Analysis of Tweets related to M.Tech Fee Hike in IITs using Text-Mining

*

Abhinav Antani
M.Tech Computer Science and Engineering
NIT-Surathkal,Karnataka
Surathkal,Karnataka
antaniabhinav@gmail.com

***Abstract*—Data is the currency of this decade. And it is increasing day by day exponentially. This data comes in many form such as numerical , image , video , texts , web pages etc. Now as this data can give us some prestigious and unknown information it is important to use it wisely and efficiently. Most of the data present in the current Web of internet is in text form. They are Tweets in tweeter, blogs in various topics by bloggers, movie reviews in ImDB, description in webpages and what not. To make the best out of this text data we need some efficient way of using and handling this data. Text Mining is one such field. This field is same like Data-Mining but here we deal only with the text based input data. In the field of Data-Mining various Techniques are established to do mining and extract unknown information from the data and gain knowledge. These Techniques are nothing but association rule mining, classification, clustering, regression etc. Now algorithms in such kind of techniques are traditionally used to access structured data as their input. Now here our data is in text form so we cannot use these Data-Mining algorithms directly to our input. Here in this work Technique by which we can mine text data is explained.**

I. Introduction

If we classify data then we can have mainly three types of data: Structured Data, Semi Structured Data and Unstructured data. As the name suggests Structured Data has some sort of structure; means it has structure and schema for example Relational Databases. Semi structured data has structure but it does not have any kind of schema example of such kind of data is XML. Unstructured data has neither structure nor schema. Text, image, video, webpages etc comes under this type of data. A data-Mining algorithm mostly deals with numerical or structural data as its input. So when it comes to text based data, those algorithms cannot be used directly for the text input.

Text-Mining is a type of data-mining. But it deals with text based data. It is discovery of knowledge. It extracts unknown and implicit information from the various sources of text based data. Text Mining is different than Data Mining here in Text Mining before applying any data mining technique or data modeling technique we have to convert this text based data into some structured or numerical formats. After doing so we can apply that to the traditional data mining algorithms. For converting text based data into some structured or numerical format we need to apply some linguistic and statistical techniques to it. Text-Mining has various applications in various fields like: spam filtering , context based advertisement , fraud detection , cyber security , customer care services , smart devices , social media data filtering, news filtering etc.

Usage of social media such as Tweeter, Facebook, Quora have increased exponentially. People from all over the world use them and they provide their opinion on every topic which is going on in the world. This whole data is also in the form of text. Using this data and text mining techniques we can do sentiment analysis of people's opinion in the particular topic. Here in this work I have analyzed one such topic with the use of the model I made using the concepts of Text-Mining. Topic which I chose to do Analysis using text mining is M.Tech course fee hike across Indian Institute of Technology (IITs).

Indian Institute of Technology (IITs) are government founded universities which provides various courses such as B.Tech, M.Tech PhD etc. To get admission in M.Tech course one has to give entrance test named as G.A.T.E (Graduate Aptitude Test for Engineering). After clearing this entrance exam G.A.T.E one makes oneself eligible for the admission is this set of university. And students who have cleared G.A.T.E and are studying in any of these colleges are given stipend from the Government of India. But recently on 28th of September 2019 Ministry of Human Resource Development announced that all the new students from year 2020 has to face hike in tuition fee upto 900 percent and will stop the stipend of some students as well. This statement made people on tweeter crazy and they started tweeting using the hashtag MTechFeeHike. This generated plenty of data under this particular hashtag.

Here in this work first I have made one model using the concepts of Text-Mining which detects emotion from the text and does sentiment analysis. And using that model I have analyzed sentiments of people who have tweeted using hashtag MTechFeeHike.

