ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

A semantic Vector Space Model approach for sentiment analysis

Vijay Dixit[1], Anil Saroliya[2]

M.Tech (CSE), Computer Science and Engineering, Amity University, Jaipur, India[1]

Coordinator (CSE), Computer Science and Engineering, Amity University, Jaipur, India[2]

Abstract: A response to growing availability of formal, informal, opinionated texts like film review, product review etc., an area of Sentiment Analysis has begun which raised the question that "What people think about a particular topic?". This paper present a semantic VSM (vector space model) which capture sentiment and semantic similarities among words which we are using on the micro blogging sites. Semantic vector space model-based approaches used for a large amount of information could be obtained by analysis of generated word text by the humane on social networking sites. The purpose of this research is the construction and estimation of algorithms for the analysis and the classification of large amount of humane generated text data, will focus on sentiment analysis on twitter or similar social community environments.

Keywords: Machine learning concept, Classification, VSM Clustering, WEKA2.6.

I. INTRODUCTION

In recent years, the extensive use of twitter is a popular social networking site focus on brief, update type information sharing. These types of action are referred to as micro blogging. Twitter allows users to post message up to 140 characters in length called tweets. A large amount of useful information can be obtained through analysis of human generated text. This information can be applied and used to perform a variety of useful task. But the word representation is analytical component of natural language processing system. It is common to represent words as basic in a vocabulary, but fails to find the bloated relation structure of the dictionary. Semantic Vector space model does more much better in this regards. So, Vector based models are used. They calculate the continuous similarities between words as distance or angle between the word vectors in very high dimensional space. The general approach of VSM has proven to be useful in tasks such as word sense disambiguation, named entity recognition, part of speech tagging (POS) and document retrieval, thing of our use. It uses term frequency (TF) and inverse document frequency (IDF) weighting to transform the values in a VSM, they often increases the performance of retrieval and categorization systems [1,2].

If let say, we are having a large collection of documents, and hence a large number of document vectors are present, it is very convenient to organize those vectors into a matrix form. The row vectors of the matrix correspond to terms usually (terms are words here, but we can discuss some other possibilities as well) and the column vectors correspond to documents (web pages, for example). This kind of matrix is known as a term document matrix. After the pre-processing of data like tokenization and stop word removal and finally generation of hash-bag, this will be processed into another method which will create Vector space model (VSM) which is word- document frequency count, resulting in a very large sparse matrix. Here we will made words as column and document as frequency. Data set is read again to make a VSM. Since the problem of twitter data is its slang language, so to make our VSM intelligent we induced semantics in it.

A semantic file is being maintained, which is a kind of mapping of some common slang words into their original counter parts. So, before inserting the data into the hash-bag, comparison is being made with this semantic pattern file which induces the semantic mean to data. Here we will check the occurrences of slang words, if a slang word is found we will check the semantic file and now if a particular slang word like 'grt' is found and somewhere in the document if 'great' occurs, they both will be treated as one only. Now, hash-bag is being processed with frequencies of the word, which we have placed in the hash-bags and thus creation of a new vector space model known as Semantic VSM takes place, which is basically a sparse matrix with lots of zeros.

Now this huge Semantic VSM will be put under test by various machine learning algorithms.

II. RELATED WORK

There exists a lot of substantial research on the subject of sentiment analysis. Although there have been some previous attempts to study the topic of sentiment analysis, most active research on the area came up with the explosion of user – generated content in social media, discussion forums, blogs and reviews. Since, most sentiment analysis studies mainly depend upon machine learning approaches the amount of user generated content provided unlimited data for training. The research on sentiment analysis so far has mainly focussed on two things:

1-Identification of whether a given textual entity is subjective or objective.

2-And identifying polarity of those subjective texts.

