{i=1}^{n} p_i$, where p_i is the probability of each class. Now we have to calculate weighted gini index, which is the total gini index of the split, which is defined as $\sum_{j=1}^{m} w_j P_j$, where w_j is the weight associated with each split. Now we will choose that feature which will have low gini index, as low gini index denotes low impurity.

- **Information Theory and Probability Based Approaches**: We can think of different approaches based on mutual information, entropy etc.

Based on Mutual Information and Bayesian Conditional Probability: For random variables X,Y the mutual information formula is as described in (1). But now let us define the conditional mutual information. For random variables X, Y, Z the mutual information between $X$and$Y$ conditioning on Z is defined as follows:

$$I(X;Y|Z) = \sum_{x,y,z} p(x,y,z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)} \quad (15)$$

And the following is the chain rule of mutual information for finding mutual information of $(X_1, X_2, ..., X_n)$ and Y:

$$I((X_1, X_2, ..., X_n); Y) = \sum_{i=1}^{n} I(X_i; Y|X_1, .., X_{i-1}) \quad (16)$$

Now, the assumption is **all the features are independent of each other and each class is independent of a word given another few words**. Now, when we get a new sentence for testing, let say the features(words) common are $X_1, X_2, .., X_n$. So, say we have to find the mutual information of the sentence (say S) and the positive class(say P). So, the formula will be:

$$I(S;P) = I((X_1, X_2, .., X_n); P) \quad (17)$$

$$= \sum_{i=1}^{n} I(X_i; P|X_1, X_2, ..., X_{i-1}) \quad (18)$$

Now, as each class is independent of a word given another few words, hence $I(X_i; P|X_1, X_2, .., X_{i-1})$ is same as $I(X_i; P)$. So, from (18), it follows that

$$I(S;P) = \sum_{i=1}^{n} I(X_i; P) \quad (19)$$

Now, mutual information quantifies the "amount of information" obtained about one random variable by observing the other random variable. So, greater mutual information implies that one random variable tells a lot about the other. But only mutual information can't tell us whether a particular sentence belongs to some class or not. It can only tell us that, given some sentence is in some class, how confidence we are about that, or how much information we have about that. So, we are proposing an algorithm combining the mutual information and gaussian/multinomial naive bayes. For handling with these(two different units), let us define a new relative information measure first, which is nothing but just for removing the units of the mutual information. So, the relative information measure is

$$I^*(X_i; Y|data) = \frac{I(X_i;Y)}{\sum_{i=1}^{n} I(X_i, Y)} \quad (20)$$

where $X_i's$ represent different words and all the others have their usual meaning. Now, let's define a new measure of characterization for this purpose.

$$C(X_i;Y) = \alpha I^*(X_i;Y) + (1-\alpha)P(X_i|Y) \quad (21)$$

where $I^*(X_i;Y)$ is as defined in (20) and $P(X_i|Y)$ is in (7) or (6) depending on gaussian or mutinomial. And, for a new sentence S having features $X_1, X_2, .., X_n$ in common, the new characterization measure will be

$$C(S;Y) = \alpha I^*(S;Y) + (1-\alpha)P(S|Y) \quad (22)$$

$$= \alpha I^*(X_1, X_2, ..., X_n; Y) + (1-\alpha)P(X_1, X_2, ..., X_n|Y) \quad (23)$$

Now, we can use the formula of joint mutual information and the joint probability **assuming all features to be independent**. α have to be decided by checking the accuracy values, the one giving the maximum accuracy is to be taken. The next process of classifying is similar to that of before.

Bibliography:

1. Twitter Sentiment Analysis
2. Sentiment analysis algorithms and applications: A survey