II. Literature Review

Text mining is a buzz field right because of its various applications in the field of sentiment analysis. Many people have worked on this topic and I have referred to some of those works.
In the simple technique, Information retrieval uses term based approach. Term is nothing but words in simple language. So it uses these terms to add and update discovered patterns from the text documents effectively. This technique has some problem. It does not work when one word has two different meaning or two words have same meaning. These problems are called polysemy and synonymy. Synonymy means two different words have same meaning whereas polysemy means same word has two different meaning. This might affect while doing the sentiment analysis. For example 'get': get means procure (I will get the food) and become as well (he got depressed) here one sentence is neutral while other depicts sadness. Here both the sentence will update get's count of some emotion but it might not be true. Same can happen for the synonymy. To overcome this problem a pattern based technique can be used where set of terms will decide the patterns and not only the terms. And in this approach these patterns are used to extract information from the text document and add or update the discovered patterns. There might be problem in such kind of approach as well. Problem of low frequency count or misinterpretation might arise. These problems arise when minimum support count is decreased and patterns with the less count are discovered. They might not affect the sentiment detected but the pattern based approach thinks so. Then one can mostly focus on repetitive patterns and can search for the long patterns. In these techniques they first search for the pattern/term and according to the association rule mining they search frequently and important patterns for the discovery. They use different association rule mining algorithms to carry out such work.

With the use of any of these techniques many have made a model which can do Text Mininig. In many real life problem many have used Text-Mining. Model which was mde using text mining was used to detect spam, it would classify emails in two types whether they are spam or not. Then Text-Mining model is also used in news filtering it will mine the text and then cluster them into different genres. Other application where Text mining was used is political opinion gathering. Someone collected all the statements given by Donald Trump and made a classification of statements like good, bad or neutral which helped others to predict the results of the election campaign. To remove plagiarism text mining technique is used. Like this in sentiment analysis as well Text mining is used and using this sentiment analysis movie review categorization was made. In which according to movie reviews given by users on ImDB movies and TV shows were rated.

In this work I have done sentiment analysis on Text Mining and using the model which was made by this, I have done analysis of reactions gathered from tweeter on the topic of Fee hike in IITs by Government of India.

III. Problem Statement:

People around the world post their opinions, thoughts and activities of their day to day life on social media such as facebook, tweeter, blogs etc. People nowadays posts rigorously on any topic which is going around the world whether it be politics, sports, business or any other subject. If we make people post in a certain way then we can generate plenty of data in any topic. Luckily, people do follow some sort of pattern to post about; it is nothing but hashtags. Now if we want to fetch information from some social media platform on any topic then we can just fetch all the posts done using some hashtag. That will generate adequately large amount of data. And later after collecting that particular data we can perform analysis on it. Now majority of this data is in the form of text. Which technically speaking is unstructured type of data. As text does not follow or have any particular type of framework.

Conventionally Data Mining Techniques uses Structured or numerical type of data as their input to the algorithm. Now data which we generated for the analysis purpose is in the form of text which is neither structured nor numerical. So How to perform Text Mining on them? And later How to analyze the data? For this we have to find some technique using which we can perform the text into some sort of numerical representation. After converting it to numerical form we can pass this data in the data mining algorithms and can mine them; and later can perform analysis.

As a real life scenario and application of Text Mining for doing sentiment analysis I have chosen the topic MTech fee hike in IITs. Now to do analysis on any topic we need data. We can get it from tweeter by fetching the tweets which includes hash tag MtechFeehike. Now I am able to fetch around 4 to 5 thousands of tweets related to this topic. If we just have this data then it is of no use. It won't say anything about the topic. To gain some knowledge we need to extract unknown information from these tweets. How to extract unknown information from this data?

IV. Proposed Work:

As discussed earlier in above sections; here the technique by which we can transform our text based data into some mathematical or numerical data is discussed, So that we can supply this mathematical representation of our text data as

input to the conventional data mining algorithm. And can mine huge text based data and extract unknown information from it.
If we try to represent text mining process in some block diagram we will end up having some diagram like this

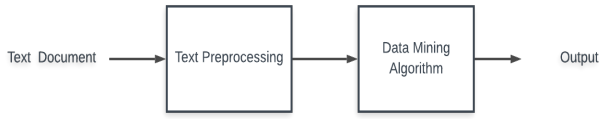

Fig. 1. Block Diagram of Text Mining Process.

Before applying any data mining technique using any algorithm for input data. This data needs to be preprocessed. If we do not preprocess the data and directly give it to the data mining algorithm then results may not be the one desired. This happens because data might have some inconsistency and noise. To remove this data inconsistency and noise we have to do data preprocessing. In text mining our data is nothing but a text document we need to do text preprocessing.
If we expand text preprocessing in major parts and represent it in a block diagram kind of structure then we will have something like this.