III. GERENAL OVERVIEW

Previous work on text- based sentiment analysis followed two main approaches:
The first approach has the assumption that semantic orientation of a document can be considered as an averaged sum of the semantic orientation of its words and phrases altogether. The pioneer work is the point wise mutual information approach pro-posed in a paper by Turney [1,2]. Also work such as [3, 4, and 5] is very good examples of this lexical based approach.
The second approach [4, 5] mainly addressed the problem of a text classification task where classifiers were built using one of the machine learning methods and trained on a dataset using features such as unigrams, bigrams, part of speech (POS) tags, etc. The vast majority of work done in sentiment analysis majorly focus in the domains of movie reviews, product reviews and blogs. Although most of the work [5, 6] achieved relatively very high sentiment classification accuracies, but they suffered from the domain dependence problem where performance often drops precipitously by applying the same classifiers on the other domains of interest. Other work also tried to overcome this problem by either building a hybrid classifier [7], or focusing on domain-independent features [8].
In answer to above problems works in [9,10] have addressed the domain- independence problem by building a weakly supervised classifiers where supervision comes from word polarity priors rather than using the labelled documents . They were able to extract the latent topics in a document with its associated sentiments. Apart from sentiment analysis at the document level there has also been some work on detecting sentiment at the phrase or sentence level[11].

IV. PROPOSED WORK

4.1 Data collection

The first work in this thesis is to collect huge amount of data. Collection of data although is not a simple task as it may seem to anyone at first thought. For this we have to make certain assumptions and decisions. Basically we have to collect two different datasets: test data, training data.

4.1.1 Twitter API

As seen in the earlier chapter, we talk of Tweet Archivist, It is a service that lets users to search Twitter for Tweets by sender, recipient, object of reference, or contents. Users may then create an archive of data corpus based on that search which they can analyse, export, and share. Users may choose to keep the search private or share the results with friends, colleagues, or the world. Users may create a maximum of the active archives. The tweet Archivist API enables users to visualize the data collected in their archives [11]. Available visualizations include volume over time, top users, Tweets vs. Retweets, top words, top URLs and source of tweets. So for data collection Tweet Archivist API was used.
The data is collected in English language only; we restricted the use of any other language.

4.1.2 Test Data

4.1.2.1 Collection

One of the objectives of this thesis is to analyse the sentiment of Twitter messages posted in reaction to movie reviews, so we have to collect tweets about movies only. However, it is not a simple task.
A corpus of around 10000 tweets about movie reviews was sampled from the test data and we manually examined and annotated that data as negative or positive because the data which we collected was unlabelled, so we formed two classes as pos, neg. Out of 10000 posts that have been annotated, 31 posts here needed context to understand and determine their sentiments. Thus 97% of our test data did not require context to determine their sentiment. Many a times it a time came when it was difficult to determine the sentiment, as they seemed both negative and positive.

4.1.2.2 Training Data

We have chosen only subjective data for training, no objective was collected here. Subjective data involves positive and/or negative sentiment only it does not consists of any neutral tweets.
A total of 5000 subjective Twitter posts were collected for corpus. Once this data was collected, it was separated into tweets that contain only negative emotions, tweets that contain only positive emoticons and tweets that contain both emoticons (negative and positive).

4.2 Supervised Approaches

This is where most of the time and effort was spent in our thesis. Under this section, different supervised machine learning approaches were used and explained. To be able to experiment with different machine learning algorithms and to enable the identification of factors that affect our results, we used a three step process. The first step is the pre processing of the training data[12,9]. The second is feature extraction and data representation in a particular format (Semantic VSM here). The third is the classifier training and testing phase with different machine learning algorithms. This approach helped in experimenting with different possibilities by varying the pre-processing operations, feature extraction and then using the machine learning algorithms. Each of this will be explained in the following subsections.[12,5]

4.2.1 Pre-processing

Cleaning of the data:
Since the corpus we are taking consists of several syntactic features that may not be useful for machine learning, the data needs to be cleaned for further use.
Tokenization: In this process, we have broken the stream of text up into words, phrases, and symbols and called them as tokens. The frequency of a particular token in the whole dataset is also counted. With that we have also taken out total number of unique words in our data set.

Stemming: In this process we have reduced the inflected (derived) words to their stem or root word form. For this we have taken into account that stem need not to be identically same to the morphological root of the word; it is sufficient that related words could map to the same stem, even if the stem is not itself a valid root.

Stop word removal: After tokenization, we are having a big list of words occurring in datasets, and mostly not all of them are useful for learning task. It is imperative to reduce the size of feature space as far as possible. So stop word removal step will be done prior to tokenization to remove all occurrences of these useless words like ‘a’, ‘an’, ‘the’, ‘is’ etc.

Filtering: During filtering process various sub tasks were done:
URL: Url were removed, in order to reduce the feature size during training.
Emoticons: All emoticons were replaced by pos and neg classes.
Username and Hash tags: We replaced them with <un> and <ht> respectively.
Removal of repeated tweet
Lowercasing: All characters were lowercased to ensure that all tokens map to the corresponding feature irrespective of casing.

4.2.3 Machine learning algorithms

According to the literature, multinomial Bayes classifier and Support vector machines are found to give better accuracy in mostly every cases. We experimented with one set of algorithms in this thesis. We used Bayes classifiers and in it we have tested with Naïve Bayes classifier and Multinomial Naïve Bayes. Basic processing of Bayes classifiers are explained in subsequent sections

4.2.3.1 Bayes classifiers

All the Bayesian models are the derivatives of the well known Bayes Rule. Bayes Rule says that the probability of a hypothesis given certain evidence, i.e. the posterior probability of a hypothesis, could be obtained in terms of the prior probability of the evidence very easily, the prior probability of the hypothesis and the conditional probability[12] of the evidence given the hypothesis already. Mathematically, it could be depicted as

$$P(A|B) = {(P(A).P(B|A))}/{P(B)}$$

Where,
P (A|B) = posterior probability of the hypothesis
P (A) = prior probability of hypothesis
P (B) =prior probability of Evidence
P (B/A) = conditional probability of Evidence given Hypothesis
In our case, we would have two hypotheses and the one that has the highest probability would be chosen as a class of the tweet whose sentiment is being predicted.

4.2.3.2 Multinomial Experiment

An experiment is a procedure that has got three things: each procedure can have more than one outcome, each outcome is known to be in advance, and there is some uncertainty in each outcome. Tossing a coin is considered as an experiment because it has more than one outcome. Tossing a coin is an experiment because it has more than one outcome (head and tail), head and tail are known as outcomes before calling them as the experiments, and whether it will be head or tail is entirely dependent on chances.
A multinomial experiment is an experiment that has got four properties[12,13].
The experiment is said to be repeated n times(n trials)- throwing dice 10 times
Each trial could result in a discrete number of outcomes – 1 through 6.
The probability of any outcome remains constant e.g. probability of getting 1, 2, 3, 4, 5 or 6 is 1/6 at any time the dice is thrown.
The trials are said to be independent; that is , getting a particular outcome on one trial does not affect the outcome

on other trials e.g. getting 3 in trial 1 does not have any effect in getting 3 or any of the other outcomes in subsequent experiments.

Suppose a multinomial experiment consists of a total of n trials, and each trial can result in any of the k possible outcomes: E1,E2,….Ek. Also suppose that each possibility can occur with probabilities p1, p2…pk. Then, the probability (P) that E1 occurs n1 times, E2 occurs n2 times… and Ek occurs nk times is given as follows:[12]

$$P = \left[n!/n1! * n2! * \ldots\ldots nk! \right] * (p1^{n1} * p2^{n2} * \ldots\ldots pk^{nk})$$

Where n=n1+n2+…..+nk upon rearranging, it becomes

$$P = \frac{n! \prod pi^{ni}}{ni!}$$

Where k is the number of outcomes

Multinomial Naïve Bayes:

Multinomial Naïve Bayes uses this Multinomial distribution only. Thus the probability of a Twitter post being in a certain class is depicted by the formula:

$$P = \frac{n! \prod pi^{ni}}{ni!}$$

In our case, n would be the total number of attributes used (the total number of trials), k is the number of outcomes of the experiment performed (the attributes found in a single Twitter post about news), Pi is the probability of the ith outcome of an experiment (an attribute found in Twitter post) and ni is the number of times the ith attribute occurs in that Twitter post. Since an attribute can occur 0, 1,2,3,4 etc times, the experiment is said to be Multinomial. This formula is used to complete the probability of each sentiment class given. During the computation of whether a certain Twitter post is positive or negative, only the values of Pi will change, all the others will remain as it is. Pi, changes here because the probability of an attribute in the positive and negative classes are obviously different. The factorials of n and ni helps us to account for the fact that the order of occurrences of the attributes in the Twitter post does not matter at all. But since factorials of n and ni are the same for each classes and thus do not help in the comparison of the probability of a Twitter post being in a certain class, they can be dropped by just simplifying the formula to the given below[12,13,7]:

$$P = \prod pi^{ni}$$

This is simply the application of multiplication rule of probabilities to certain probabilities of each attributes to the number of times that attribute occur. The formula here will be a modification of the standard Bayes rule to accommodate the frequency of occurrence of an attribute in a certain Twitter post. Basically what it means is that if an attribute is occurring twice, it will be taken into account by multiplying its probability twice.