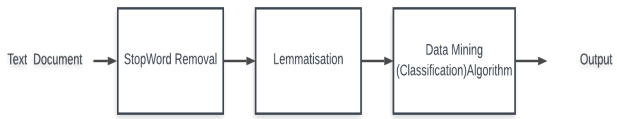

Fig. 2. Extended Block Diagram of Text Mining Process.

Here Text preprocessing is divided mainly into two parts that are 1) Stop Word Removal and other is 2) Lemmatization. After going through this process our text based data will now be ready to be given as input to the data mining algorithms. Here in this work we will use classification technique under data mining to mine the data and extract unknown information from it efficiently. After data mining algorithm we will have our discovered patterns and can do sentiment analysis on it. Main aim of doing text preprocessing is to find key terms and patterns from the text document and by using these key terms and pattern enhance the relevancy between some particular terms and the whole document. By doing text preprocessing we can also discard the terms that does not contribute to distinguish the document.
To start with our text preprocessing first we need to remove the stop words. This step is called stop word removal.

A. Stop Word Removal

Sentence formation in any language is combination of words put in some certain manner. Now this words can be noun, pronoun, prepositions, verb, adjective, article etc. From these types of words most commonly articles, pronoun and prepositions does not provide any meaning to the document. These kinds of words are treated as stop words. So, putting it in a nut-shell if a word does not provide any meaning to the document then such kind of word is called stop word. Now as we are doing sentiment analysis importance of such words in our document is very less. So if we remove them and do not provide them to the data mining algorithm as input then it will be beneficial for us.
For example the sentence “I will not be able to have fun this weekend as I have to submit the report” is input to the stop word removal. After removal of stop words this sentence will be converted to something like this: “not fun weekend submit report”. As the second statement is sufficient enough to tell what is the sentiment behind that. It would have been unnecessary to supply whole sentence as the input and waste the processing time.
After doing this stop word removal the processed text is given to second preprocessing technique which is lemmatization.

B. Lemmatization

Lemmatization is also known as word stemming. Stemming or Lemmatization is the technique to convert every word to its root form. What is root form of the word? Every word in English language has a root word. All the words which has same root convey same or relatively close meaning. And for sentiment analysis we only care about the meaning and not what was the tense of the word? we can convert them to their root form. So if we convert every word to its root form then we can reduce the size of our dictionary and increase the performance of the algorithm and decrease the processing time.

Goal of lemmatization or stemming is to reduce words and sometimes derivatively related form of word to its base form. Lemmatization usually does vocab and morphological study of the words and then converts them to the base or root or dictionary form of word. As the meaning of the word will not change and it will reduce the number of words to be saved in the dictionary it is beneficial to use lemmatization before applying our text based input to the data mining algorithms.

Consider two sentences “I saw organized room of a student” and “see organization of this room by the student”. After stop word removal both the sentences will become “saw organized room student” and “see organization room student”. Now if lemmatization is not used then we would have to save “saw see organization organized room student” in our dictionary. But if Lemmatization is used then saw will be converted to its base form see and organized and organization both will be converted to organize. And our entries which are done in the dictionary will be “see organize room student”. So, from 6 entries in dictionary after lemmatization it reduced to 4 entries. This is how lemmatization is beneficial for the data mining algorithms.

After doing preprocessing of the data we now have noise free and consistent data. But the main question still remains intact, that How to convert this text based data in some mathematical format? If we don't convert it then it is of no use as data mining algorithms use mathematical or structural data as input. Here in this work I have used two techniques to give our text based data some sort of mathematical structure. These techniques are TF/IDF and Count vectorization. Out of these two techniques only one of them is used in the final model making. We will see the performance comparison of these techniques with each other later on this work.

C. TF-IDF(Term Frequency – Inverse Document Frequency)

Till now we have a sequence of words which are not having any stop words and they are in their base form. We want to give them some mathematical representation. This technique named as TF-IDF will give these words a statistical measure of how important this word is to the document. So putting it in a nutshell TF-IDF will give some score to each word and whichever word has higher score than the other is more important. Now the question is how does this thing work?