The probability of each attribute in each sentiment class is obtained during the training. For example, if we have got an attribute, say 'season' in a certain Twitter post, it , will have two probabilities, one for each sentiment class negative or positive. Given a tweet, the evidence (E), the formulas for the probability for a Twitter post being positive becomes

$$P(pos|E) = \prod Ppos^{ni}$$

The same formula [9] is applied for negative class also. After the probabilities of a certain Twitter post being produced by each of the three classes has been computed, the one that produces it with the highest probability becomes the sentiment of the class. This has assumed that the two sentiment classes have the same probability of existence. If they do not have this, the formula will become [14,9]

$$P(pos|E) = \left(\prod Pnew\right) * P(pos)$$

The result will be multiplied by the probability of the sentiment class.

❖ *Clustering:*

We have also applied two clustering algorithms on our Semantic VSM they are:
1) K-means clustering.
2) Hierarchical clustering.

❖ *EXPERIMENTS*

. All work is done using WEKA's filtering tools and machine learning algorithms. Filtered Classifier algorithm of WEKA was very useful in doing that. The Filtered Classifier algorithm also helps to enable the processing of test data in the same way the training data was processed. In Filtered Classifier, the learning algorithm (Naive Bayes, Naive Bayes

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

Multinomial) and filtering tools (StringToWord Vector) are provided as parameters.

4.3 Results

In all the tables below, the same training data and test data has been used. The total training data used is 5000 (positive and negative tweets respectively). This is done not to bias the classifier in favour of any of the classes.
Now, since the first task we done were conversion of unlabelled data into labelled data. Shown in the figure below are labelled and unlabelled data formats.

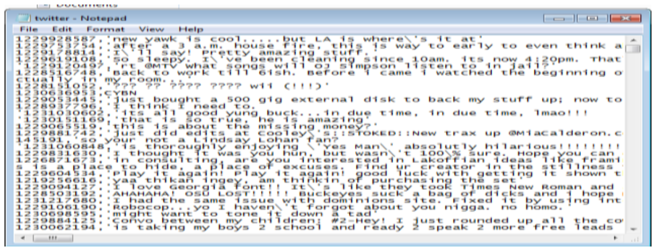

```
1229928587,'new yawk is cool.....but LA is where\'s it at'
1229753754,'after a 3 a.m. house fire, this is way to early to even think a
1229178814,'I\'ll say! Pretty amazing stuff.'
1229619108,'So sleepy. I\'ve been cleaning since 10am. its now 4:20pm. That
'1229120497','rt @MTV what songs will OJ Simpson listen to in jail?'
1228516748,'Back to work till 6ish. Before I came I watched the beginning o
ctually in my room...
1228151052,'??? ?? ???? ???? wii (!!!)'
1230636953,CYBN
1229053445,'just bought a 500 gig external disk to back my stuff up; now to
1228937796,'I think I need to.
'1231030602,'its all good yung buck...in due time, in due time, lmao!!!
'1230151169,'that is so true, he is amazing'
1229065519,'this is about the missing money?'
1229881742,'just did edits at Cooley\'s::STOKED::New trax up @Miacalderon.c
145191,'so your not a Lindsay Lohan fan?
'1231060848,'is thoroughly enjoying \'Yes Man\' absolutly hilarious!!!!!!!!
1229831630,'I thought it was you hun, but wasn\'t 100\% sure. Hope you can
1226871673,'in consulting, are you interested in Lakoffian ideas like frami
s is a place to hide, a place of excuses. Find ur creator in the stillness
1229604534,'Play it again! Play it again! good luck with getting it shown t
1219256616,'yaa thikan ingey. am thinkin of purchasing the set'
1229094127,'I love Georgia font!! It\'s like they took Times New Roman and
1228503192,'AHAHAHA! OSU LOST!!!!!! Buckeyes suck a bag of dicks and i hope
1231217680,'I had the same issue with dominions site. Fixed it by using int
1229106190,'Robocop...yo I haven\'t forgot about you nigga. no homo.'
1230698595,'might want to tone it down a tad'
1229884125,'Convo between my children: #2-Hey! I just rounded up all the co
1230062194,'is taking my boys 2 school and ready 2 speak 2 more free leads
```

Figure 1: unlabelled data

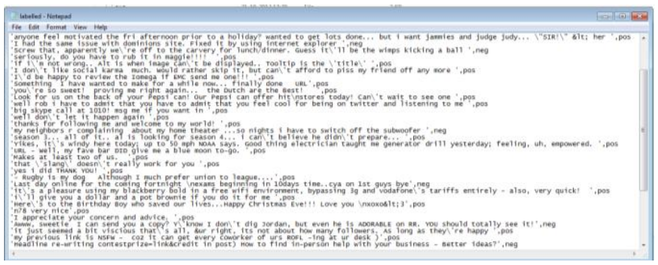

```
'anyone feel motivated the fri afternoon prior to a holiday? wanted to get lots done... but i want jammies and judge judy... \"SIR!\" &lt; her ',pos
'I had the same issue with dominions site. Fixed it by using internet explorer ',neg
'Screw that, apparently we\'re off to the carvery for lunch/dinner. Guess it\'ll be the wimps kicking a ball ',neg
'seriously, do you have to rub it in maggie!!!!  ',pos
'if i\'m not wrong.. Alt is when image can\'t be displayed.. Tooltip is the \'title\' ',pos
'I don\'t like social karma  much. would rather skip it, but can\'t afford to piss my friend off any more ',pos
'I\'d be happy to review the Iomega if EMC send me one!!! ',pos
'Something  I have wanted to make for a while now... finally done   URL',pos
'you\'re so sweet!  proving me right again...  the Dutch are the Best!    ',pos
'Look for us on the back of your Pepsi can! Our Pepsi can offer hit\nstores today! Can\'t wait to see one ',pos
'well rob i have to admit that you have to admit that you feel cool for being on twitter and listening to me ',pos
'big skype call at 1010! msg me if you want in ',pos
'well don\'t let it happen again ',pos
'thanks for following me and welcome to my world! ',pos
'my neighbors r complaining  about my home theater ...so nights i have to switch off the subwoofer ',neg
'season 3... all of it.. al is looking for season 4... i can\'t believe he didn\'t prepare... ',pos
'Yikes, it\'s windy here today; up to 50 mph NOAA says. Good thing electrician taught me generator drill yesterday; feeling, uh, empowered. ',pos
'URL - well, my fave bar DID give me a blue moon to-go. ',pos
'Makes at least two of us.  ',pos
'that \'slang\' doesn\'t really work for you ',pos
'yes i did THANK YOU! ',pos
'- Rugby is my dog   Although I much prefer union to league....',pos
'Last day online for the coming fortnight \nexams beginning in 10days time..cya on 1st guys bye',neg
'it\'s a pleasure using my blackberry bold in a free wifi environment, bypassing 3g and vodafone\'s tariffs entirely - also, very quick!  ',pos
'i\'ll give you a dollar and a pot brownie if you do it for me ',pos
'Here\'s to the Birthday Boy who saved our lives...Happy Christmas Eve!!! Love you \nxoxo&lt;3',pos
'n78 very nice',pos
'I appreciate your concern and advice. ',pos
'Awww, sweetie  I can send you a copy? Y\'know I don\'t dig Jordan, but even he is ADORABLE on RR. You should totally see it!',neg
'It just seemed a bit viscious that\'s all, &ur right, its not about how many followers. As long as they\'re happy ',pos
'my previous link is NSFW -  coz it can get every coworker of urs ROFL -ing at ur desk )',pos
'Headline re-writing contestprize=link&credit in post) How to find in-person help with your business - Better ideas?',neg
```

Figure 2: labelled data

Here in the labelled data the classes positive and negative has been assigned by removing the smiley and frowsy by pos and neg classes respectively.
After this the data is ready for pre processing, during pre processing several activities will be done:

- ❖ Conversion of upper-case into lower-case.
- ❖ Removal of URLs.
- ❖ Stop-word removal
- ❖ Tokenization.

For stop-word removal we have maintained a file in which we have kept generally used stop- words. Our program will go through this file of commonly used stop-words, which are more than 200 words. As soon as a stop-word is encountered it is removed because it is not helpful in imparting any kind of sentiment to our data.

And finally after the removal of stop-words, words are tokenized. Below will be shown the file used for stop-word removal and final output of tokenized words.