If we split TF-IDF then there are two parts 1) TF (Term Frequency) and 2) (IDF) Inverse Document Frequency.

Term Frequency: Frequency is a count. It counts the presence a particular term (word) t in the document d. If we represent term frequency of term (word) t in a document d as 'tf(t,d)' then,

$$tf\left(t,d\right) = N\left(t,d\right) \tag{1}$$

Where, N(t,d) represents presence of a particular term(word) in the document. From this we can say that if a term will be present more number of times in a document then it will be important than the other. Now that is logically correct. These values are represented in a vector. This vector has every unique term in the document and the tf(t,d) count in it. Now if one document is larger than the other document in the set of documents, then the larger document will have the larger term frequency and the smaller document will have the smaller term frequency. Now in this case, picking the one with higher term frequency will not be fair. So to make this thing fair we can normalize this term frequency by introducing total number of terms in the document. If we denote total number of terms in a document by 'D' then the formula for Normalized term frequency will be.

$$tf\left(t,d\right) = N\left(t,d\right) \div D \tag{2}$$

Inverse Document Frequency: This measure typically tells us how important a term is in whole set of documents. The main purpose of doing this is to not ignore the terms with lower tf count to the ones with the higher tf count because many times only a single word can change the meaning of the whole sentence. In that case if only tf count is used as the statistical measure then term with lower tf count will not create any impact in front of term with higher tf count. So, basically we want to scale down the importance of high tf count and scale up the importance of low tf counts. Logarithms can be very useful for achieving this sort of behavior.

Now term frequency is only associated with a document. But Inverse Document Frequency is associated to whole corpus (set of documents). Idf measure is nothing but number of document in the corpus divided by the document frequency of a term. Now we want to scale up the importance of low tf count terms and scale down importance of high tf count but this seems harsh so we use logarithms to avoid that. So, final formula to compute idf can be given as follows.

$$Idf\left(t\right) = N \div df\left(t\right) \tag{3}$$

Where, idf(t) is inverse document frequency of term t, N is total number of documents is the corpus (set of documents) and df(t) represents number of documents where term t has occurred.

Now we have the term frequency (tf) and inverse document frequency (idf) for a particular term so now we will compute the final tf-idf count which will be used for the ranking purpose of term. Higher the score important the term is.

$$tf - Idf\left(t\right) = Idf\left(t\right) \times tf\left(t,d\right) \tag{4}$$

D. Count Vectorization

We have same input to count vectorizer which we had in TF-IDF. Both of them models text based data into some sort of mathematical representation. TF-IDF and count vectorizers are used independently.

We have a text based input to this count vectorizer, basically what count vectorizer will do is it will count the number of occurrences of a particular word in a document and will store it in a vector type of form. Now count vectorizer count number of occurrences of particular word in a document. To do this it needs to scan entire document and then update the count of a word if it exists in the vector or add entry of some word in the vector. Now if the document is too large then it might take more time to compute it. To solve this problem count vectorizer has one parameter named as max-df. Value of this max-df parameter says that how many words to count. By this we can restrict out count vectorizer and can have improvement in the time for computation.

So far this count vectorizer is counting occurrence of each word. But many times what happens is some words individually do not carry any meaning to the statement but

when composite it with other words it makes sense. For example “great effort”, “missed the catch”, “cat is out of the hat” etc. So if these words taken individually will not make impact in the sentence. But if we take them to gather then it makes sense. So taking this words independently we might lose the meaning of the sentence. We need to count them in the set. To count them in the set count vectorizer has one parameter called as ngram-range. By setting the value of this ngram-range to some value k count vectorizer will form the set of k words as well.

After applying count vectorization technique we will be having one vector which will have numerical data about our text based input. That is what we wanted at the first place. Now this can be given as input to the data mining algorithms and later we can mine this data to extract unknown information from it.

E. Data Mining Algorithm

Now we have our text based data represented in the form of mathematical data. Thus we can use some data mining algorithm on it. Before selecting algorithm we need to keep application in our mind. Here we want to do sentiment analysis of text based data and then apply the model made to tweeter data collected for the hash tag MtechFeeHike. To achieve this I have used data classification.