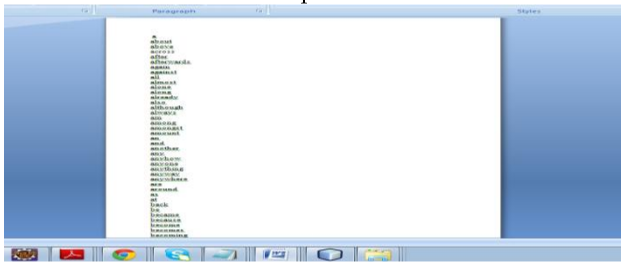
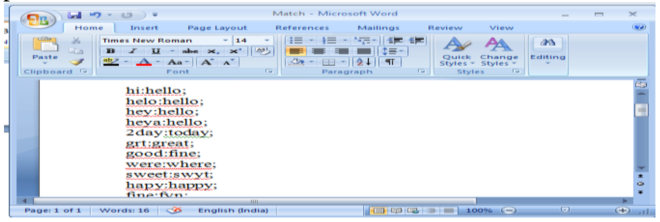

```
a
about
above
across
after
afterwards
again
against
all
almost
alone
along
already
also
although
always
am
among
amongst
amount
an
and
another
any
anyhow
anyone
anything
anyway
anywhere
are
around
as
at
back
be
became
because
become
becomes
becoming
```

Figure 3: stop-word file

This file has the common stop-words used which does not impart any kind of knowledge to our data.
We will provide three files as input to our program they are:
Data file:-
It is basically the file which has got the Twitter corpus data.
Stop-word file:-
It will have the common stop-words which we have to remove from our corpus which does not impart any sentiment to our data.
Semantic file:-It has a pattern match syntax, where 'yeah: yes; yeh: yes' are all equal and will be treated as one if seen in the data file. Below shown in the figure is the semantic pattern file for data.



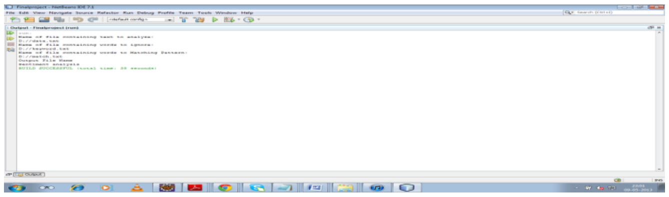
```
hi:hello;
helo:hello;
hey:hello;
heya:hello;
2day:today;
grt:great;
good:fine;
were:where;
sweet:swyt;
hapy:happy;
fine:fyn;
```

Figure 4: Semantic match file

Data pre processing was the second step in our thesis, now our aim is to design a semantic vector space model for knowledge representation purpose.
We have stored the tokens which we got in hash-bags. Simply checking the frequency of above tokens in our data file will give us a Vector space model, which has rows as tokens or words and documents as columns, and the matrix gives the frequency count of the token present. If token is not present then it is shown by 0 and if token is present once it is shown by 1. Basically what we will get will be a very large sparse matrix.
But, we did not adopted the idea of forming a simple VSM, instead we opted for Semantic VSM. Here in Semantic VSM as discussed earlier we made a semantic correction file .
We have below shown the output of pre processing in the form of tokenization which is breaking of words into tokens.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013



```
run:
Name of file containing text to analyse:
D://data.txt
Name of file containing words to ignore:
D://keyword.txt
Name of file containing words to Matching Pattern:
D://match.txt
Output File Name
Sentiment analysis
BUILD SUCCESSFUL (total time: 30 seconds)
```

Figure 5: Output will produce .ARFF file as output

Here in this output window, as we can see we are providing three files as input: Data file consisting of Twitter corpus to be analysed, stop-word removal file and lastly the semantic matching file and lastly the output comes out to be an Arff file at a destination given in the program itself.
We are keeping frequency calculations for making the Semantic VSM. Finally we will made an n*m matrix where row corresponds to the data file or data corpus, and column corresponds to tokens or words present in the hash bags.

Generation of Semantic Vector space matrix has been taken place; now the most important work will be testing it under different machine learning algorithms, and comparing them with the previous results.

Machine learning algorithm	Accuracies
Naïve Bayes	83.908%
Multinomial Naïve Bayes	92.954%

Table 1: Accuracies using frequency representation

Since we are making use of WEKA learning kit here, we will need to convert our Semantic VSM into CSV format (Comma Separated Values) or into Arff format (Attribute Relation File Format). Here we have chosen Arff file format to covert our Semantic VSM into.