Classification is the technique where supervised learning is done to train the model. Supervised learning is type of learning where along with the attributes class label is also provided. And from this data we can train the model and after training is done we can test the model with any data. And later we can check whether predicted results and real results are same or not. To achieve this we first need data with label. So I have used a data set with around 40000 tweets which are in general topic. They also have class label associated with it. Here in our case class label is nothing but the emotion depicted by the tweet. Here in this data set there are 13 human emotions, out of this 13 we need just 2 emotions which are happiness and sadness. So after taking out these 2 emotions from this dataset now we have data set having around 10000 tweets. Now these 10000 tweets data have author of the tweet, likes, number of re-tweets, hashtags used and content of the tweet. Now to predict emotion from the tweet we don’t need information like author of the tweet, likes, number of re-tweets, hashtags; only the tweet content is enough. So we will drop all these columns from the data set and will keep only content of the tweets to supply it to classification algorithm for training purpose.

Here in this work I have used 4 different classification algorithms: 1) Naïve Bayesian 2) Linear SVM 3) Logistic regression 4) Random Forest classifier In the analysis and results section I have compared the performance of these algorithms.

Naïve Bayesian: This is a very basic algorithm in classification technique. This algorithm uses the concept of Bayes Theorem. In Naïve Bayesian classification algorithm we assume that all the features are independent. And because of this calculating the probability becomes easy. Here basically from training data we calculate the P(Y—X) where Y is the class label and X is the feature vector. Now after calculating P(Y—X) for all the class labels we compare these probabilities. Class label for which P(Y—X) is maximum that class label is predicted to be the class label of the testing vector X.

Linear SVM: Full form of linear SVM is Support Vector Machines. It is also one classification technique. In this technique labeled data is given as input and it finds an optimal hyper plane that separates the data points from each other. Here in our case there will be only two data label: Happiness and Sadness so algorithm will separate this data using some optimal hyper plane in two parts. Then points which are on one side of the hyper plane will be labeled as Happiness and one on the other will be labeled as Sadness. Now when any new testing or new data will come it will check on which part of the hyper plane does the data point belongs. And it then predicts the emotion accordingly.

Logistic Regression: Regression analysis is a type of predictive modeling where it finds relation between the target class label and the given set of input vector. When target class label is categorical which in our case is we can use logistic regression. It finds the probability of data and predicts which class label it will belong to. Here as we just want to classify whether this sentence depicts happiness or sadness we can easily and effectively use Logistic regression. This will train itself from the training data and then for testing or new data it will directly calculate the probability and will predict the class label where our text belongs.

Random Forest Classifier: Random Forest classifier is an Ensemble algorithm. Ensemble algorithms are the one which combines some algorithm to find the outcome. Random Forest algorithm basically generated a set of decision trees from the subset of the training data. And later it collects vote from this decision trees and then at the end which ever class label has the highest votes that class label is given to the input. For the decision tree making it uses either Information Gain or Gini Index to find which attribute will take root position at every level. This method also works fine with our case as we need to classify whether given text depicts happiness or sadness.

Now using these four algorithms independently the model was trained. And this model will be used to do the analysis of the tweets related to hash tag MTechFeeHike.

V. Analysis and Results

So far I have used two techniques to convert our text based input data into some mathematical format and they are 1) TF-IDF and 2) Count Vectorization. And after modeling text based data in mathematical format I have used 4 classification algorithms to train the model for sentiment analysis independently.

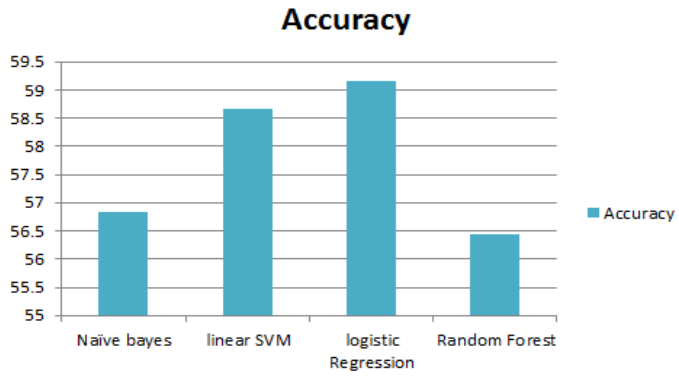

Fig. 3. Accuracy analysis when TF-IDF is used

Above graph shows the data when TF-IDF was used as modeling to convert text based data in to some mathematical format. And from the data you can see different classifier gives different results. As you can see from the graph if TF-IDF is used as modeling technique then the highest accuracy of the model which I am getting is near to 59 percent which is of logistic Regression.