Figure 6: Generated Arff. file

This is the generated .Arff file at a destination location given already in the program. Now this .Arff file can be fed as input to the WEKA software and various types of testing experiments like clustering and classification can be performed over it without any hassle as there is restriction with WEKA software that either it can use CSV file format or Arff file format, normal file formats other than these are not allowed.

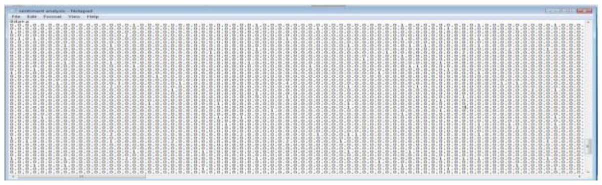


Figure 7: Generated SVSM

This generated Semantic VSM will be kept under testing of various machine learning algorithms.
In each of the algorithms, the same number of attributes are selected on the basis of frequency are used. The total number of attributes used is around 400. Before attribute selection, the training data was converted to lowercase. Stop words file was used and thus stop words which does not impart any knowledge were removed. The lower limit of frequency for an attribute to be chosen was set to 1. The results are presented in the tables below:
The tables show performance of two different machine learning algorithms under two choices of representations. The representations are frequency and Semantic VSM. Under each representation, we have used uni-grams. Multinomial Naïve Bayes achieves best result for each representation. Its best result is an accuracy of 92.954% .

Machine learning algorithm	Accuracies
Naïve Bayes	79%
Multinomial Naïve Bayes	84%

Table 2: Accuracies using Semantic VSM representation

Machine learning algorithm	**Precision (By class)**	**Recall**	**F-measure**
Naïve Bayes	0.927 positive class 0.784 negative class	0.75 positive class 0.93 negative class	0.825 positive class 0.851 negative class
Multinomial Naïve Bayes	0.898 positive class 0.974 negative class	0.987 positive class 0.86 n egative class	0.935 positive class 0.914 negative class

Table 3: Performance measurements for classification

Clustering algorithm	**Correctly clustered instances (%)**
K-means	56.12%
Hierarchical clustering	59.13%

Table 4: Clustering accuracies

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

Vary the number of attributes keeping the training data constant and see if that improves our performance. The experiment showed that performance is not linearly proportional to the number of attributes. However, it showed that the best result out of the attributes seen in the graph was to be found at attribute numbers of 400 and 500. At both attribute numbers, an accuracy of 91% is recorded. The result of varying attributes is presented in the graph below.

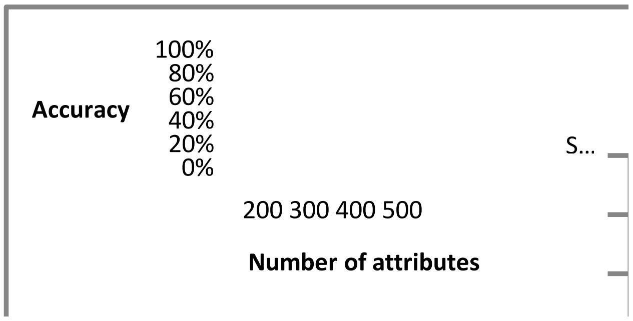


Figure 8: Graph of increasing attributes

4.4.2 Increasing the number of training data

We saw above that varying the number of attributes varies the accuracy. Initially, a total number of 5000 tweets were collected to be used as training data. As the experimentation was being done, some more training data was collected on the go. Now, it might be interesting to see how varying the number of training data affects our accuracy. This was done keeping the number of attributes constant at 600 (at which one of the best results was obtained). Increasing the number of training data did lead to some increment of accuracy, although not in a linear fashion particularly. An accuracy of 89% was obtained using 5100 number of training data, but it started to decrease altogether. Normally, the learning algorithm is expected to achieve better results as the number of training data is increased. But is this accuracy going to increase indefinitely as the number of training data is increased? If yes, then we will reach 100% accuracy. This is a theoretical assumption, and it may not happen in reality nor are we going to prove it here. The interest here is to obtain as high accuracy as possible with the number of training examples that has been collected.
The maximum number of training data collected was 6000 tweets. This is after all the processing operations including duplicate removal has been completed. We took this training data and experimented with it by varying the number of attributes. However, there was no improvement in the performance recorded. The best performance in terms of accuracy remains the one achieved using 5100 number of training data and 400 to 500 attributes, accuracy was 91%.

V. Conclusion

In this paper, the way to represent a semantic VSM form was proposed. Using the Semantic VSM we were able to achieve an accuracy of more than 90% using Multinomial Naïve Bayes. The main challenge was to find a perfect representation of the data set. It was to be keeping in mind that the representatives do not overlap each other and they constitute a optimal representative for their category. It is really very important to identify the factors that improve the performance as well. Factors that affect performance are the choice of training data, attribute or feature selection, representation of instances, and the choice of the algorithm used. A decision to make equal number of each class that is positive and negative was being made to avoid biasing of classifier in favour of any particular class. Attribute Selection affected the accuracy that we can get from the classifier. Too few attributes can cause the under fitting and too many attributes causes over fitting. Under fitting is a situation in which the classifier is not biased enough to make prediction on some unseen instances. Over fitting is a situation where the trained model is highly fitted to the training data that it basically fails while predicting new instances successfully. Thus it is good to find a balance between too few and too many attributes. Unfortunately, there seems to be no as such simple way of finding this balanced number of attributes except by trial and error. The other factor that affected the performance is the representation of the instances. This basically includes whether to use presence/absence or count/ frequency. It also includes whether to convert count into tf-idf so that term's importance is taken into account. The best results for multinomial Naïve Bayes can be obtained using this presence. Finally, the choice of machine learning algorithm for the task also affected the performance. Multinomial Naïve Bayes was found to outperform others in the task of sentiment analysis/ classification

ACKNOWLEDGMENT

Thanks to Anil Saroliya and my friends for their comments. Thanks to Swati Agnihotri for promoting this paper and for developing the distributed vector-space model and sparse-matrix. Thanks to the IJARCCE reviewers for their very helpful comments and suggestions.

References

1. BERNARD J. JANSEN, MIMI ZHANG, KATE SOBEL, ABDUR CHOWDURY. Twitter Power: Tweets as Electronic Word of Mouth. Journal of the American Society for Information Science and Technology (2009).
2. TURNEY, P. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02) (2002).
3. HATZIVASSILOGLOU, V., AND WIEBE, J. Effects of adjective orientation and tradability on sentence subjectivity. In Proceedings of the

18th conference on Computational linguistics-Volume 1, Association for Computational Linguistics, pp. 299–305 (2000).
4. HU, M., AND LIU, B. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining ACM, pp. 168–177 (2004).
5. READ, J., AND CARROLL, J. Weakly supervised techniques for domain-independent sentiment classification. In Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, pp. 45–52 (2009).
6. PANG, B., AND LEE, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, p. 271 (2004).
7. PANG, B., L EE, L., AND V AITHYANATHAN, S. Thumbs up?: sentiment classification Using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, pp. 79–86 (2002).
8. BOIY, E., H ENS, P., D ESCHACHT, K., AND MOENS, M. Automatic sentiment analysis in on-line text. In Proceedings of the 11th International Conference on Electronic Publishing, pp. 349–360 (2007).
9. CHAOVALIT, P., AND ZHOU, L. Movie review mining: A comparison between supervised and unsupervised classification approaches (2009).
10. LI, S., AND ZONG, C. Multi-domain sentiment classification. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers , Association for Computational Linguistics, pp. 257– 260 (2008).
11. YANG, H., S I , L., AND CALLAN, J. Knowledge transfer and opinion detection in the 32 trec 2006 blog track. In Proceedings of TREC vol. 120, Cite seer (2006).
12. LIN, C., AND HE Y. Joint sentiment /topic model for sentiment analysis. In Proceeding of the 18th ACM conference on Information and Knowledge management ACM, pp. 375-384 (2009).
13. MEI, Q., LING, X., W ONDRA, M., S U, H., AND ZHAI, C. Topic sentiment mixture: modeling facets and opinions in weblogs. In Proceedings of the 16th international conference on World Wide Web , ACM, pp. 171–180 (2007).
14. YU, H., AND HATZIVASSILOGLOU, V. towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Proceedings of the 2003 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, pp. 129–136 (2003). GO, A., B HAYANI, R., AND HUANG, L. Twitter sentiment classification using distant supervision.

BIOGRAPHY

Vijay Dixit is a software engineer and research scholar .He received his M.Tech in computer science and engineering from the Amity University at Jaipur. His current research interests are sentiment analysis and semantic vector space model.