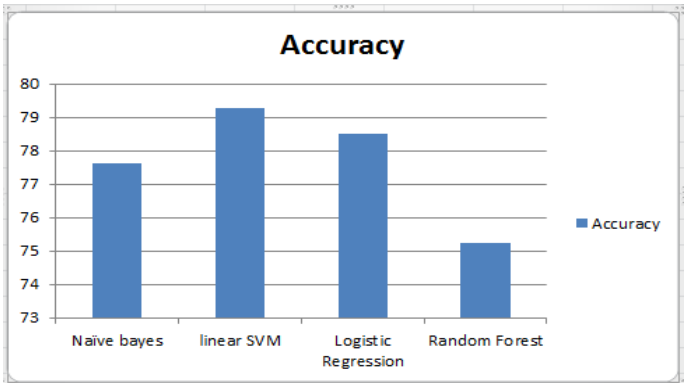

Fig. 4. Accuracy analysis when Count Vectors are used

Above graph shows the stats when Count vectorization was used as the modeling technique. As you can see there is more than 20 percent of improvement from the last version and if Linear SVM algorithm is used as the classification along with count vectorization as the modeling technique then accuracy of the model is nearly 79 percent. Now this Linear SVM along with count vectorization as the modeling technique is used for the analysis of tweets related to hashtag MTechFeeHike.

Analysis of tweets related to M.Tech Fee Hike: To analyze the data first we need data. So to collect the data related to M.Tech Fee Hike I used tweeter's api. Now tweets related to M.Tech Fee hike were having hashtags like mtechfeehike, MTechFeeHike, Mtechfeehike and iitfeehike. The decision to hike the fee in iits was disclosed on 28th of September 2019. And people started tweeting about this topic since the announcement of this decision. Now after fetching the data from the tweeter I was able to generate around 4000 tweets. These tweets contain the information such as author of the tweet, number of likes , number of re-tweets , content , mentions , hashtags used , tweet id. Out of these information content is mainly used for doing sentiment analysis. This content of every tweet is sent to the model and its sentiment is predicted.

Now if a human being wants to gather and analyze these tweets using their brain power then it will take a lot of time. At this time the model which was made to do sentiment analysis can be used. We can supply these 4000 tweets as an input to the model which uses linear svm and count vectorization for the modeling of text data to mathematical data. And we can see the results.

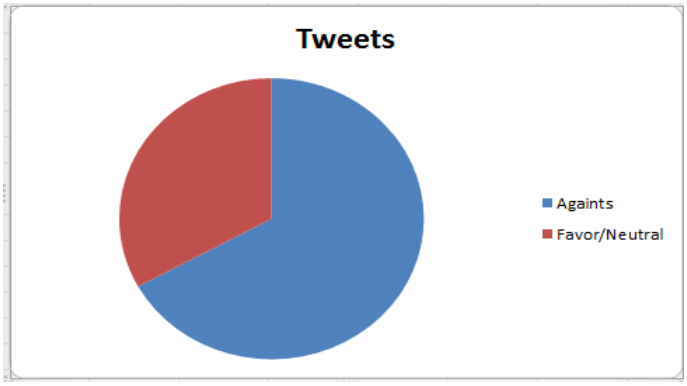

Fig. 5. Analysis of Sentiment of Tweets

Above graph shows the graphical representation of how were the emotions analyzed from the tweets taken related to MTech Fee Hike. In this above pie chart blue region states the percentage of people which were predicted to be sad by the model and the red region were predicted to be happy. In the model if there is nothing which can tell about happiness or sadness then it predicts the sentiment of such texts as happy by default. So here Blue region shows percentage of people who were against the decision and red region shows the percentage of people who were either happy or neutral for the situation.

Out of these tweets many have mentioned top authorities like Prime minister narendra modi , Prime Minister Office , Ministry of Human resource and development and even BJP's official tweeter account. People wanted to show their rage towards the decision by taking this matter to these account directly in tweeter. After plotting graph for these kind of tweets

below graph was obtained.

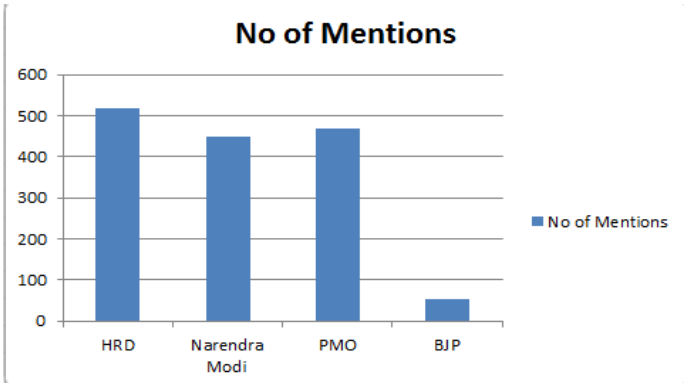

Fig. 6. No of Mentions to official accounts

From the graph it can be seen that around 500 , 400 and 450 tweets were mentioning Ministry of Human resource and development , Prime minister narendra modi , and prime minister's office. But in turn there was not a single tweet from these official account on this matter.

After this people started using the hashtag engineeratjantar-manter stating that they will protest against the government of India because of this decision. And plotting the graph for such tweets below results came.

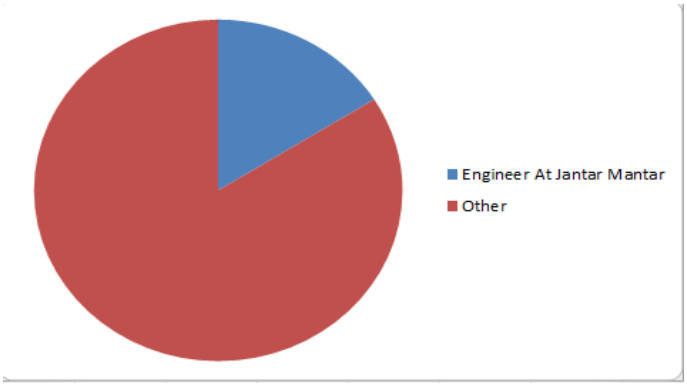

Fig. 7. Posts about protesting at Jantar Mantar

From the graph it can be seen that out of around 4000 tweets around 700 tweets tried to start a protest against the government of India because of this decision. They all were going to gather around Jantar Mantar and were about to do the protest.

VI. Conclusion

Text mining is discovery. It is used to discover unknown information from the text based. We can not use data mining algorithms directly by taking text based data as input. First we need to model them using some technique called as TF-IDF or count vectorization into some mathematical format. Here for the data set I have used count vectorization gives best accuracy of nearly 79 to 80 percent when used with the linear svm algorithm. But it does not mean that every time count vectorization will work better or linear svm will work better. In this data set they were proven to be best.

Text mining has many application from many of them sentiment analysis was implemented. After doing sentiment analysis we can use this model to analyze any topic. Here people's state of mind for the decision on MTech Fee Hike was analyzed using tweeter. And after analyzing it occurs that more than 70 percent of the people who tweeted about it are unhappy and against the decision taken. And many people were about to protest at Jantar Mantar ground against this decision.

References

[1] M.Sukanya , S.Biruntha, "Techniques on Text Mining".
[2] Octavian Rusu, Ionela Halcu, Oana Grigoriu, Giorgian Neculoiu, Virginia Sandulescu, Mariana Marinescu, Viorel Marinescu, "Converting Unstructured and Semi structured data into knowledge"
[3] Kundan A. Dhande, Jayant S. Umale, Parag A. Kulkarni "Fine particles, thin films and exchange anisotropy,3. "Context based text document sharing system using association rule mining"
[4] V.Ashwini,S.K.Lavanya "Pattern Discovery for Text Mining"
[5] Shadi Shaheen ,Wassim El-Hajj, Hazem Hajj ,Shady Elbassuoni "Emotion Detection from Text based Automatically generated rules"